Coursework - Probability for Statistics

CID: 02091191

November 23, 2021

1 First exercise

Let Ω be a nonempty set.

(a) Let $\mathcal{C} = \{A, B, C\}$ be a partition of Ω. The σ-algebra generated by the partition $\mathcal{C}$ noted $\sigma(\mathcal{C})$ is

$$\sigma(\mathcal{C}) = \{\emptyset, \{A, B, C\}, \{A\}, \{B\}, \{C\}, \{B, C\}, \{A, C\}, \{A, B\}\}$$

□

(b) Suppose that $\mathcal{C}$ is a countable partition of Ω. Let us prove that every element of $\sigma(\mathcal{C})$ is a countable union of elements of $\mathcal{C}$.

Let us write $\mathcal{C} = \{C_i,\ i \in \mathbb{N}\}$ a countable partition of Ω. Let $\Gamma = \left\{\bigcup_{i\in I} C_i,\ I \subseteq \mathbb{N}\right\}$ be the set of all countable unions of elements of $\mathcal{C}$. We first show that Γ is a σ-algebra:

- $\Omega \in \Gamma$: as $\mathcal{C}$ is a partition of Ω, $\bigcup_i C_i = \Omega$. Hence, Γ is not empty.
- Γ is closed under countable unions: since every element of Γ are sets of countable unions of elements of $\mathcal{C}$, Γ is closed under countable unions: let K be a countable set and $(A_k)_{k\in K}$ be a sequence of elements of Γ. Then, for all $k \in K$, there exists J_k countable such that $A_k = \bigcup_{j\in J_k} C_j$. Hence, $\bigcup_{k\in K} A_k = \bigcup_{k\in K}\bigcup_{j\in J_k} C_j$ which is a countable union of elements of $\mathcal{C}$.
- Γ is closed under complement: let $A \in \Gamma$. Thus, there exists $I \subseteq \mathbb{N}$ countable such that $A = \bigcup_{i\in I} C_i$. Then, $A^c = \Omega \setminus A = \bigcup_{i\in\mathbb{N}} C_i \setminus \bigcup_{i\in I} C_i = \bigcup_{i\in\mathbb{N},\ i\notin I} C_i$ which is a countable union of elements of $\mathcal{C}$ and thus $A^c \in \Gamma$.

Hence, Γ is a σ-algebra. Also, as Γ is the set of all countable unions of elements of $\mathcal{C}$, we have $\mathcal{C} \subseteq \Gamma$: thus, Γ is a σ-algebra containing $\mathcal{C}$. Moreover, by definition, $\sigma(\mathcal{C})$ is the intersection of all σ-algebras containing $\mathcal{C}$ (i.e. the smallest σ-algebra containing $\mathcal{C}$), thus $\sigma(\mathcal{C}) \subseteq \Gamma$. Hence, every element of $\sigma(\mathcal{C})$ is a countable union of elements of $\mathcal{C}$. □

Remark: As $\sigma(\mathcal{C})$ is the σ-algebra generated by $\mathcal{C}$, $\sigma(\mathcal{C})$ contains all countable unions of elements of $\mathcal{C}$, therefore $\sigma(\mathcal{C})$ contains all the elements of Γ. Hence, $\Gamma \subseteq \sigma(\mathcal{C})$ and therefore, $\Gamma = \sigma(\mathcal{C})$.

(c) Let $f : \Omega \to \mathbb{R}$ be a function. Let us show that f is measurable with respect to $(\Omega, \sigma(\mathcal{C}))$ if and only if f is constant on each member of that partition.

f is measurable with respect to $(\Omega, \sigma(\mathcal{C}))$ if and only if, for all $y \in \mathbb{R}$, $f^{-1}(\{y\}) \in \sigma(\mathcal{C})$, i.e. $f^{-1}(\{y\})$ is a countable union of elements of $\mathcal{C}$ as shown in (b), i.e. $f^{-1}(\{y\}) = \bigcup_{i\in I} C_i$ where $I \subseteq \mathbb{N}$. Hence, f is constant on each member of the partition $\mathcal{C}$. □

2 Second exercise

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and X be a non-negative random variable on $(\Omega, \mathcal{F}, \mathbb{P})$ with distribution function $F(\cdot)$.

(a) First, let us show that for all $p \in \mathbb{N}$

$$\mathbb{E}[X^p] = \int_0^\infty p s^{p-1}(1 - F(s))ds$$

By definition, for a non-negative random variable,

$$\mathbb{E}[X^p] = \int_\Omega (X(\omega))^p d\mathbb{P}(\omega) = \int_0^\infty x^p dF(x)$$

However, notice that

$$\int_0^x p u^{p-1} du = x^p$$

Thus,

$$\mathbb{E}[X^p] = \int_0^\infty \int_0^x p u^{p-1} du dF(x)$$

Then, since the Lebesgue measure is a σ-finite measure on $([0,\infty), \mathcal{B}_{[0,\infty)})$ and given that $g : (u, x) \longmapsto p u^{p-1}$ is non-negative and measurable on $\overline{\mathcal{B}_{[0,\infty)} \times \mathcal{B}_{[0,\infty)}}$, Tonelli's theorem allows to write

$$\mathbb{E}[X^p] = \int_0^\infty \int_u^\infty dF(x) p u^{p-1} du$$
$$\mathbb{E}[X^p] = \int_0^\infty p u^{p-1}(1 - F(u))du$$

□

(b) Let $X_1, \ldots, X_p$ be a set of independent identically distributed random variables with the same distribution as X. Let us show that

$$\mathbb{E}[\min(X_1, \ldots, X_p)] = \int_0^\infty (1 - F(s))^p ds$$

This can be shown by several ways, but we will here use the result found in (a) to prove this new result. Consider the distribution function of the random variable $\min(X_1, \ldots, X_p)$ defined as, for all x in $\mathbb{R}$

$$\begin{aligned} F_{\min(X_1,\ldots,X_p)}(x) &= \mathbb{P}[\min(X_1, \ldots, X_p) \leq x] \\ &= 1 - \mathbb{P}[\min(X_1, \ldots, X_p) > x] \\ &= 1 - \prod_{i=1}^p \mathbb{P}[X_i > x] \\ &= 1 - \prod_{i=1}^p (1 - F(x)) \\ F_{\min(X_1,\ldots,X_p)}(x) &= 1 - (1 - F(x))^p \end{aligned}$$

Therefore, we can now use the result proved in (a) as $\min(X_1, \ldots, X_p)$ is a non-negative random variable

$$\mathbb{E}[\min(X_1, \ldots, X_p)^1] = \int_0^\infty 1 \cdot s^{1-1}(1 - F_{\min(X_1,\ldots,X_p)}(s))ds$$

Thus,

$$\mathbb{E}[\min(X_1, \dots, X_p)] = \int_0^\infty (1 - (1 - (1 - F(s)))^p) ds$$
$$\mathbb{E}[\min(X_1, \dots, X_p)] = \int_0^\infty (1 - F(s))^p ds$$

□

Remark: the same result could have been found using that, for all non negative random variable X, $\mathbb{E}[X] = \int_0^\infty \mathbb{P}[X > x] dx$. Applying this result to the random variable $\min(X_1, \dots, X_p)$ would give

$$\begin{aligned}
\mathbb{E}[\min(X_1, \dots, X_p)] &= \int_0^\infty \mathbb{P}[\min(X_1, \dots, X_p) > x] dx \\
&= \int_0^\infty \mathbb{P}[X_1 > x, \dots, X_p > x] dx \\
&= \int_0^\infty \mathbb{P}[X_1 > x] \dots \mathbb{P}[X_p > x] dx \\
&= \int_0^\infty (1 - \mathbb{P}[X \leq x])^p dx \\
\mathbb{E}[\min(X_1, \dots, X_p)] &= \int_0^\infty (1 - F(x))^p dx
\end{aligned}$$

3 Third exercise

Let X_n for $n \in \mathbb{N}$ be sequence of random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ and let X be a random variable on the same probability space.

(a) Suppose that $X_n \xrightarrow{L^r} X$ where $r \geq 1$, i.e.

$$\lim_{n \to \infty} \mathbb{E}\left[|X_n - X|^r\right] = 0$$

Using Minkowski's inequality,

$$(\mathbb{E}|X_n + (X - X_n)|^r)^{1/r} \leq (\mathbb{E}|X_n|^r)^{1/r} + (\mathbb{E}|X - X_n|^r)^{1/r}$$
$$(\mathbb{E}|X|^r)^{1/r} - (\mathbb{E}|X_n|^r)^{1/r} \leq (\mathbb{E}|X - X_n|^r)^{1/r}$$

In the same way,

$$(\mathbb{E}|X_n|^r)^{1/r} - (\mathbb{E}|X|^r)^{1/r} \leq (\mathbb{E}|X - X_n|^r)^{1/r}$$

And combining both gives

$$0 \leq |(\mathbb{E}|X_n|^r)^{1/r} - (\mathbb{E}|X|^r)^{1/r}| \leq (\mathbb{E}|X_n - X|^r)^{1/r}$$

Thus,

$$\lim_{n \to \infty} (\mathbb{E}|X_n|^r)^{1/r} - (\mathbb{E}|X|^r)^{1/r} = 0$$

Hence,

$$\lim_{n \to \infty} (\mathbb{E}|X_n|^r)^{1/r} = (\mathbb{E}|X|^r)^{1/r}$$

Then, we can apply $f : x \mapsto x^r$ with $r \geq 1$ on both sides, and we get by continuity of f

$$\lim_{n \to \infty} (\mathbb{E}|X_n|^r) = (\mathbb{E}|X|^r)$$

□

(b) Suppose that $X_n \xrightarrow{L^1} X$, i.e.

$$\lim_{n\to\infty} \mathbb{E}\left[|X_n - X|\right] = 0$$

Notice that, using Jensen's inequality with $\varphi : x \mapsto |x|$ providing $\varphi(\mathbb{E}[X_n - X]) \leq \mathbb{E}[\varphi(X_n - X)]$ as φ is convex on $\mathbb{R}$, we have

$$\begin{aligned}\mathbb{E}[|X_n - X|] &\geq |\mathbb{E}[X_n - X]| \\ &= |\mathbb{E}[X_n] - \mathbb{E}[X]|\end{aligned}$$

Hence, for all n,

$$0 \leq |\mathbb{E}[X_n] - \mathbb{E}[X]| \leq \mathbb{E}[|X_n - X|]$$

And as $\lim_{n\to\infty} \mathbb{E}\left[|X_n - X|\right] = 0$,

$$\lim_{n\to\infty} \mathbb{E}[X_n] - \mathbb{E}[X] = 0 \text{ i.e. } \lim_{n\to\infty} \mathbb{E}[X_n] = \lim_{n\to\infty} \mathbb{E}[X]$$

□

(c) Suppose that $X_n \xrightarrow{L^2} X$. Consider

$$\mathrm{Var}[X_n] = \mathbb{E}[X_n^2] - \mathbb{E}[X_n]^2$$

By (a) with $r = 2$, we show that on the one hand,

$$\lim_{n\to\infty} \mathbb{E}[X_n^2] = \mathbb{E}[X^2]$$

On the other hand, as convergence in L^q implies convergence in L^p for $1 \leq p \leq q$, $X_n \xrightarrow{L^2} X \implies X_n \xrightarrow{L^1} X$ and using (b) by continuity of $x \mapsto x^2$

$$\lim_{n\to\infty} \mathbb{E}[X_n] = \mathbb{E}[X] \implies \lim_{n\to\infty} \mathbb{E}[X_n]^2 = \mathbb{E}[X]^2$$

Hence,

$$\lim_{n\to\infty} \mathrm{Var}[X_n] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \mathrm{Var}[X]$$

□

(d) Suppose $\lim_{n\to\infty} \mathbb{E}[X_n] = \mathbb{E}[X]$. We do not have necessarily $X_n \xrightarrow{L^1} X$. For issue, let (X_n) be a sequence of random variables defined as

$$\mathbb{P}[X_n = 0] = 1 - \frac{1}{n} \text{ and } \forall n \geq 1,\ \mathbb{P}[X_n = n] = \frac{1}{n}$$

And let X be defined as

$$\mathbb{P}[X = 1] = 1$$

Then, for all n,

$$\begin{aligned}\mathbb{E}[X_n] &= 0 \cdot \mathbb{P}[X_n = 0] + n \cdot \mathbb{P}[X_n = n] \\ &= 0 \cdot \left(1 - \frac{1}{n}\right) + n \cdot \frac{1}{n} \\ &= 1\end{aligned}$$

Also, $\mathbb{E}[X] = 1$. Hence, $\mathbb{E}[X_n] \to \mathbb{E}[X]$. However,

$$\mathbb{P}[|X_n - X| = 1] = 1 - \frac{1}{n} \text{ and } \mathbb{P}[|X_n - X| = n - 1] = \frac{1}{n}$$

Therefore,

$$\begin{aligned}\mathbb{E}[|X_n - X|] &= 1 \cdot \left(1 - \frac{1}{n}\right) + (n-1) \cdot \frac{1}{n} \\ &= 2 - \frac{2}{n} \to 2 \neq 0\end{aligned}$$

Hence, X_n does not converge to X in L^1. □

4 Fourth exercise

Consider an aircraft engine being driven by a engine control unit and especially a total of N replica processing units running in parallel. Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we model the time of failure of the i^{th} unit as the random variable X_i on $(\Omega, \mathcal{F}, \mathbb{P})$. We assume that $X_1, \ldots, X_N$ are independent and identically distributed. We define the system failure time $M_N = \max(X_1, \ldots, X_N)$ i.e. the entire system fails when all the units have failed.

(a) Let us show that M_N is a random variable on $(\Omega, \mathcal{F}, \mathbb{P})$, i.e. show that for all $t \in \mathbb{R}$, $\{M_n \leq t\}$ sets are measurable. Let $\omega \in \Omega$,

$$\begin{aligned}
\omega \in \{M_N \leq t\} &\iff \omega \in \{\max(X_1 \ldots, X_N) \leq t\} \\
&\iff (X_1(\omega) \leq t) \cap \ldots \cap (X_N(\omega) \leq t) \\
&\iff \omega \in \{X_1 \leq t\} \cap \ldots \cap \{X_N \leq t\} \\
&\iff \omega \in \bigcap_{i=1}^{N} \{X_i \leq t\}
\end{aligned}$$

where all $\{X_i \leq t\}$ are measurable sets as X_i are random variables on $(\Omega, \mathcal{F}, \mathbb{P})$. Hence, $\{M_n \leq t\}$ is measurable as finite intersection of measurable sets. Thus, M_N is a random variable on $(\Omega, \mathcal{F}, \mathbb{P})$. □

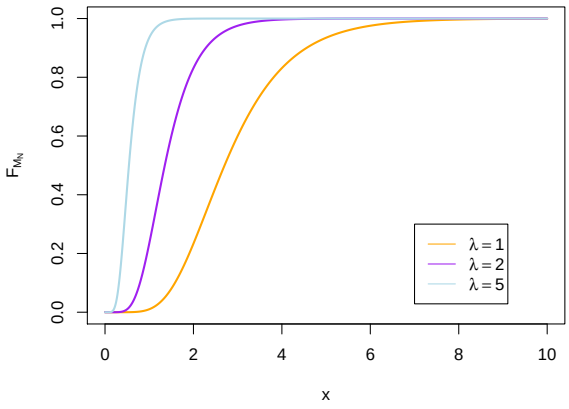
(b) Suppose that $X_1, \ldots, X_N$ are exponentially distributed random variables with rate λ. Let F_{M_N} be the distribution function of M_N. By definition, for all x in $[0, \infty)$

$$\begin{aligned}
F_{M_N}(x) &= \mathbb{P}\left[M_N \leq x\right] \\
&= \mathbb{P}\left[\max(X_1, \ldots, X_N) \leq x\right] \\
&= \mathbb{P}\left[X_1 \leq x, \ldots, X_N \leq x\right] \\
&= \prod_{i=1}^{N} \mathbb{P}\left[X_i \leq x\right] \text{ by independence} \\
&= \prod_{i=1}^{N} F_{X_i}(x) \\
&= \prod_{i=1}^{N} \left(1 - e^{-\lambda x}\right) \\
F_{M_N}(x) &= \left(1 - e^{-\lambda x}\right)^N
\end{aligned}$$

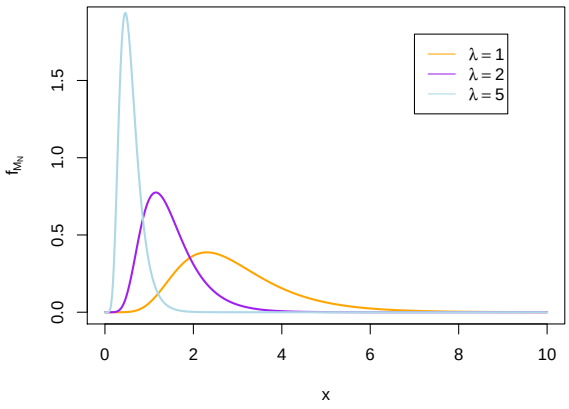
We then obtain the probability density function of M_N noted f_{M_N} defined by, for all x in $[0, \infty)$

$$\begin{aligned}
f_{M_N}(x) &= \frac{d}{dx} F_{M_N}(x) \\
f_{M_N}(x) &= N\lambda e^{-\lambda x} \left(1 - e^{-\lambda x}\right)^{N-1}
\end{aligned}$$

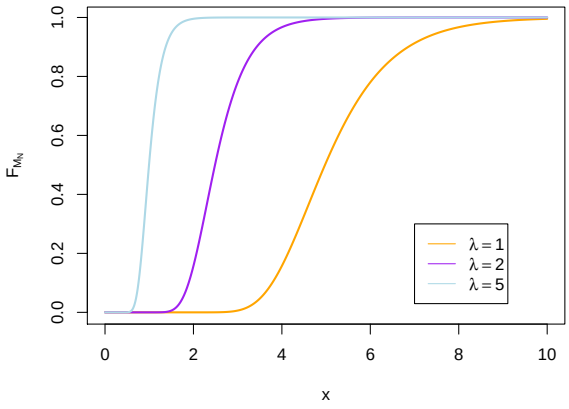
□



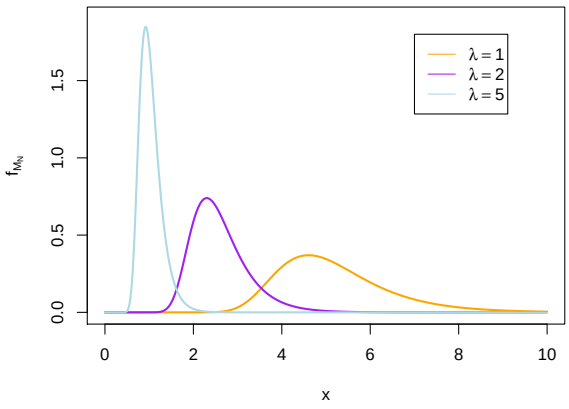
(a) Distribution functions, $N = 10$



(b) Density functions, $N = 10$



(c) Distribution functions, $N = 100$



(d) Density functions, $N = 100$

Figure 4.1: Examples of distribution and density functions for some values of λ and N

Figure 4.1 shows the effect of parameters λ and N on the shape and location of the distribution of M_N. As for a simple exponential distribution, as λ is getting smaller, the distribution and density functions show that high values of x are more likely to occur than for higher values of λ. Also, as N increases, the curves move to the right which is consistent since it corresponds to increase the number of experiments X_i, and thus the probability to have large values of M_N increases.

(c) Consider the distribution function of the sequence of random variables $M_N - \frac{1}{\lambda}\log(N)$, for all x in $[0, \infty)$

$$\begin{aligned}
\mathbb{P}\left[M_N - \frac{1}{\lambda}\log(N) \leq x\right] &= \mathbb{P}\left[\max(X_1,\dots,X_N) \leq x + \frac{1}{\lambda}\log(N)\right] \\
&= \mathbb{P}\left[X_1 \leq x + \frac{1}{\lambda}\log(N), \dots, X_N \leq x + \frac{1}{\lambda}\log(N)\right] \\
&= \prod_{i=1}^{N} \mathbb{P}\left[X_i \leq x + \frac{1}{\lambda}\log(N)\right] \\
&= \prod_{i=1}^{N} \left(1 - e^{-\lambda\left(x + \frac{1}{\lambda}\log(N)\right)}\right) \\
&= \left(1 - e^{-\lambda x - \log(N)}\right)^N \\
&= \left(1 - \frac{e^{-\lambda x}}{N}\right)^N \xrightarrow[N\to\infty]{} e^{-e^{-\lambda x}}
\end{aligned}$$

Hence, $M_N - \frac{1}{\lambda}\log(N)$ converges in distribution to a random variable X on $(\Omega, \mathcal{F}, \mathbb{P})$ with distribution $F(x) = e^{-e^{-\lambda x}}$ for all x in $[0, \infty)$. □

(d) Suppose we wish to study the system failure time of the engine under increasing redundancy. To do so, we need to generate independent realisations of M_N with N *very large*. Assume we can generate independent identically distributed $U[0,1]$ random variables $U_1, U_2, \dots$. We can then implement two different algorithms, one exact but inefficient, and another one approximate but efficient.

Exact but inefficient algorithm: Assume we want to generate p realisations of $M_N = \max(X_1,\dots,X_N)$ where $X_1,\dots,X_N$ are independent identically exponentially distributed random variables with parameter λ. The idea is to generate N exponentially distributed samples, compute the maximum and repeat this procedure p times. We can generate independent exponentially distributed observations from $U[0,1]$ observations by solving

$$y = 1 - e^{-\lambda x} \iff x = -\frac{1}{\lambda}\log(1-y) \text{ or similarly } x = -\frac{1}{\lambda}\log y$$

This algorithm is inefficient as it requires to generate $p \times N$ uniform observations, apply transformations and find a maximum p times over N observations.

Algorithm 1 Exact but inefficient

1: Generate N samples from $U[0,1]$ distribution, noted U
2: Transform U components to $Exp(\lambda)$ distribution $X = -\frac{1}{\lambda}\log(U)$
3: Generate M_N from the X observations by $M_N = \max(X_1,\dots,X_N)$
4: Repeat steps 1 to 3 p times

Not exact but efficient algorithm: Assume again we want to generate p realisations of $M_N = \max(X_1,\dots,X_N)$. The idea of this approximate algorithm is to consider that $M_N - \frac{1}{\lambda}\log(N)$ has distribution $e^{-e^{-\lambda x}}$ on $[0,\infty)$, which is only approximate. Then, we could write that M_N has (approximate) distribution $e^{-Ne^{-\lambda x}}$, and we could then generate directly M_N samples by inverting the distribution:

$$y = e^{-\lambda\left(x - \frac{1}{\lambda}\log N\right)} = e^{-Ne^{-\lambda x}} \iff x = -\frac{1}{\lambda}\log\left(-\frac{\log y}{N}\right)$$

This method is called inversion method and consists in computing the quantile function (analytically or numerically). Hence, this method only requires to generate p uniform observations and apply a transformation.

Algorithm 2 Approximate but efficient

1: Generate p samples from $U[0,1]$ distribution, noted U
2: Transform U components to M_N approximate distribution using $X = -\frac{1}{\lambda}\log(-\frac{\log U}{N})$

(e) Now that we have written two different algorithms to generate an empirical distribution of p observations of M_N for some N, we may compare the quality of sampling for each algorithm. The idea we have in the following is to proceed to Kolmogorov-Smirnov tests in **R** in order to compare the samples to the real distribution, with exact distribution function $F_{M_N}(x) = \left(1 - e^{-\lambda x}\right)^N$ as computed in (b). Doing such a test is more powerful than just comparing some statistic (like only comparing the mean, the median or the mode for example). The formal hypothesis of the KS tests are here: the null hypothesis H_0: samples from algorithm k are from M_N distribution against the alternative hypothesis H_1: samples from algorithm k are not from M_N distribution. We will do these tests for multiple values of N and for both algorithms $k = 1, 2$.

We need to be careful as we use samples from a uniform distribution U, which would thus lead to different results for different samples. First, as comparing for only one situation would not be very consistent, we will run the procedure 100 times so that we will obtain 100 different p-values for each chosen N and therefore have a more precise idea of the accuracy of the sampling algorithms in general and not only in one case. Second, we will fix the seed in **R** so that for all values of N, we will use exactly the same uniform distributions. The full procedure may be summarised as follows:

Algorithm 3 Multiple KS tests for a given N

1: Fix the seed
2: Initialise λ, N, p (the number of samples) and Ps (an empty list to store all p-values)
3: **for** `i in 1:100` **do**
4: $U \sim U[0,1]$
5: Compute `Mn1` samples using the exact Algorithm 1
6: Compute `Mn2` samples using the approximate Algorithm 2
7: Proceed to KS test for `Mn1` and `Mn2` comparing them separately, in two different tests, to the known cdf F_{M_N}
8: Add to Ps the p-values of each test
9: **end for**
10: Compute the mean p-values from each algorithm (mean of Ps columns) ▷ Ps is here a matrix with dimensions (100,2)
11: Repeat for all chosen N

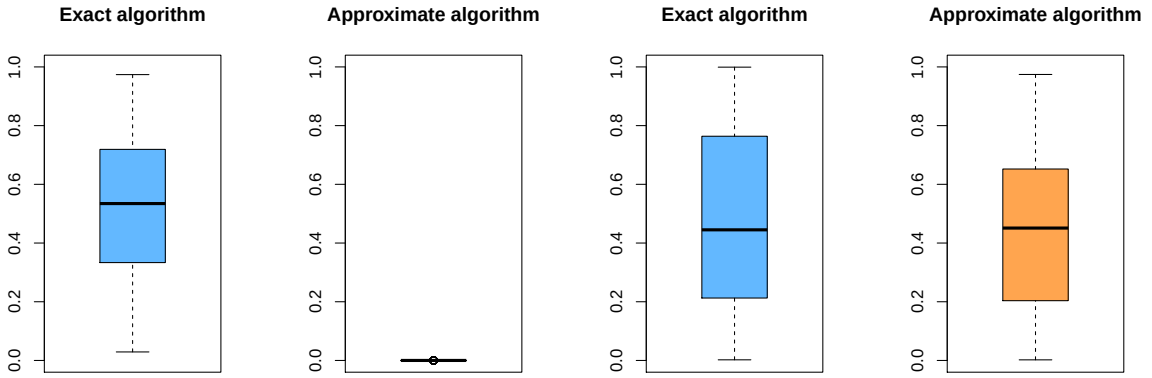
Using Algorithm 3, we could generate the following results (the whole following will use $\lambda = 3$ and $p = 10^4$).

	Exact	**Approximate**
$N = 10$	0.5305	$3.821 \cdot 10^{-6}$
$N = 10^2$	0.4727	0.4399
$N = 10^3$	0.5026	0.4949
$N = 10^4$	0.5069	0.4213

Table 1: Mean p-values from KS tests obtained from comparing both sampling algorithms (exact and approximate) to real cdf F_{M_N}, for $N = 10, 10^2, 10^3, 10^4$

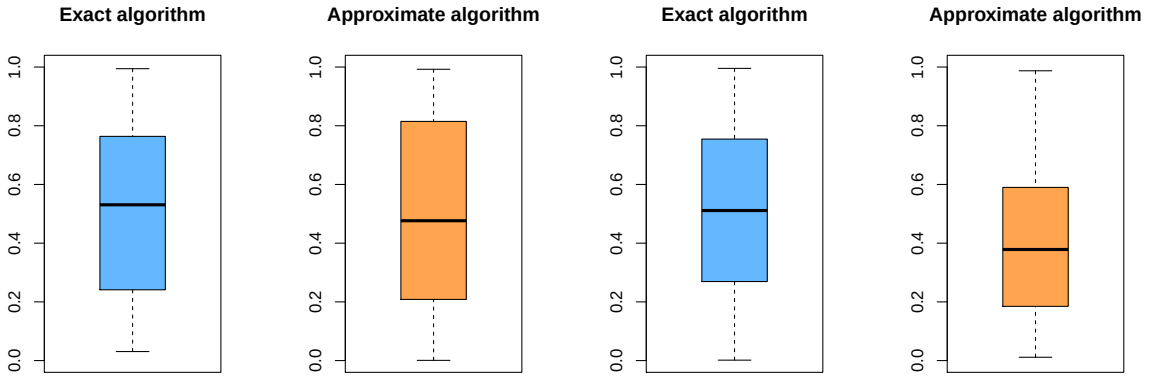
Table 1 shows that for each value of N, the exact sampling Algorithm 1 provides samples that come from real M_N distribution: in fact, as all p-values are greater than the threshold 0.05, we fail to reject the null hypothesis. Therefore, we may think that Algorithm 1 has been correctly

designed and implemented. However and as expected, this is not the case for Algorithm 2: with small values of N (especially here $N = 10$), this algorithm doesn't provide good samples of M_N as the p-value is less than 0.05, and therefore we reject the null hypothesis that samples have the same distribution as M_N. However, as N increases, Algorithm 2 seems more and more efficient as the obtained p-values for $N = 10^2, 10^3, 10^4$ are greater than 0.05: thus, we fail to reject the null hypothesis. This is consistent since we have shown in (c) that the approximation we used for F_{M_N} is true when $N \to \infty$, this is why Algorithm 2 seems better and better as N increases (even if the mean p-value obtained for $N = 10^4$ is lower than for $N = 10^3$, it is still high enough to accept H_0).



(a) p-values, $N = 10$

(b) p-values, $N = 10^2$

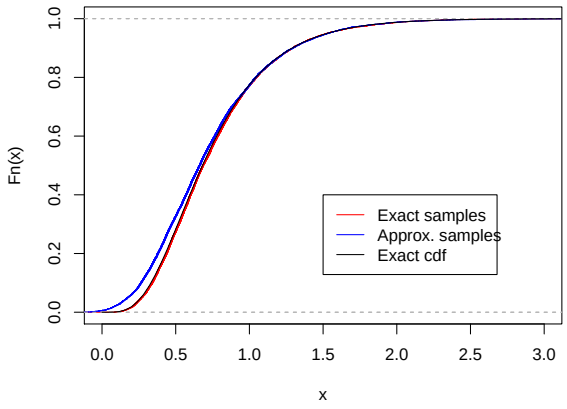


(c) p-values, $N = 10^3$

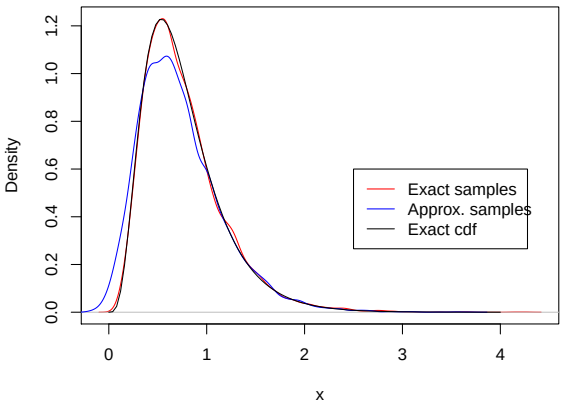
(d) p-values, $N = 10^4$

Figure 4.2: Box plots of obtained p-values for each iteration (from 1 to 100), for $N = 10, 10^2, 10^3, 10^4$

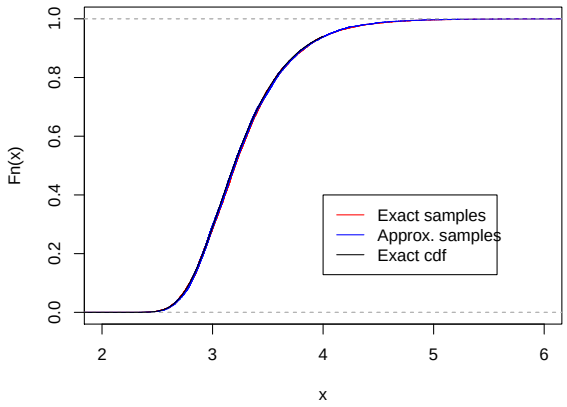
Figure 4.2 shows the distribution of obtained p-values as box plots for all chosen N. As N increases, the accuracy of generating process used in Algorithm 2 seems to increase, whereas the accuracy of Algorithm 1 is always good enough (in general) to consider this algorithm as exact. We may notice that even if Algorithm 2 is only approximate, it still converges with acceptable p-values at around $N = 10^2$.



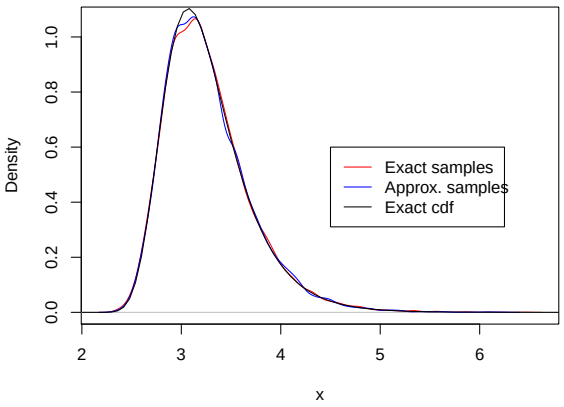
(a) Distribution functions, $N = 5$



(b) Density functions, $N = 5$



(c) Distribution functions, $N = 10^4$



(d) Density functions, $N = 10^4$

Figure 4.3: Densities and distributions for $N = 5$ and $N = 10^4$ obtained empirically from Algorithms 1 and 2, compared to the real ones

Furthermore, Figure 4.3 shows that whatever value of N, Algorithm 1 should generate samples that are coming from M_N distribution as the difference between the real density and cumulative distribution functions and its empirical density is almost zero (red and black curves are similar). However, this is not true for Algorithm 2 whose distribution and density for $N = 5$ are not exactly the same as the theoretical ones. Moreover, as N increases (here $N = 10^4$), we can state that Algorithm 2 generates samples from almost same distribution as M_N.

To study more in deep the convergence of Algorithm 2 samples to the real distribution, we may focus on "small" values of N and look at the mean p-values obtained from distributions with repeated sampling.

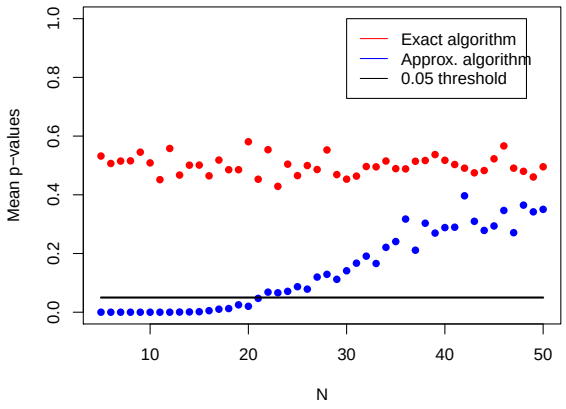


Figure 4.4: Evolution of mean p-values of KS tests on samples distribution (H_0: the samples have the same distribution as M_N) versus N for both algorithms

Figure 4.4 represents the mean p-values obtained from KS tests after sampling from both algorithms multiple times (50 times). It can then be inferred that, given a 0.05 p-value threshold, we may consider that for $N > 25$, Algorithm 2 generates accurate samples from M_N distribution as the mean p-values obtained are greater than the threshold. This plot also shows clearly that as N increases, the quality of sampling from Algorithm 2 is getting better and better, whereas the quality of sampling from Algorithm 1 remains constant and "perfect". □

References

[1] Dr Andrew. Duncan (October 2021) *MATH70082 - Probability for Statistics 2021-2022 Lecture Notes*, Imperial College London MSc Statistics resources

Appendices

```
FMn <- function(x, lambda, N) (1-exp(-lambda*x))^N
fMn <- function(x, lambda, N) N*lambda*exp(-lambda*x)*(1-exp(-lambda*x))^(N-1)

x <- seq(0, 10, by=0.01)

plot(x, FMn(x, lambda=1, N=100), type="l", col="orange", lwd=2, xlab=expression(x),
     ylab=expression(F[M[N]]))
lines(x, FMn(x, lambda=2, N=100), col="purple", lwd=2)
lines(x, FMn(x, lambda=5, N=100), col="lightblue", lwd=2)
legend(7, 0.3, c(expression(lambda == 1), expression(lambda == 2),
                 expression(lambda == 5)), col=c("orange", "purple", "lightblue"),
       lty=1:1)

plot(x, fMn(x, lambda=1, N=100), type="l", col="orange", lwd=2, xlab=expression(x),
     ylab=expression(f[M[N]]), ylim=c(0, 1.9))
lines(x, fMn(x, lambda=2, N=100), col="purple", lwd=2)
lines(x, fMn(x, lambda=5, N=100), col="lightblue", lwd=2)
legend(7, 1.8, c(expression(lambda == 1), expression(lambda == 2),
                 expression(lambda == 5)), col=c("orange", "purple", "lightblue"),
       lty=1:1)

FMn <- function(x, lambda, N) (1-exp(-lambda*x))^N
fMn <- function(x, lambda, N) N*lambda*exp(-lambda*x)*(1-exp(-lambda*x))^(N-1)

x <- seq(0, 10, by=0.01)

plot(x, FMn(x, lambda=1, N=10), type="l", col="orange", lwd=2, xlab=expression(x),
     ylab=expression(F[M[N]]))
lines(x, FMn(x, lambda=2, N=10), col="purple", lwd=2)
lines(x, FMn(x, lambda=5, N=10), col="lightblue", lwd=2)
legend(7, 0.3, c(expression(lambda == 1), expression(lambda == 2),
                 expression(lambda == 5)), col=c("orange", "purple", "lightblue"),
       lty=1:1)

plot(x, fMn(x, lambda=1, N=10), type="l", col="orange", lwd=2, xlab=expression(x),
     ylab=expression(f[M[N]]), ylim=c(0, 1.9))
lines(x, fMn(x, lambda=2, N=10), col="purple", lwd=2)
lines(x, fMn(x, lambda=5, N=10), col="lightblue", lwd=2)
legend(7, 1.8, c(expression(lambda == 1), expression(lambda == 2),
                 expression(lambda == 5)), col=c("orange", "purple", "lightblue"),
       lty=1:1)


library(pbapply)

set.seed(4567)
N <- 20
p <- 1e4
lambda <- 3
F <- function(x) (1-exp(-lambda*x))^N
f <- function(x) N*lambda*exp(-lambda*x)*(1-exp(-lambda*x))^(N-1)

P <- c()
for(N in 5:50){
  F <- function(x) (1-exp(-lambda*x))^N
  f <- function(x) N*lambda*exp(-lambda*x)*(1-exp(-lambda*x))^(N-1)
```

```
  Ps <- c()
  for(i in 1:50){
    U <- runif(p)

    # Exact but inefficient
    allMn.exact <- pbreplicate(p, max(-1/lambda * log(runif(N))))

    # Approximate but efficient
    f <- function(u) -1/lambda * log(-log(u)/N)
    allMn.approx <- f(U)

    Ps <- rbind(Ps, c(ks.test(allMn.exact, "F")$p.value,
                      ks.test(allMn.approx, "F")$p.value))
  }
  P <- rbind(P, c(mean(Ps[,1]), mean(Ps[,2])))
}
plot(5:50, P[,1], col="red", ylim=c(0,1), pch=16, xlab="N", ylab="Mean p-values")
points(5:50, P[,2], col="blue", pch=16)
lines(5:50, rep(0.05, 46), lwd=2)
legend(30, 1, c("Exact algorithm", "Approx. algorithm", "0.05 threshold"),
       col=c("red", "blue", "black"), lty=1:1)


Ps <- c()
for(i in 1:50){
  U <- runif(p)

  # Exact but inefficient
  allMn.exact <- pbreplicate(p, max(-1/lambda * log(runif(N))))

  # Approximate but efficient
  f <- function(u) -1/lambda * log(-log(u)/N)
  allMn.approx <- f(U)

  Ps <- rbind(Ps, c(ks.test(allMn.exact, "F")$p.value,
                    ks.test(allMn.approx, "F")$p.value))
}

mean(Ps[,1])
mean(Ps[,2])

par(mfrow=c(1,2))
boxplot(Ps[,1], ylim=c(0,1), main="Exact algorithm", col="steelblue1")
boxplot(Ps[,2], ylim=c(0,1), main="Approximate algorithm", col="tan1")

allMn.exact <- pbreplicate(p, max(-1/lambda * log(runif(N))))
f <- function(u) -1/lambda * log(-log(u)/N)
allMn.approx <- f(U)

par(mfrow=c(1,1))
plot(ecdf(allMn.exact), col="red", main="", xlim=c(2,6))
plot(ecdf(allMn.approx), add=TRUE, col="blue")
curve(F, 0, 4, col="black", add=TRUE)
legend(4, 0.4, c("Exact samples", "Approx. samples", "Exact cdf"),
       col=c("red", "blue", "black"), lty=1:1)


par(mfrow=c(1,1))
plot(density(allMn.exact), col="red", main="", xlab="x")
```

```
lines(density(allMn.approx), col="blue")
curve(f, 0, 4, col="black", add=TRUE)
legend(2.5, 0.6, c("Exact samples", "Approx. samples", "Exact cdf"),
       col=c("red", "blue", "black"), lty=1:1)
```