These are exercises for self-study and exam preparation. All material is examinable unless otherwise mentioned.

Exercise 1. ***Importance sampling to estimate tail probabilities***

We would like to use importance sampling to compute the probability that a standard Gaussian random variable x takes on a value larger than 5, i.e

$$\mathbb{P}(x > 5) = \int_5^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \tag{1}$$

We know that the probability equals

$$\begin{aligned}\mathbb{P}(x > 5) &= 1 - \int_{-\infty}^5 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx && (2)\\ &= 1 - \Phi(5) && (3)\\ &\approx 2.87 \cdot 10^{-7} && (4)\end{aligned}$$

where $\Phi(.)$ is the cumulative distribution function of a standard normal random variable.[1]

(a) With the indicator function $\mathbb{1}_{x>5}(x)$, which equals one if x is larger than 5 and zero otherwise, we can write $\mathbb{P}(x > 5)$ in form of the expectation

$$\mathbb{P}(x > 5) = \mathbb{E}[\mathbb{1}_{x>5}(x)], \tag{5}$$

where the expectation is taken with respect to the density $\mathcal{N}(x; 0, 1)$ of a standard normal random variable,

$$\mathcal{N}(x; 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right). \tag{6}$$

This suggests that we can approximate $\mathbb{P}(x > 5)$ by a Monte Carlo average

$$\mathbb{P}(x > 5) \approx \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{x>5}(x_i), \qquad x_i \sim \mathcal{N}(x; 0, 1). \tag{7}$$

Explain why this approach does not work well.

(b) Another approach is to use importance sampling with an importance distribution $q(x)$ that is zero for $x < 5$. We can then write $\mathbb{P}(x > 5)$ as

$$\begin{aligned}\mathbb{P}(x > 5) &= \int_5^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx && (8)\\ &= \int_5^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \frac{q(x)}{q(x)} dx && (9)\\ &= \mathbb{E}_{q(x)}\left[\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \frac{1}{q(x)}\right] && (10)\end{aligned}$$

[1] Credit: This exercise is based on example and exercise 3.5 in Robert and Casella's *Introducing Monte Carlo Methods with R*, Springer 2010.

©Gutmann, University of Edinburgh, 2026 CC BY 4.0

and estimate $\mathbb{P}(x > 5)$ as a sample average.

We here use an exponential distribution shifted by 5 to the right. It has pdf

$$q(x) = \begin{cases} \exp(-(x-5)) & \text{if } x \geq 5 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

For background on the exponential distribution, see e.g. https://en.wikipedia.org/wiki/Exponential_distribution.

Provide a formula that approximates $\mathbb{P}(x > 5)$ as a sample average over n samples $x_i \sim q(x)$.

(c) Numerically compute the importance estimate for various sample sizes $n = 1, \ldots, 1000$. Plot the estimate against the sample size and compare with the ground truth value.

Exercise 2. ***Sampling from the exponential distribution***

The exponential distribution has the density

$$p(x; \lambda) = \begin{cases} \lambda \exp(-\lambda x) & x \geq 0 \\ 0 & x < 0, \end{cases} \tag{12}$$

where λ is a parameter of the distribution. Use inverse transform sampling to generate n iid samples from $p(x; \lambda)$.

Exercise 3. ***Sampling from a Laplace distribution***

A Laplace random variable x of mean zero and variance one has the density $p(x)$

$$p(x) = \frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}|x|\right) \qquad x \in \mathbb{R}. \tag{13}$$

Use inverse transform sampling to generate n iid samples from x.

Exercise 4. ***Rejection sampling***

Most compute environments provide functions to sample from a standard normal distribution. Popular algorithms include the Box-Muller transform, see e.g. https://en.wikipedia.org/wiki/Box-Muller_transform. We here use rejection sampling to sample from a standard normal distribution with density $p(x)$ using a Laplace distribution as our proposal/auxiliary distribution.[2]

The density $q(x)$ of a zero-mean Laplace distribution with variance $2b^2$ is

$$q(x; b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right). \tag{14}$$

We can sample from it by sampling a Laplace variable with variance 1 as in Exercise 3 and then scaling the sample by $\sqrt{2}b$.

Rejection sampling then repeats the following steps:

[2] Credit: This exercise is loosely based on Exercise 2.8 in Robert and Casella's *Introducing Monte Carlo Methods with R*, Springer 2010.

©Gutmann, University of Edinburgh, 2026 CC BY 4.0

- Generate $x \sim q(x; b)$
- Accept x with probability $f(x) = \frac{1}{M}\frac{p(x)}{q(x)}$, i.e. generate $u \sim U(0,1)$ and accept x if $u \leq f(x)$.

(a) Compute the ratio $M(b) = \max_x \frac{p(x)}{q(x;b)}$.

(b) How should you choose b to maximise the probability of acceptance?

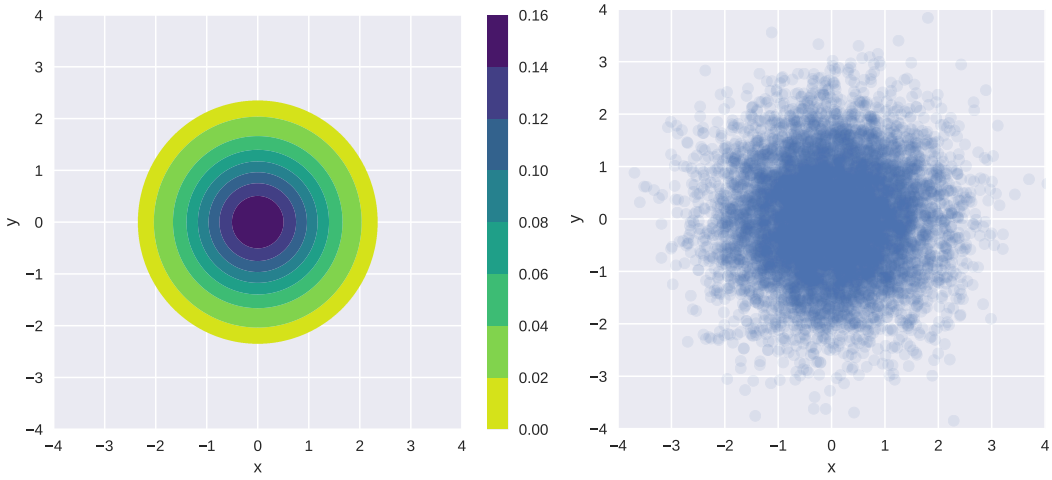
(c) Assume you sample from $p(x_1, \ldots, x_d) = \prod_{i=1}^d p(x_i)$ using $q(x_1, \ldots, x_d) = \prod_{i=1}^d q(x_i; b)$ as auxiliary distribution without exploiting any independencies. How does the acceptance probability scale as a function of d? You may denote the acceptance probability in case of $d = 1$ by A.

Exercise 5. ***Basic Markov chain Monte Carlo inference (optional, not examinable)***

This exercise is on sampling and approximate inference by Markov chain Monte Carlo (MCMC). MCMC can be used to obtain samples from a probability distribution, e.g. a posterior distribution. The samples approximately represent the distribution, as illustrated in Figure 1, and can be used to approximate expectations.

We denote the density of a Gaussian with mean μ and variance σ^2 by $\mathcal{N}(x; \mu, \sigma^2)$, i.e.

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{15}$$



(a) True density

(b) Density represented by $10,000$ samples.

Figure 1: Density and samples from $p(x, y) = \mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$.

Consider a vector of d random variables $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ and some observed data $\mathcal{D}$. In many cases, we are interested in computing expectations under the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$, e.g.

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\left[g(\boldsymbol{\theta})\right] = \int g(\boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) \mathrm{d}\boldsymbol{\theta} \tag{16}$$

for some function $g(\boldsymbol{\theta})$. If d is small, e.g. $d \leq 3$, deterministic numerical methods can be used to approximate the integral to high accuracy.[3] But for higher dimensions, these methods are

[3] See e.g. https://en.wikipedia.org/wiki/Numerical_integration.

©Gutmann, University of Edinburgh, 2026 CC BY 4.0

generally not applicable any more. The expectation, however, can be approximated as a sample average if we have samples $\boldsymbol{\theta}^{(i)}$ from $p(\boldsymbol{\theta} \mid \mathcal{D})$:

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\left[g(\boldsymbol{\theta})\right] \approx \frac{1}{S}\sum_{i=1}^{S} g(\boldsymbol{\theta}^{(i)}) \tag{17}$$

Note that in MCMC methods, the samples $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(S)}$ used in the above approximation are typically not statistically independent.

Metropolis-Hastings is an MCMC algorithm that generates samples from a distribution $p(\boldsymbol{\theta})$, where $p(\boldsymbol{\theta})$ can be any distribution on the parameters (and not only posteriors). The algorithm is iterative and at iteration t, it uses:

- a proposal distribution $q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$, parametrised by the current state of the Markov chain, i.e. $\boldsymbol{\theta}^{(t)}$;
- a function $p^*(\boldsymbol{\theta})$, which is proportional to $p(\boldsymbol{\theta})$. In other words, $p^*(\boldsymbol{\theta})$ is unnormalised[4] and the normalised density $p(\boldsymbol{\theta})$ is

$$p(\boldsymbol{\theta}) = \frac{p^*(\boldsymbol{\theta})}{\int p^*(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}. \tag{18}$$

For all tasks in this exercise, we work with a Gaussian proposal distribution $q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$, whose mean is the previous sample in the Markov chain, and whose variance is ϵ^2. That is, at iteration t of our Metropolis-Hastings algorithm,

$$q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}) = \prod_{k=1}^{d} \mathcal{N}(\theta_k; \theta_k^{(t-1)}, \epsilon^2). \tag{19}$$

When used with this proposal distribution, the algorithm is called Random Walk Metropolis-Hastings algorithm.

(a) Read Section 27.4 in Barber's book "Bayesian Reasoning and Machine Learning" to familiarise yourself with the Metropolis-Hastings algorithm.

(b) Write a function `mh` implementing the Metropolis Hasting algorithm, as given in Algorithm 27.3 in Barber's book, using the Gaussian proposal distribution in (19) above. The function should take as arguments

- `p_star`: the function $p^*(\boldsymbol{\theta})$, which is proportional to the density of interest $p(\boldsymbol{\theta})$;
- `param_init`: the initial sample — a value for $\boldsymbol{\theta}$ from where the Markov chain starts;
- `num_samples`: the number S of samples to generate;
- `vari`: the variance ϵ^2 for the Gaussian proposal distribution q;

and return $[\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(S)}]$ — a list of S samples from $p(\boldsymbol{\theta}) \propto p^*(\boldsymbol{\theta})$. For example:

[4] We used the notation $\tilde{p}$ in the lecture slides; p^* is also commonly used, e.g. in Barber's book.

©Gutmann, University of Edinburgh, 2026 CC BY 4.0

```python
def mh(p_star, param_init, num_samples=5000, vari=1.0):
    # your code here
    return samples
```

(c) Test your algorithm by sampling $5,000$ samples from $p(x, y) = \mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$. Initialise at $(x = 0, y = 0)$ and use $\epsilon^2 = 1$. Generate a scatter plot of the obtained samples. The plot should be similar to Figure 1b. Highlight the first 20 samples only. Do these 20 samples alone adequately approximate the true density?

Sample another $5,000$ points from $p(x, y) = \mathcal{N}(x; 0, 1)\mathcal{N}(y; 0, 1)$ using mh with $\epsilon^2 = 1$, but this time initialise at $(x = 7, y = 7)$. Generate a scatter plot of the drawn samples and highlight the first 20 samples. If everything went as expected, your plot probably shows a "trail" of samples, starting at $(x = 7, y = 7)$ and slowly approaching the region of space where most of the probability mass is.

(d) In practice, we don't know where the distribution we wish to sample from has high density, so we typically initialise the Markov Chain somewhat arbitrarily, or at the maximum a posteriori (MAP) estimate if available. The samples obtained in the beginning of the chain are typically discarded, as they are not considered to be representative of the target distribution. This initial period between initialisation and starting to collect samples is called "warm-up", or also "burn-in".

Extend your function mh to include an additional warm-up argument W, which specifies the number of MCMC steps taken before starting to collect samples. Your function should still return a list of S samples as in (b).

Exercise 6. *Bayesian Poisson regression* (optional, not examinable)

Consider a Bayesian Poisson regression model, where outputs y_n are generated from a Poisson distribution of rate $\exp(\alpha x_n + \beta)$, where the x_n are the inputs (covariates), and α and β the parameters of the regression model for which we assume a broad Gaussian prior:

$$\alpha \sim \mathcal{N}(\alpha; 0, 100) \tag{20}$$
$$\beta \sim \mathcal{N}(\beta; 0, 100) \tag{21}$$
$$y_n \sim \text{Poisson}(y_n; \exp(\alpha x_n + \beta)) \quad \text{for } n = 1, \ldots, N \tag{22}$$

$\text{Poisson}(y; \lambda)$ denotes the probability mass function of a Poisson random variable with rate λ,

$$\text{Poisson}(y; \lambda) = \frac{\lambda^y}{y!}\exp(-\lambda), \qquad y \in \{0, 1, 2, \ldots\}, \quad \lambda > 0 \tag{23}$$

Consider $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ where $N = 5$ and

$$(x_1, \ldots, x_5) = (-0.50519053, -0.17185719, 0.16147614, 0.49480947, 0.81509851) \tag{24}$$
$$(y_1, \ldots, y_5) = (1, 0, 2, 1, 2) \tag{25}$$

We are interested in computing the posterior density of the parameters (α, β) given the data $\mathcal{D}$ above.

(a) Derive an expression for the unnormalised posterior density of α and β given $\mathcal{D}$, i.e. a function p^* of the parameters α and β that is proportional to the posterior density $p(\alpha, \beta \mid \mathcal{D})$, and which can thus be used as target density in the Metropolis Hastings algorithm.

©Gutmann, University of Edinburgh, 2026 CC BY 4.0

(b) Implement the derived unnormalised posterior density p^*. If your coding environment provides an implementation of the above Poisson pmf, you may use it directly rather than implementing the pmf yourself.

Use the Metropolis Hastings algorithm from Question 5(c) to draw $5,000$ samples from the posterior density $p(\alpha, \beta \mid \mathcal{D})$. Set the hyperparameters of the Metropolis-Hastings algorithm to:

- $\mathsf{param_init} = (\alpha_{\text{init}}, \beta_{\text{init}}) = (0, 0)$,
- $\mathsf{vari} = 1$, and
- number of warm-up steps $W = 1000$.

Plot the drawn samples with x-axis α and y-axis β and report the posterior mean of α and β, as well as their correlation coefficient under the posterior.

Exercise 7. ***Mixing and convergence of Metropolis-Hasting MCMC (optional, not examinable)***

Under weak conditions, an MCMC algorithm is an asymptotically exact inference algorithm, meaning that if it is run forever, it will generate samples that correspond to the desired probability distribution. In this case, the chain is said to converge.

In practice, we want to run the algorithm long enough to be able to approximate the posterior adequately. How long is long enough for the chain to converge varies drastically depending on the algorithm, the hyperparameters (e.g. the variance vari), and the target posterior distribution. It is impossible to determine exactly whether the chain has run long enough, but there exist various diagnostics that can help us determine if we can "trust" the sample-based approximation to the posterior.

A very quick and common way of assessing convergence of the Markov chain is to visually inspect the *trace plots* for each parameter. A trace plot shows how the drawn samples evolve through time, i.e. they are a time-series of the samples generated by the Markov chain. Figure 2 shows examples of trace plots obtained by running the Metropolis Hastings algorithm for different values of the hyperparameters vari and param_init. Ideally, the time series covers the whole domain of the target distribution and it is hard to "see" any structure in it so that predicting values of future samples from the current one is difficult. If so, the samples are approximately uncorrelated and the chain is said to be well "mixed".

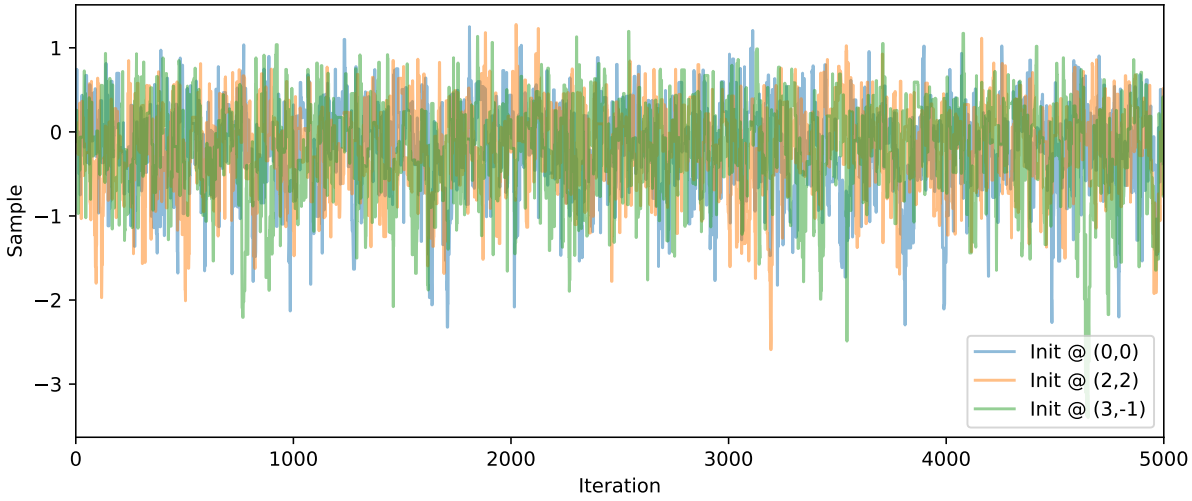
(a) Consider the trace plots in Figure 2: Is the variance vari used in Figure 2b larger or smaller than the value of vari used in Figure 2a? Is vari used in Figure 2c larger or smaller than the value used in Figure 2a?

In both cases, explain the behaviour of the trace plots in terms of the workings of the Metropolis Hastings algorithm and the effect of the variance vari.

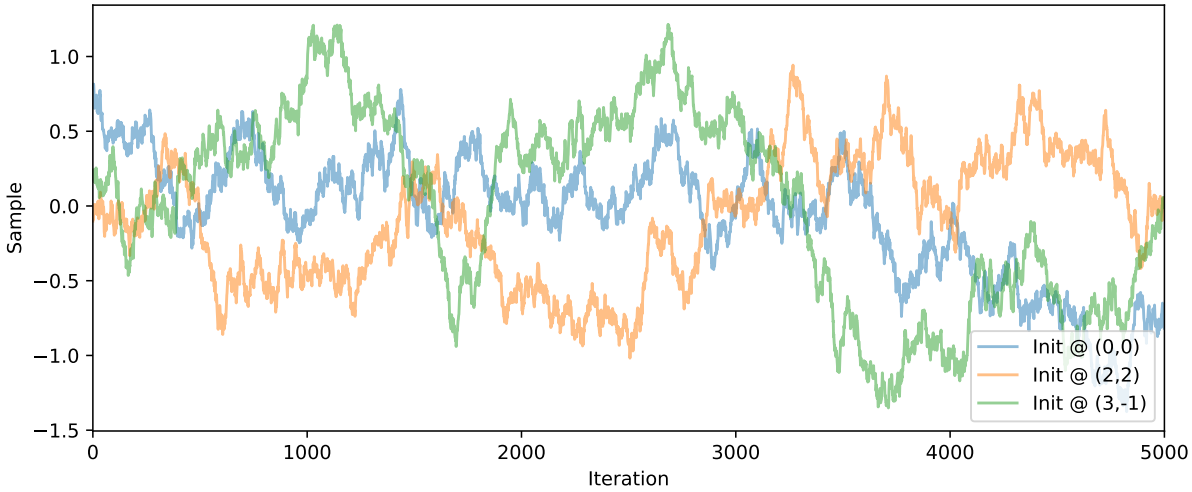
(b) In Metropolis-Hastings, and MCMC in general, any sample depends on the previously generated sample, and hence the algorithm generates samples that are generally statistically dependent. The *effective sample size* of a sequence of dependent samples is the number of independent samples that are, in some sense, equivalent to our number of dependent samples. A definition of the effective sample size (ESS) is

$$\text{ESS} = \frac{S}{1 + 2\sum_{k=1}^{\infty} \rho(k)} \tag{26}$$

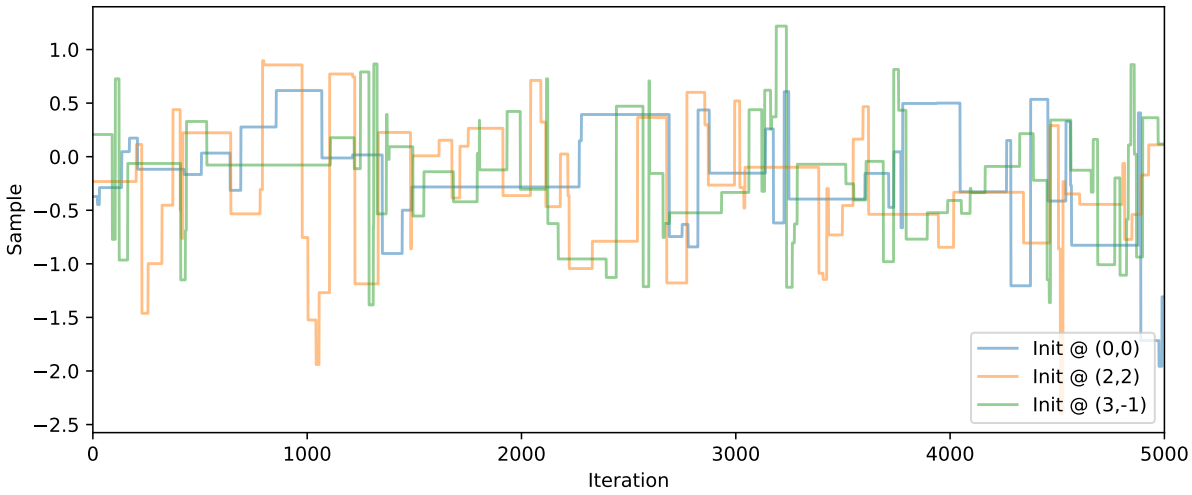
©Gutmann, University of Edinburgh, 2026 CC BY 4.0



(a) variance vari: 1



(b) Alternative value of vari



(c) Alternative value of vari

Figure 2: For Question 7(a): Trace plots of the parameter β from Question 6 drawn using Metropolis-Hastings with different variances of the proposal distribution.

©Gutmann, University of Edinburgh, 2026 CC BY 4.0

where S is the number of dependent samples drawn and $\rho(k)$ the correlation coefficient between two samples in the Markov chain that are k time points apart. We can see that if the samples are strongly correlated, $\sum_{k=1}^{\infty} \rho(k)$ is large and the effective sample size is small. On the other hand, if $\rho(k) = 0$ for all k, the effective sample size is S.

ESS, as defined above, is the number of independent samples which are needed to obtain a sample average that has the same variance as the sample average computed from correlated samples. Note that this is a different formula than the effective sample size for importance sampling in the lecture, $(\sum_i w_i)^2 / \sum_i w_i^2$, which measures the loss due to uneven weights rather than due to correlated samples. However, both can be interpreted in terms of an equivalent number of independent samples.

To illustrate how correlation between samples is related to a reduction of sample size, consider two pairs of samples (θ_1, θ_2) and (ω_1, ω_2). All variables have variance σ^2 and the same mean μ, but ω_1 and ω_2 are uncorrelated while the covariance matrix for θ_1, θ_2 is $\mathbf{C}$,

$$\mathbf{C} = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \tag{27}$$

with $\rho > 0$. The variance of the average $\bar{\omega} = 0.5(\omega_1 + \omega_2)$ is

$$\mathbb{V}(\bar{\omega}) = \frac{\sigma^2}{2}, \tag{28}$$

where the 2 in the denominator is the sample size.

Derive an equation for the variance of $\bar{\theta} = 0.5(\theta_1 + \theta_2)$ and compute the reduction α of the sample size when working with the correlated (θ_1, θ_2). In other words, derive an equation of α in

$$\mathbb{V}\left(\bar{\theta}\right) = \frac{\sigma^2}{2/\alpha}. \tag{29}$$

What is the effective sample size $2/\alpha$ as $\rho \to 1$?

©Gutmann, University of Edinburgh, 2026 CC BY 4.0