Assignment 3: Approximation via Monte Carlo (worth 6 points)
Human and Machine Learning
Chiba Institute of Technology, School of Design & Science
Prof. Joseph Austerweil
Due Fri Jul 10, 2026 at 8:00pm

In this problem set, you will explore how to use different Monte Carlo methods. You will (1) compare naïve Monte Carlo with importance sampling, (2) build a Markov chain Monte Carlo (MCMC) sampler — interleaving Gibbs and Metropolis–Hastings steps — for a hierarchical Bayesian model, and (3) quantify how "efficient" a set of samples is using the effective sample size.

1. (Monte Carlo). In this problem, you will compare two Monte Carlo methods for estimating the probability that a standard normal distribution has a value greater than two (i.e., $P(Y > 2|Y \sim N(0,1))$). For all parts of problem 1, we will use $T = 1000$.

   (a) The first method we will use is naïve Monte Carlo. Monte Carlo estimates the expected value of a function of a random variable. The probability a random variable is greater than two can be written as an expectation of a function of a random variable — it is the expected number of times that the random variable is greater than two if we drew a lot of samples from the distribution. So, $P(Y > 2) = E[I(Y > 2)]$, where $I(\cdot)$ is an indicator function, which returns 1 when its argument is true and 0 otherwise.[1]
   Write code that samples T standard normally distributed variables $x_{MC}^{(1)}, \ldots, x_{MC}^{(T)} \sim N(0,1)$. Create a vector which cumulatively estimates the Monte Carlo estimate of the probability of seeing a value greater than two after observing each subsequent data point. In other words, create a vector $\mathbf{f}_{MC}$ such that the value of element t in $\mathbf{f}_{MC}$ is the Monte Carlo estimate of the probability that a standard normal randomly generated value is greater than 2 from t samples (the proportion of samples in $x_{MC}^{(1)}, \ldots, x_{MC}^{(t)}$ that are greater than 2). Plot the $\mathbf{f}_{MC}$ vector. Is there anything strange about it? (Does it look like the example we did in class of the Monte Carlo estimate of the average number of dots seen when you roll a die?) Write 1-2 sentences explaining why the plot of the updated Monte Carlo estimate looks the way it does. Also report the final estimate according to the Monte Carlo estimate ($f_{MC}(T)$).

   (b) The second method we will use is called *importance sampling*. This is another method for estimating the expectation of a function of a random variable. Rather than using the distribution of the random variable itself to generate samples, it uses a different distribution (q – called the *proposal* distribution)[2] and then re-weights each sample by how probable they would have been under the true distribution. So, if $x_{IS}^{(1)}, \ldots, x_{IS}^{(T)}$ are my T samples generated from distribution q rather than p, my estimate would be

   $$\frac{1}{T}\sum_{t=1}^{T}\frac{p\left(x_{IS}^{(t)}\right)}{q\left(x_{IS}^{(t)}\right)}I\left(x_{IS}^{(t)} > 2\right)$$

   Lets apply this to calculating the probability that $X > 2$ given $X \sim N(0,1)$. Because we are interested in the probability $X > 2$, lets use a distribution that is closer to 2:

[1] In your favorite programming language, it is easy to calculate the probability that a standard normal distribution is greater than two. However, I want you to implement these estimators to get a feel for how they behave.

[2] If you are curious why we use q instead of p for the probability under the proposal distribution, at least one reason is to remind us that q is not the actual probability. Rather, it is a surrogate that we are using as part of our approximation method.

$q = N(2, 1)$. So, samples from the proposal distribution come from $N(2, 1)$ (a normal distribution with a mean of 2 and variance of 1). This can be drawn from a standard normal distribution ($N(0, 1)$) by adding two to a random variable generated from a standard normal distribution.

One way to solve the problem then is to sample each data point from q and then use a function that returns the probability density of some value under a Normal distribution with a specified mean and variance. Alternatively, you could divide p by q explicitly. The explicit solution is

$$\frac{1}{T}\sum_{t=1}^{T} e^{-2x_{IS}^{(t)}+2} I(x_{IS}^{(t)} > 2)$$

Now, make the same plot as in 1a, but now using the importance sampling estimate (i.e., plot a vector $\mathbf{f}^{IS}$ such that f_t^{IS} is the importance sampling estimate of the probability that a standard normal randomly generated variable is greater than 2 using $q \sim N(2, 1)$ after t samples). Write 1-2 sentences explaining why the plot of the updated importance sampling estimate looks the way it does. Also, please report the importance sampling estimate given all of the samples.

(c) Write a few sentences comparing Figures 1a and 1b: How and why do they differ? What is a broad lesson about Monte Carlo approximation that we can learn from this comparison?

2. (Markov chain Monte Carlo). For this problem, you will be implementing a hierarchical Bayesian model and using Markov chain Monte Carlo (interleaving Gibbs sampling and Metropolis–Hastings) to approximate its predictions. In particular, you will implement the Beta-binomial model used in Kemp et al. (2007)[3] to understand how people learn *overhypotheses* — beliefs about how properties tend to be distributed over different situations. So, the Beta distribution is prominent in their model, which is a probability distribution over values between 0 and 1.

The mean/concentration parameterization. A Beta distribution is usually written $\text{Beta}(a, b)$ with two shape parameters. In this assignment (following the lecture) we will parameterize it instead by its *mean* φ and *concentration* κ, where

$$\varphi = \frac{a}{a+b} \in (0, 1), \qquad \kappa = a + b > 0, \qquad \text{equivalently} \qquad a = \kappa\varphi, \quad b = \kappa(1-\varphi).$$

Intuitively, φ is where the distribution is centered (the prior proportion of white marbles) and κ is how tightly it concentrates around that mean (large $\kappa \Rightarrow$ all bags look alike; small $\kappa \Rightarrow$ bags are pushed toward 0 or 1).[4] The generative model is

$$\begin{aligned} \kappa &\sim p(\kappa) & \varphi &\sim \text{Uniform}(0, 1) \\ \theta_i \mid \kappa, \varphi &\sim \text{Beta}\left(\kappa\varphi,\ \kappa(1-\varphi)\right) & y_i \mid n_i, \theta_i &\sim \text{Binomial}(\theta_i; n_i) \end{aligned}$$

where $p(\kappa)$ is a prior on the concentration that we specify in part (b).

[3] Note that Kemp et al. (2007) used the more general Dirichlet-multinomial model, which is essentially the same as a Beta-binomial model, but generalized to more than two choices.

[4] This is the same model as the (α, β) parameterization in Kemp et al. (2007) (their α is our κ and their β is our φ); we use (φ, κ) because the mean/concentration split makes the sampler below cleaner and matches the lecture.

Remember their cover story. Imagine that you come across a box filled with bags of marbles. Each bag has a lot of marbles. Each marble is either black or white. You get to observe n_i marbles from bag i and write down that y_i of them are white. The proportion of white marbles in bag i is θ_i (and because there are lots and lots of marbles in each bag, we can safely assume that y_i is Binomial distributed with parameters θ_i and n_i).

(a) First, you will estimate the posterior distribution of a simplified version of the model: the posterior probability of the proportion of white marbles after seeing some marbles from a single bag, with κ and φ *fixed* (no hierarchical prior over them yet). So, the generative process for this part is: $\theta_i \mid \kappa, \varphi \sim \text{Beta}\left(\kappa\varphi, \kappa(1-\varphi)\right)$ and $y_i \mid n_i, \theta_i \sim \text{Binomial}(\theta_i; n_i)$. Because the Beta is conjugate to the Binomial (as you can look up on the Conjugate Prior page of Wikipedia, and as we derived in class), the posterior distribution is

$$\theta_i \mid \kappa, \varphi, y_i, n_i \;\sim\; \text{Beta}\left(\kappa\varphi + y_i,\; \kappa(1-\varphi) + (n_i - y_i)\right).$$

This is the equation you need to complete this task. For each (κ, φ) pair below, please graph, all on the same plot: the prior distribution (i.e., no marbles observed), the posterior after observing one white marble (and zero black), the posterior after observing 5 white and 5 black, and the posterior after observing 9 white and 1 black. The (a, b) values are given so you can check your conversion.

i. $\kappa = 0.1$ and $\varphi = 0.5$ $\quad (a = 0.05,\ b = 0.05)$
ii. $\kappa = 0.1$ and $\varphi = 0.9$ $\quad (a = 0.09,\ b = 0.01)$
iii. $\kappa = 1$ and $\varphi = 0.5$ $\quad (a = 0.5,\ b = 0.5)$
iv. $\kappa = 1$ and $\varphi = 0.9$ $\quad (a = 0.9,\ b = 0.1)$
v. $\kappa = 5$ and $\varphi = 0.5$ $\quad (a = 2.5,\ b = 2.5)$
vi. $\kappa = 5$ and $\varphi = 0.9$ $\quad (a = 4.5,\ b = 0.5)$

Although I expect you to execute and think about each of these, you are only required to turn in one of the graphs. Please make sure the graph is visible and each distribution is distinct and clearly labeled. Please write a paragraph describing how changing κ and φ affects the posterior update.

(b) Now you will build a sampler to estimate the posterior distribution over κ, φ, and $\theta_1, \ldots, \theta_M$ after observing some marbles from M bags (i.e., given $y_1, \ldots, y_M, n_1, \ldots, n_M$), and the predictive probability that a marble is white in a *new* bag. We approximate the posterior using two moves per sweep: a **Gibbs** step that resamples $\theta_1, \ldots, \theta_M$, and a **Metropolis–Hastings (MH)** step that updates κ and φ. The two moves are *separate* kernels — each on its own leaves the posterior invariant, so alternating them is valid — and the θ_i are held fixed (conditioned on) during the MH step, so they never enter the MH proposal.[5]

The prior on κ. We place a weak, *proper* prior on the concentration: $\log \kappa \sim N(\mu_0, \sigma_0^2)$ with $\mu_0 = \log 10$ and $\sigma_0 = 1.5$ (a log-normal prior on κ).[6] The prior on φ is Uniform$(0, 1)$.

[5] This is why the asymmetry of the Gibbs draw needs no correction in the MH step: the θ_i are not *proposed* by the MH move, they are part of the state it conditions on. (Gibbs is itself the special case of MH whose proposal is the exact conditional, so its own acceptance probability is always 1 — which is why a Gibbs step "always accepts" and needs no accept/reject test.)

[6] The lecture and Kemp et al. (2007) use an (improper) log-uniform / exponential prior on the concentration. In practice an improper prior lets κ run off to very large values when the bags happen to look alike, which makes the chain hard to tune and the results hard to reproduce. A weak *proper* prior is better-behaved and more robust, so we use one here; it is broad enough that the data still dominate.

The Metropolis–Hastings step on (φ, ℓ). As in lecture, we sample the concentration on the log scale — let $\ell = \log \kappa$ — so that a Gaussian step never proposes a negative concentration. We propose φ and ℓ with *symmetric* Gaussian random walks (you tune the two widths separately):

$$\varphi' = \varphi + \epsilon_\varphi, \quad \epsilon_\varphi \sim N(0, s_\varphi^2); \qquad \ell' = \ell + \epsilon_\ell, \quad \epsilon_\ell \sim N(0, s_\ell^2),$$

then map back to evaluate the model: $\kappa' = e^{\ell'}$, $a' = \kappa'\varphi'$, $b' = \kappa'(1-\varphi')$. The general Metropolis–Hastings acceptance probability has three factors — a likelihood ratio, a prior ratio, and a *proposal asymmetry correction* (the factor that, in full Hastings, fixes up a proposal that is more likely to step one way than the other):

$$c = \min\left(1,\ \underbrace{\frac{\prod_{i=1}^{M} \mathrm{Beta}(\theta_i;\, a',\, b')}{\prod_{i=1}^{M} \mathrm{Beta}(\theta_i;\, a,\, b)}}_{\text{likelihood of the current } \theta_i\text{'s}} \times \underbrace{\frac{N(\ell';\, \mu_0, \sigma_0^2)}{N(\ell;\, \mu_0, \sigma_0^2)}}_{\text{prior ratio on } \log\kappa} \times \underbrace{\frac{q(\varphi, \ell \mid \varphi', \ell')}{q(\varphi', \ell' \mid \varphi, \ell)}}_{\substack{\text{asymmetry correction} \\ = 1 \text{ (symmetric)}}}\right).$$

The proposal asymmetry correction is 1 because our Gaussian step is symmetric ($q(\varphi', \ell' \mid \varphi, \ell) = q(\varphi, \ell \mid \varphi', \ell')$) — so you never have to compute it. The φ prior is uniform on $(0, 1)$, so it does not appear (it only enforces $0 < \varphi' < 1$; if φ' falls outside, reject). The only nontrivial correction is the *prior ratio* on ℓ from our weak log-normal prior on κ.[7] Work in log space when you compute it: $\log c = \left[\sum_i \log \mathrm{Beta}(\theta_i; a', b') - \sum_i \log \mathrm{Beta}(\theta_i; a, b)\right] + \left[\log N(\ell'; \mu_0, \sigma_0^2) - \log N(\ell; \mu_0, \sigma_0^2)\right]$.

(c) **Implement the sampler.** Initialize $\varphi^{(1)}$ and $\kappa^{(1)}$ (e.g. $\varphi^{(1)} = 0.5$, $\kappa^{(1)} = 10$), initialize each θ_i, and then for $t = 1, \ldots, T$ perform one Gibbs sweep followed by one MH step:

I. **Gibbs sweep.** Resample each $\theta_i^{(t)}$ once (in a random order each sweep[8]), using the conjugate posterior from part (a):

$$\theta_i^{(t)} \mid \kappa^{(t)}, \varphi^{(t)}, y_i, n_i \sim \mathrm{Beta}\left(\kappa^{(t)}\varphi^{(t)} + y_i,\ \kappa^{(t)}(1-\varphi^{(t)}) + (n_i - y_i)\right).$$

II. **MH proposal.** Set $\ell = \log \kappa^{(t)}$, propose $\varphi' = \varphi^{(t)} + \epsilon_\varphi$ and $\ell' = \ell + \epsilon_\ell$ as above, and let $\kappa' = e^{\ell'}$, $a' = \kappa'\varphi'$, $b' = \kappa'(1-\varphi')$. If $\varphi' \notin (0, 1)$, reject immediately.

III. **Accept/reject.** Compute $\log c$ using the equation above (with the *current* $\theta_i^{(t)}$ from step I), and accept with probability $\min(1, e^{\log c})$: sample $Y \sim U(0, 1)$, and if $Y \le \min(1, e^{\log c})$ set $(\kappa^{(t+1)}, \varphi^{(t+1)}) = (\kappa', \varphi')$, otherwise keep $(\kappa^{(t+1)}, \varphi^{(t+1)}) = (\kappa^{(t)}, \varphi^{(t)})$.

[7] This is the one place the assignment differs from the lecture's Kemp slide, and it is worth understanding why. The lecture (i) puts a *flat* prior on $\ell = \log \kappa$, which makes the prior ratio 1, and (ii) *marginalizes out* the θ_i analytically, so its acceptance ratio uses the Beta-Binomial marginal likelihood $\prod_i \mathrm{BetaBin}(y_i \mid n_i, a, b)$ rather than $\prod_i \mathrm{Beta}(\theta_i; a, b)$ with the sampled θ_i. Here we instead *Gibbs-sample* the θ_i (so you implement and see *both* a Gibbs step and an MH step in one sampler — the point of the problem) and score the proposal against those current θ_i; and we use a weak *proper* prior on κ. Both samplers target the correct posterior. Note there is no change-of-variables Jacobian to worry about: we put the prior directly on the coordinate we sample in (ℓ), and we sample φ directly rather than through a transform.

[8] Generate a random permutation of M elements and use that order. The order does not change the stationary distribution, but randomizing it is good practice.

Note that the accept/reject in step III only changes (κ, φ); the $\theta_i^{(t)}$ you drew in step I stay as they are (they are not reverted on a rejected MH move), and the *next* sweep's Gibbs step will resample them fresh under whichever (κ, φ) survived. Record $\kappa^{(t)}$ and $\varphi^{(t)}$ each sweep (after burn-in).

(d) **Run the sampler on two sets of marble bags.** Use:

I. 10 bags where each bag has 9 white and 11 black marbles ($y_i = 9$, $n_i = 20$).

II. 5 bags with 1 white and 19 black marbles, and 5 bags with 19 white and 1 black marble.

For each set, run the sampler for $T = 3000$ recorded sweeps (after discarding a burn-in of a few hundred). Include (two separate) histograms of the κ samples and the φ samples for both I. and II. Also report the posterior mean $\bar{\kappa}$ and $\bar{\varphi}$, and the predictive probability that a marble is white in a new bag, θ_{M+1}, which (because $E[\theta \mid \kappa, \varphi] = \varphi$) is estimated by $\bar{\varphi}$.[9]

(e) **Tune the step sizes.** Report the *acceptance rate* (fraction of MH proposals accepted) for each bag set, and tune the step sizes s_φ and s_ℓ so that the acceptance rate lands in roughly the 0.2–0.5 range for *each* set. You will likely find that the two bag sets need *different* step sizes. In 2–3 sentences, explain (i) what happens to the acceptance rate when a step size is too small versus too large, and (ii) why the two bag sets favor different step sizes (hint: think about how concentrated the posterior over κ is in each case — one bag set pins κ down much more tightly than the other — and connect it to the mixing discussion from lecture).

(f) **Write up your results.** In one paragraph, describe the histograms for κ and φ given each set of marble bags and explain what their values mean. In a second paragraph, compare the two bag sets: what is similar and what is different in $\bar{\kappa}$, $\bar{\varphi}$, and the predictive value, and why?

3. (Effective sample size). One way to summarize how useful a set of Monte Carlo samples is at estimating some quantity is the *effective sample size* (N_{eff}). It answers: "my T weighted samples are worth how many equally-weighted independent samples?" A commonly used definition is

$$N_{\text{eff}} = \frac{1}{\sum_{t=1}^{T} \left(w^{(t)}\right)^2}, \qquad \text{where } w^{(t)} \geq 0 \text{ and } \sum_t w^{(t)} = 1. \tag{1}$$

For an importance sampler, the (normalized) weight of sample t is $w^{(t)} \propto p(x^{(t)})/q(x^{(t)})$:

$$w_{IS}^{(t)} = \frac{p(x_{IS}^{(t)})/q(x_{IS}^{(t)})}{\sum_{t'=1}^{T} p(x_{IS}^{(t')})/q(x_{IS}^{(t')})}.$$

(Note: these weights are defined from the proposal/target *mismatch* alone — the indicator $I(x > 2)$ from Problem 1 is *not* part of the weight here; we are measuring the quality of the samples, not the particular quantity being estimated.) For a set of plain Monte Carlo samples drawn directly from p, every sample has weight $w_{MC}^{(t)} = 1/T$.

(a) **Good proposal vs. bad proposal.** Here N_{eff} measures how well a proposal q *covers the whole target* $p = N(0, 1)$ — a q that matches p everywhere gives even weights and

[9] More precisely, the posterior predictive mean is $E[\varphi \mid \text{data}]$, which you estimate by the average of your φ samples, $\bar{\varphi}$.

$N_{\text{eff}} \approx T$; a q that misses much of p's mass gives a few dominant weights and a small N_{eff}. Using the importance-sampling weights in Eq. (1), compute N_{eff} for two proposals, both targeting $p = N(0, 1)$:

(i) a *good* proposal that overlaps p well — for example $q = N(0, 1.5^2)$ (same center as p, a bit wider); and

(ii) a *bad* proposal that overlaps p poorly — for example $q = N(4, 1)$, shifted well off p's mass. (A much narrower q, such as $N(0, 0.5^2)$, also lowers N_{eff}, though less drastically than a far-shifted one.)

Use $T = 1000$ for both. Report the two effective sample sizes (with $q = N(0, 1.5^2)$ versus $q = N(4, 1)$ you should see the good q give an N_{eff} in the high hundreds and the far-shifted bad q give one in the single digits). In 2–3 sentences, explain why the bad proposal gives a much smaller N_{eff} — what does a small N_{eff} tell you about the weights, and how does this connect to the weight-variance figure from lecture (good $q \Rightarrow$ even weights $\Rightarrow$ high N_{eff}; bad $q \Rightarrow$ a few weights dominate $\Rightarrow$ low N_{eff})? *(Note: do not use the $q = N(2, 1)$ from Problem 1 here — that proposal was tuned for the* tail $X > 2$*, not for covering all of p, and it is exactly the subject of part (b).)*

(b) **A puzzle: importance sampling vs. plain Monte Carlo.** Now compute N_{eff} for *two ways of estimating* $P(Y > 2)$ from Problem 1: (i) the importance sampler with $q = N(2, 1)$, using $w_{IS}^{(t)} \propto p/q$; and (ii) plain Monte Carlo drawing from $p = N(0, 1)$ directly, using $w_{MC}^{(t)} = 1/T$. You will find something that looks backwards: plain Monte Carlo reports $N_{\text{eff}} = T = 1000$ (its weights are all equal), while the importance sampler reports a *much smaller* N_{eff} (around 50). Yet from Problem 1 you know the importance sampler's estimate of $P(Y > 2)$ is the *more accurate* one (lower variance for the same T).

Resolve the puzzle in a short paragraph. Address: (i) what quantity is N_{eff} in Eq. (1) actually measuring — the *evenness of the weights*, or the *accuracy of the estimate*? (ii) Why does plain Monte Carlo trivially get $N_{\text{eff}} = T$ (is that a real efficiency advantage, or an artifact of all weights being equal)? (iii) Why can the importance sampler be the more accurate estimator of $P(Y > 2)$ despite a smaller weight-based N_{eff} (hint: almost all of the plain Monte Carlo samples land where $I(x > 2) = 0$ and contribute nothing to the tail estimate, whereas the importance sampler places samples where they matter)? The lesson: "effective sample size" is an overloaded term — the weight-evenness N_{eff} in Eq. (1) is a useful diagnostic of *proposal quality* (part a), but it is *not* a direct measure of estimator accuracy, and comparing it across two estimators that weight their samples differently can be misleading.

References

Kemp, C., Perfors, A., and Tenenbaum, J. B. (2007). Learning overhypotheses with hierarchical Bayesian models. *Developmental Science*, 10(3):307–321.

Variable/Equation	Meaning
M	Number of bags of marbles.
y_i	Number of white marbles observed in bag i.
n_i	Total number of marbles observed from bag i.
θ_i	Proportion of white marbles in bag i.
$\mathbf{y}, \mathbf{n}$	Vectors of the observed white counts $\mathbf{y} = (y_1, \ldots, y_M)$ and totals $\mathbf{n} = (n_1, \ldots, n_M)$.
$y_i \mid n_i, \theta_i \sim \text{Binomial}(\theta_i; n_i)$	The number of white marbles drawn from bag i is Binomial with parameters θ_i and n_i.
φ	*Mean* of the Beta distribution over bag proportions, $\varphi = a/(a+b) \in (0,1)$. The prior proportion of white marbles.
κ	*Concentration* of the Beta distribution, $\kappa = a + b > 0$. Large κ: bags look alike (mass near φ); small κ: bags pushed toward 0 or 1.
$\theta_i \mid \kappa, \varphi \sim \text{Beta}(\kappa\varphi, \kappa(1-\varphi))$	θ_i is Beta distributed with shape parameters $a = \kappa\varphi$ and $b = \kappa(1-\varphi)$.
$\text{Beta}(\kappa\varphi + y_i,\ \kappa(1-\varphi) + (n_i - y_i))$	The conjugate posterior over θ_i after observing (y_i, n_i).
$\theta_i^{(t)}$	The t-th sample of the proportion of white marbles in bag i.
$\ell = \log \kappa$	The log concentration; the MH random walk on the concentration is done on this log scale so it never proposes $\kappa \leq 0$.
$s_\varphi,\ s_\ell$	The MH proposal step sizes (standard deviations of the Gaussian random walks on φ and ℓ); you tune these.
N_{eff}	Effective sample size, $1/\sum_t (w^{(t)})^2$ for normalized weights $w^{(t)}$.

Table 1: The symbols and variables in Problems 2 and 3 and their meanings. (This assignment is language-agnostic: the GenJAX, Python, and R stencils provide the distribution functions you need — e.g. a Beta pdf, a Beta sampler, and a Normal log-pdf.)