Chapter 4

Methods based on coupled Markov chains

From the previous chapters, we know that we can sometimes couple chains (X_t) and (Y_t) such that they marginally evolve according to a Markov transition K, but also such that some distance $d(X_t, Y_t)$ decreases as $t \to \infty$; either in expectation, or possibly $d(X_t, Y_t) = 0$ for all t larger than some meeting time τ. In these notes we discuss methods that implement such couplings of Markov chains. Outside of Markov chains, there are numerous settings where two things prove superior than one, a striking example being wings on a plane, and, to a lesser degree, wheels on a bicycle. The methods of this chapter provide further examples of this universal phenomenon.

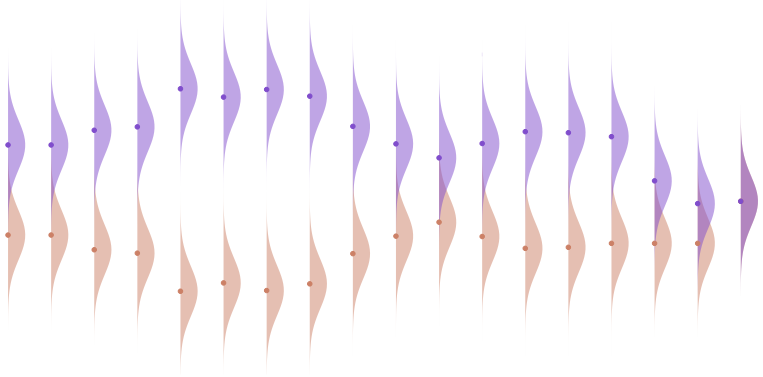

Figure 4.1: Proposal distributions of two chains evolving according to a random walk Metropolis–Hastings.

4.1 Simulation of coupled chains

We start by aggregating and summarizing some implementable couplings of MCMC algorithms. These will be directly useful in the implementation of the methods described in the following sections.

4.1.1 Coupling Metropolis–Hastings

We start with the canonical MCMC algorithm: Metropolis–Hastings. For such algorithm, with proposal $q(x \to \cdot)$, a generic coupling construction is described in Algorithm 1. Two proposals $(X^\star, Y^\star)$ are drawn from a coupling of $q(X \to \cdot)$ and $q(Y \to \cdot)$. If such coupling is maximal, then $X^\star = Y^\star$ occurs with maximal probability under the constraints $X^\star \sim q(X \to \cdot)$ and $Y^\star \sim q(Y \to \cdot)$. If such coupling results from using "common random numbers" in some implementation of $q(X \to \cdot)$ and $q(Y \to \cdot)$, then perhaps $X^\star$ and $Y^\star$ are closer to one another than X and Y were; an example of this was provided with Hamiltonian Monte Carlo. Next, the acceptance mechanism is implemented for $X^\star$ and $Y^\star$ using two uniforms, U and V. One can for instance use $U = V$, or independent draws, or another coupling. Overall we can check that the procedure satisfies the marginal constraints: $X' \sim K(X \to \cdot)$ and $Y' \sim K(Y \to \cdot)$ where K denotes the MH kernel. Jointly we might write $(X', Y') \sim K((X, Y) \to \cdot)$.

Note that there is a difference between coupling the proposals, $q(X \to \cdot)$ and $q(Y \to \cdot)$ and coupling the transition kernels $K(X \to \cdot)$ and $K(Y \to \cdot)$. Coupling the proposals is easier, since the distributions $q(X \to \cdot)$ and $q(Y \to \cdot)$ are typically of a simple form, for example Normal or Student, whereas the kernels $K(X \to \cdot)$ are less explicit. Thus, we can sample typically more easily from a maximal or contractive coupling of $q(X \to \cdot)$ and $q(Y \to \cdot)$. Figure 4.1.

Regarding the coupling of the uniforms used in Algorithm 1 to accept or not

Algorithm 1 A coupled Metropolis–Hastings kernel

1. Sample $(X^\star, Y^\star)$ from a coupling of $q(X \to \cdot)$ and $q(Y \to \cdot)$.
2. Sample (U, V) from a coupling of Uniform$(0, 1)$ with itself.
3. If $U < \alpha_{\text{MH}}(X, X^\star)$, set $X = X^\star$.
4. If $V < \alpha_{\text{MH}}(Y, Y^\star)$, set $Y = Y^\star$.
5. Return (X', Y').

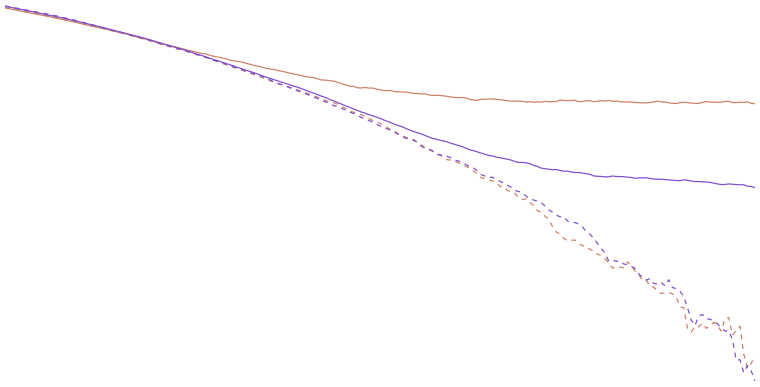

Figure 4.2: Distance between chains, using four different types of couplings of the proposal distributions, in a random walk Metropolis–Hastings.

the proposals, the simplest strategy is undoubtedly to draw $U \sim \text{Uniform}(0, 1)$ and to set $V = U$. An alternative strategy is to view $(X, X^\star)$ as the possible values for X' and likewise $(Y, Y^\star)$ are the possible values for Y'. If one wants to minimize the expected distance between X' and Y', this suggests drawing (X', Y') from an optimal transport coupling of the two discrete measures $(1 - \alpha(X, X^\star))\delta_X + \alpha(X, X^\star)\delta_{X^\star}$ and $(1 - \alpha(Y, Y^\star))\,\delta_Y + \alpha(Y, Y^\star)\delta_{Y^\star}$. If $X = Y$ and $X^\star = Y^\star$ this is equivalent to the simplest strategy of using common uniforms $U = V$, but otherwise it is a different coupling. See 4.2 a graph of the distances between two chains, when employing four different types of couplings between proposal distributions, and identical uniforms for acceptance.

See Johnson [1996, 1998], Neal [1999] for early methodological uses of such couplings of MH kernels; this is not exactly new!

4.1.2 Coupling Gibbs samplers

For Gibbs samplers, one can design an implementable coupling by considering each conditional update in sequence, and designing a coupling as in the previous

Figure 4.3: Component become identical in a run of a coupled Gibbs sampler.

section, using common random numbers, maximal couplings, etc. With a random scan Gibbs sampler, there is also a choice of components to update. In general, if the marginal kernel samples a component I uniformly in $\{1, \ldots, d\}$, and then updates $X_t[I] \sim \pi(dx[I]|X_{t-1}[\backslash I])$, the coupled kernel might first sample (I, I') from a coupling of the Uniform distribution on $\{1, \ldots, d\}$ and itself. Then we sample $(X_t[I], Y_t[I'])$ from a coupling of $\pi(dx[I]|X_{t-1}[\backslash I])$, and $\pi(dx[I']|Y_{t-1}[\backslash I'])$. The simplest strategy here is undoubtedly to sample I and to set $I' = I$ so that the same component is updated in both chains. However it seems like better strategies would update components according to how they differ to one another across the chains.

In Gibbs samplers, it is often the case that some of the components are conditionally independent given some others. It might be the case that coupling a small number of components in a smart way is enough for all of the other components to contract or meet. See Section 4.4 in Neal [1999] and Section 4.3 in Jacob et al. [2020] about couplings of Gibbs samplers. Figure 4.3.

4.1.3 Combining kernels to trigger exact meetings

Coupled Markov kernel for random walk MH with Normal proposals, Algorithm 1, can be interweaved with whatever other kernel that would induce contractions. Following Heng and Jacob [2019], consider two chains evolving in $\mathbb{R}^d$. Suppose that you have a way of making the two chains close to one another regularly enough; perhaps by using couplings of Hamiltonian Monte Carlo kernels, or couplings of Gibbs sampler, or others. A coupling of random walk MH, with maximal couplings of the proposals, can then be employed to obtain exact meeting. More precisely, if the proposal $q(x \to \cdot)$ is Normal with mean x and variance $\sigma^2 I_d$, then with a maximal coupling, $\mathbb{P}(X^\star = Y^\star)$ is $1 - (2\pi)^{-1/2}\delta/\sigma + \mathcal{O}(\delta^2/\sigma^2)$ as $\delta/\sigma \to 0$, where $\delta = |x - y|$ and $|\cdot|$ is the Euclidean distance. This means that, if we have some idea of the distance $\delta = |x - y|$ between the chains during the run of the coupled chains, we can choose σ accordingly to achieve a desired $\mathbb{P}(X^\star = Y^\star)$. Furthermore, if σ is small the acceptance probability of the random

walk MH algorithm is large, and thus $\{X' = Y'\}$ might occur with a significant probability.

If we denote the original kernel by K and its contractive coupling by $\bar{K}$, and if we denote the random walk MH kernel by K_σ and a coupling of it with maximal coupling of proposals by $\bar{K}_\sigma$, then we can introduce a new kernel defined as a mixture and denoted by M, and a coupled kernel $\bar{M}$,

$$M(x \to \cdot) = (1-\omega)K(x \to \cdot) + \omega K_\sigma(x \to \cdot),$$
$$\bar{M}((x,y) \to \cdot) = (1-\omega)\bar{K}((x,y) \to \cdot) + \omega \bar{K}_\sigma((x,y) \to \cdot),$$

with ω a mixture weight in $(0,1)$. If ω is small, the new kernel M is very similar to the original one, and thus its mixing properties will be similar. The possibility of sampling from K_σ from time to time offers opportunities for exact meetings, provided that the states (x,y) are close-by.

Alternatively, and somewhat similarly, exact meetings could be triggered via the regeneration approach of Brockwell and Kadane [2005]. Let us start again from a Markov kernel K, deemed satisfactory overall but not necessarily easy to couple in a way that the resulting chains would meet, rapidly, or at all. The idea of Brockwell and Kadane [2005] is to introduce an artificial state, denoted by "α" and such that the original state space $\mathbb{X}$ is now extended to $\mathbb{X} \cup \alpha$. A target distribution $\bar{\pi}$ is defined on $\mathbb{X} \cup \alpha$ as $\bar{\pi}(x) \propto \pi(x) + p_\alpha \mathbb{1}(x \in \alpha)$ where p_α is an arbitrary constant to be chosen. The method requires the introduction of a proposal distribution from the state α to $\mathbb{X}$, denoted by $q_\alpha(\cdot)$. This is meant to be a distribution with support wherever some mass of the target π is; it does not have to be a good approximation of π itself.

With these ingredients, a chain (X_t) evolves according to the following rules,

- if $X_{t-1} \in \mathbb{X}$, with probability $1-\omega$, sample $X_t \sim K(X_{t-1} \to \cdot)$; otherwise, propose a move to α, accepted with probability $\min(1, p_\alpha q_\alpha(x)/\pi(x))$, otherwise $X_t = X_{t-1}$,
- if $X_{t-1} \in \alpha$, sample a draw $X^\star \sim q_\alpha(\cdot)$, and accept it with probability $\min(1, \pi(x)/(p_\alpha q_\alpha(x)))$; otherwise $X_t = X_{t-1} = \alpha$.

The construction creates a chain that lives in $\mathbb{X} \cup \alpha$ with the property that it "regenerates" whenever it enters the state α: the segments of the chain between two states α are independent of one another. In the context of coupling, the construction can be used to create exact meetings since the chains meet as soon as they are simultaneously in state α.

4.1.4 Lagged chains

The above discussions concern the design of coupled kernels $\bar{K}$ such that $(X', Y') \sim \bar{K}((X,Y) \to \cdot)$ implies that $X' \sim K(X \to \cdot)$ and $Y' \sim K(Y \to \cdot)$, and further (X', Y') tends to get closer to one another, or to meet exactly. In the previous chapter such couplings were introduced to propagate two chains, (X_t) and (Y_t), by generating $(X_t, Y_t) \sim \bar{K}((X_{t-1}, Y_{t-1}) \to \cdot)$, possibly started

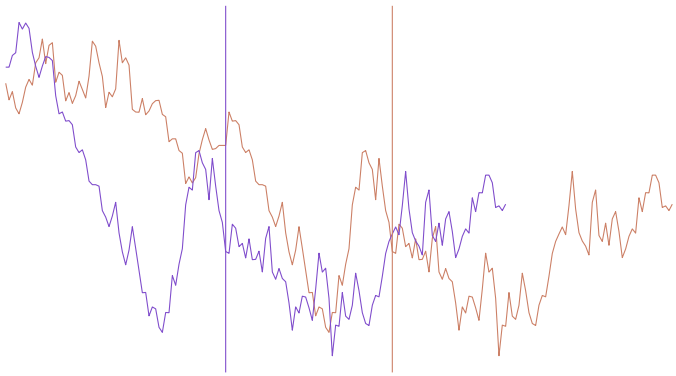

Figure 4.4: Coupled chains with a lag.

from different distributions. In this chapter we will encounter pairs pairs of chains started from the same distribution π_0, and propagated with $\bar{K}$ but with a time lag. The construction is described in Algorithm 2.

Algorithm 2 Coupled chain with lag L, coupled kernel $\bar{K}$ and single kernel K.

1. Sample $X_0 \sim \pi_0$ and $Y_0 \sim \pi_0$, independently or not.
2. Sample $X_1, \ldots, X_L$ recursively using the Markov kernel K.
3. For $t \geq L$, sample $(X_{t+1}, Y_{t-L+1}) \sim \bar{K}((X_t, Y_{t-L}) \to \cdot)$.
   If $X_{t+1} = Y_{t-L+1}$ set $\tau^{(L)} = t + 1$.

See Figure 4.4.

4.1.5 Simulating many chains

Let us first discuss how many chains can be coupled together. Let us denote by $C \in \mathbb{N}$ the number of chains, and denote the chains by $X_t^{(c)}$ for the c-th chain at time t. As always the simplest coupling is the independent coupling, where the C chains evolve independently of one another. The use of multiple independent chains is routine in MCMC, since

- running one chain per processing unit provides an obvious route to parallel computing,
- comparing multiple independent chains to one another, and checking that they all end up "in agreement" is a simple way of diagnosing possible issues with an MCMC procedure.

There seems to have been some debate among statisticians about "many short chains" versus "one long run", see e.g. Charles Geyer's page about it. I am not sure what came out of this debate. Geyer [1992].

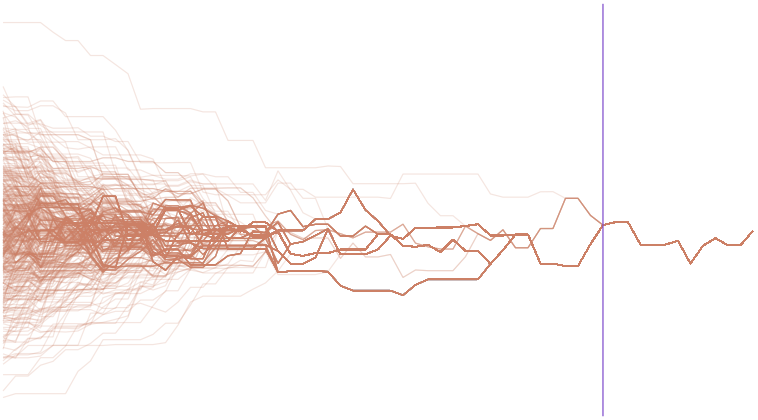

Figure 4.5: Many chains, eventually coinciding.

In the context of these notes we are of course more excited about non-independent couplings, and what we can do with them. If we assume that some random function representation of the Markov kernel K is available, whereby $X_t = \psi(X_{t-1}, U_t)$ for some function ψ and Uniform U_t in $(0, 1)$, then a possible coupling of C chains is obtained by using common random numbers, so that $X_t^{(c)} = \psi(X_{t-1}^{(c)}, U_t)$ for all chains c and the same variable U_t. If the variables (U_t) can be stored in memory or re-computed from scratch, the simulation of the C chains can be done entirely in parallel. Johnson [1996] discusses the use of common random number strategies in Gibbs samplers.

If we are looking for exact meetings and the state space is continuous, common random number strategies might not cut it. We can turn to generalizations of maximal couplings to more than two distributions. Generally, couplings of more than two distributions are called "multi-marginal" couplings. Consider for example random walk MH, where at step $t-1$ the states $(X_{t-1}^{(c)})$ for $c \in \{1, \ldots, C\}$ are available. The next proposals $(X^{\star(c)})$ should be drawn from couplings of the C proposal distributions $(q(X_{t-1}^{(c)} \to \cdot))$. We might the coupling to be such that $\{X^{\star(c)} = X^{\star(c')}\}$ occurs for some pairs (c, c'), or for as many pairs as possible. A possible coupling is obtained by stitching together maximal couplings of $(q(X_{t-1}^{(c)} \to \cdot), (q(X_{t-1}^{(c')} \to \cdot)))$ for an arbitrary sequence of pairs (c, c') covering $\{1, \ldots, C\}$, e.g. $(1, 2)$, $(2, 3)$, $\ldots$, $(C-1, C)$. This does not seem optimal in any way, but it enables meetings $\{X^{\star(c)} = X^{\star(c')}\}$ for some pairs (c, c'), and thus over multiple iterations all the chains can possibly coincide. There are relevant discussions in Reutter and Johnson [1995]. Figure 4.5.

4.2 Unbiased estimation of stationary expectations

Here we describe how coupled chains can be used to obtain unbiased estimators of $\pi(h) = \int h(x)\pi(dx)$ where π is the stationary distribution of a Markov chain, and h a test function. Recall that MCMC estimators of the form $(T - b + 1)^{-1} \sum_{t=b}^{T} h(X_t)$ are biased, for any choice of burn-in b and total number of steps T, because the Markov chain (X_t) in general starts outside of stationarity and has no precise reason to be stationary from step b onwards. Thus in general

$$\mathbb{E}\left[\frac{1}{T-b+1}\sum_{t=b}^{T} h(X_t)\right] \neq \pi(h).$$

Although, if b is "very large" then the bias is "very small", the main question about the number of MCMC iterations remains "how many?".

One could question the relevance of the bias; certainly other aspects are important too, such as cost and variance. The benefits of unbiased estimators in the Monte Carlo setting are described in Rosenthal [2000], or Section 3.3 in Jacob et al., 2020 which summarises some of the literature on budget-constrained simulations [Glynn and Heidelberger, 1990, 1991]. An obvious advantage of unbiased estimators is that they can be simulated in parallel independently and averaged over, leading to consistent estimators for which the variance can be quantified easily. In other words, there are advantages of unbiased estimators in the setting of Monte Carlo, both in terms of parallel computing and on the quantification of errors. As we will see in this section, with certain couplings the "burn-in" bias can be completely removed.

4.2.1 Telescoping sums

Let us start with Forsythe and Leibler [1950]. A bit of trivia: Forsythe was the husband of Alexandra Illmer Forsythe who wrote the first-ever published computer science textbook. Leibler is the Leibler in Kullback–Leibler. These authors credit the following idea to John von Neumann and Stanisław Ulam, themselves generally considered the pioneers of Monte Carlo methods. The problem is to estimate some coefficient of the inverse of a matrix B without fully inverting it; see Wasow [1952], Kuti [1982] for related applications. Assuming that all the eigenvalues of $A = I - B$ are less than one in absolute value, we can express $(B^{-1})_{ij}$ as $\sum_{k=0}^{\infty}(A^k)_{ij}$, a matrix version of the series expansion $(1-x)^{-1} = \sum_{k=0}^{\infty} x^k$. The infinite series does not seem too practical, and a naive approach would be to truncate it, and return $\sum_{k=0}^{K}(A^k)_{ij}$ for some large, arbitrary K. This certainly introduces a bias, which might not be easy to quantify... exactly the same issue as in the choice of burn-in for MCMC estimators.

The key idea here is to relate the infinite series of interest to the expectation of some random variable that we can simulate. If we can come up with such a

variable, if we can sample it fast enough, and if it has a finite variance, then the central limit theorem provides us with an estimator that converges at the Monte Carlo rate. We will not describe the specific construction in Forsythe and Leibler [1950] here (see also Wasow [1952] for an explanation), and instead will focus on a simpler setting.

Consider an arbitrary series $\sum_{k\geq 0} a_k$ which we want to estimate. We can always write

$$\sum_{k\geq 0} a_k = \sum_{k\geq 0} \frac{a_k \mathbb{P}(K=k)}{\mathbb{P}(K=k)} = \sum_{k\geq 0} \mathbb{E}\left[\frac{a_k \mathbb{1}(K=k)}{\mathbb{P}(K=k)}\right], \tag{4.1}$$

where K is an integer-valued random variable such that $\mathbb{P}(K=k) > 0$ for all $k \geq 0$. Denote $p_k = \mathbb{P}(K=k)$ for the probabilities associated with K. We cannot always swap the expectation and the limit, but if we would, the random variable

$$G = \frac{a_K}{p_K} \tag{4.2}$$

obtained by sampling K and reporting a_K/p_K would have expectation equal to $\sum_{k\geq 0} a_k$. Another, similar estimator is defined as $\sum_{k=0}^{K} a_k/\mathbb{P}(K \geq k)$, for example; variants are numerous.

If we can sample K and evaluate $p_k = \mathbb{P}(K=k)$ for all $k \geq 0$, then we can implement G, and its cost is of the order of K operations assuming that we can obtain the j-th term of the sequence (a_k) in $\mathcal{O}(j)$ operations. We will need to pay attention to the variance of G, but before that note that the same "debiasing trick" applies to limits. Assume that the quantity of interest can be written $\lim_{k\to\infty} x_k$. We can always write this as a telescoping sum:

$$\lim_{k\to\infty} x_k = x_0 + \sum_{k=1}^{\infty} (x_k - x_{k-1}), \tag{4.3}$$

and, in passing, we could have started the telescoping series at any index j instead of zero, and we could have use any lag other than one. Both of these points will be important in some of the estimators considered later on. In any case, we can see that with appropriate definitions of (a_k) we can write $\lim_{k\to\infty} x_k$ as $\sum_{k\geq 0} a_k$, and apply a debiasing trick to obtain an unbiased estimator.

Next we consider the variance of G, and the expectation of its cost. The finiteness of the variance and of the expected cost are of paramount importance if we want to use central limit theorems for estimators that have random computing times [Glynn and Whitt, 1992]. This is crucial if we want our estimators to have the canonical "$\sqrt{N}$" rate of convergence, and to construct asymptotically valid confidence intervals. Thus we take finiteness of the variance and of the expected cost to be very important, minimal properties, that any practical Monte Carlo estimator should satisfy.

So, let's proceed to the cost and variance of G. This estimator G is termed "single-term estimator" in Rhee [2013]. For the cost, we require $\mathbb{E}[K]$ to be finite,

thus

$$\sum_{k\geq 0} k p_k < \infty. \tag{4.4}$$

This requires that (p_k) goes to zero faster than k^{-2}. On the other hand, for the second moment $\mathbb{E}[G^2]$ we require

$$\sum_{k\geq 0} \frac{a_k^2}{p_k} < \infty. \tag{4.5}$$

which means that (a_k^2) must decay faster than k^{-3} in combination with $p_k = o(k^{-2})$. Similar restrictions apply to related estimators, such as the "coupled sum estimator" $\sum_{k=0}^{K} a_k / \mathbb{P}(K \geq k)$. Essentially, only if the original sequence $(a_k)_{k\geq 0}$ decreases fast enough is it possible to obtain an unbiased estimator of $\sum_{k\geq 0} a_k$ with both finite variance and finite computing cost.

These techniques have received renewed attention lately, see McLeish [2011], Rhee and Glynn [2012, 2015], Vihola [2018], Kahale [2019]. These articles treat a setting where the object of interest is not only the limit of a deterministic sequence, such as (b_k) above, but more specifically the limit of the expectation of a sequence of random variables; see Glynn [1983], Wagner [1987], McLeish [2011], Rhee and Glynn [2012], and outside of Monte Carlo, e.g. Rychlik [1990] or Gajek [1995]. The impactful work of Rhee and Glynn [2015] presents an unbiased multilevel Monte Carlo estimator for SDEs, see also Giles [2015], Vihola [2018]. Next we will focus on the application of the debiasing idea to MCMC, pioneered by Glynn and Rhee [2014].

4.2.2 Telescoping sums of Markov chain expectations

The starting point is that, in the light of the possibility of debiasing sequences that converge fast enough, it might be possible to debias sequences generated by Markov chain Monte Carlo estimators. A first approach would be to define (b_t) as the MCMC ergodic average computed on t iterations, $t^{-1}\sum_{s=1}^{t} h(X_s)$, for some Markov chain $(X_t)_{t\geq 0}$ and test function h. However, the resulting sequence (b_t) does not converge fast enough for practical debiasing with finite expected cost and finite variance; see McLeish [2011] for a brave attempt.

Contrarily to ergodic averages, the marginal distributions (π_t) of Markov chains might converge to their limit π at a geometric rate, as measured by various distances. Geometric rates indeed appear fast enough for practical debiasing. Thus Glynn and Rhee [2014] proceed as follows. Assume that the test function h is integrable with respect to any distribution encountered below. The expectation of interest is $\pi(h) = \lim_t \pi_t(h)$, thus

$$\pi(h) = \mathbb{E}[h(X_0)] + \mathbb{E}[h(X_1)] - \mathbb{E}[h(X_0)] + \mathbb{E}[h(X_2)] - \mathbb{E}[h(X_1)] + \dots \tag{4.6}$$

Each difference $\mathbb{E}[h(X_t) - h(X_{t-1})]$ might not be particularly small if $(X_t)_{t\geq 0}$ is generated by an MCMC algorithm. However this difference is also the expectation of $\Delta_t = h(X_t) - h(Y_{t-1})$, where Y_{t-1} is a random variable with marginal

distribution equal to π_{t-1}. With appropriate coupling strategies, it might be possible to construct a joint process $(X_t, Y_t)_{t\geq 0}$ such that $Y_t \sim \pi_t$ for all $t \geq 0$, but also such that $\mathbb{E}[|\Delta_t|]$ goes to zero as $t \to \infty$. Typically, the construction would look like that of Algorithm 2. The coupled kernel $\bar{K}$ is such that, marginally, $X_{t+1} \sim K(X_t \to \cdot)$ and $Y_t \sim K(Y_{t-1} \to \cdot)$. The kernel $\bar{K}$ might for example correspond to the procedure described in Algorithm 1.

Glynn and Rhee [2014] consider various settings, including that of iterated random functions and couplings obtained with common random numbers. There, $\bar{K}$ corresponds to drawing U_t at step t and setting $X_{t+1} = \psi(X_t, U_t)$ and $Y_t = \psi(Y_{t-1}, U_t)$.Under the assumption

$$\mathbb{E}[\|\psi(x, U) - \psi(y, U)\|^2] \leq b\|x - y\|^2$$

for some $b < 1$, and assume that the test function h is κ-Lipschitz for some $\kappa < \infty$, then

$$\mathbb{E}[|\Delta_t|^2] \leq \kappa^2 b^{t-1} \mathbb{E}[\|X_1 - Y_0\|].$$

Thus, under regularity assumptions on π_0 the term $\mathbb{E}[|\Delta_t|^2]$ would go to zero geometrically. Then, with the convention $\Delta_0 = h(X_0)$, Glynn and Rhee [2014] show that the estimator

$$G = \sum_{t=0}^{K} \frac{\Delta_t}{\mathbb{P}(K \geq t)}, \tag{4.7}$$

is unbiased for $\pi(h)$, and has a finite variance provided that the "truncation variable" K satisfies

$$\sum_{k\geq 0} \frac{b^k}{\mathbb{P}(K \geq k)} < \infty.$$

This is satisfied for any K with polynomial tails, and for Geometric variables with parameter smaller than $1 - b$. This illustrates again that if Δ_t goes faster to zero, then the contraction rate b is smaller, and then there is more freedom in the choice of K. With this construction the cost of constructing the estimator is the cost of running two chains for K steps with common random numbers, essentially of the order of K operations.

This estimator G, termed the "Glynn–Rhee" estimator, has been directly applied by Agapiou et al. [2018] to the MCMC setting, while Glynn and Rhee [2014] considered also other sets of conditions, other couplings and settings outside of MCMC. On the one hand, this estimator is really extraordinary, because it manages to be unbiased for $\lim_{t\to\infty} \mathbb{E}[h(X_t)]$. It completely avoids the issue of burn-in bias in that way. Its variance and expected cost are finite under reasonable conditions, for example the coupling employed is contractive and if K is appropriately chosen. Nevertheless, as made quite apparent in the numerical experiments of Agapiou et al. [2018], for example in the fact that these experiments concern toy examples, the estimator in (4.7) is not necessarily practical.

- The choice of truncation variable K is not easy, as the optimal choice in terms of mean squared error would depend on the rate of decay of $\mathbb{E}[|\Delta_t|^2]$ to zero, which is rarely known analytically.

- Furthermore, there is no reason for G to be comparable to the MCMC ergodic average that could be obtained with the same amount of computing budget. Note that fair comparisons between Glynn–Rhee estimators and ergodic averages are not that straightforward. In part because ergodic averages require a choice of burn-in, which could be made judiciously or not; but is often done in an ad-hoc manner anyway. Also in part because these estimators can benefit differently from different computing architectures.

4.2.3 Unbiased Markov chain Monte Carlo

We next move to the unbiased MCMC method described in Jacob et al. [2020]. Conceptually, this is a mild modification of the Glynn–Rhee estimator in (4.7), but with a difference: its efficiency is typically comparable to that of standard MCMC ergodic averages. Thus it can be considered a generic alternative to such MCMC estimators. In passing, we will also remove the truncation variable K, and instead we require coupled chains to meet exactly, with a meeting time τ. We here describe a general version, that incorporates an arbitrary lag parameter L, a "starting time" k, a "prospective end time" m. The tuning parameters k and m were introduced in Jacob et al. [2019, 2020] to improve the efficiency of (4.7). The lag was suggested by both Paul Vanetti and Niloy Biswas and described in Biswas et al. [2019].

The construction of the chain using a kernel K and a coupled kernel $\bar{K}$ is given in 2. The assumptions on $\bar{K}$ are:

- Given X, Y, $(X', Y') \sim \bar{K}((X, Y) \to \cdot)$ is such that $X' \sim K(X \to \cdot)$ and $Y' \sim K(Y \to \cdot)$, and if $X = Y$ then $X' = Y'$ almost surely.
- The meeting time $\tau^{(L)} = \inf\{t \geq L : X_t = Y_{t-L}\}$ has Geometric tails.

The latter assumption is relaxed in Middleton et al; instead $\tau^{(L)}$ can be assumed to have polynomial tails, e.g. $\mathbb{P}(\tau > t) \leq Ct^{-\kappa}$ as long as the exponent κ is large enough.

The construction corresponds to a telescopic sum representation of $\pi(h)$ as

$$\pi(h) = \mathbb{E}[h(X_0)] + \mathbb{E}[h(X_L)] - \mathbb{E}[h(X_0)] + \mathbb{E}[h(X_{2L})] - \mathbb{E}[h(X_L)] + \ldots \quad (4.8)$$

We can define the estimator $H_0^{(L)}$ as

$$H_0^{(L)} = h(X_0) + \sum_{j=1}^{\infty} \{h(X_{jL}) - h(Y_{(j-1)L})\}.$$

It might be useful to think of $H_0^{(L)}$ as a function of the entire trajectory of $(X_t, Y_t)_{t \geq 0}$. The sum contains only finitely many terms since there is a finite meeting time $\tau^{(L)}$ such that $X_t = Y_{t-L}$ for all $t \geq \tau^{(L)}$. To further improve the efficiency, we start the telescoping sum at some index k, instead of 0, which leads to

$$H_k^{(L)} = h(X_k) + \sum_{j=1}^{\infty} \{h(X_{k+jL}) - h(Y_{k+(j-1)L})\}. \quad (4.9)$$

In this construction the first $k-1$ iterations are discarded. Thus we can view k as analogous to a choice of "burn-in". However, the bias of $H_k^{(L)}$ is zero for all k. Instead k impacts the cost and variance of $H_k^{(L)}$.

The cost of $H_k^{(L)}$ is that of running a pair of chains until they meet, at time $\tau^{(L)}$, and further until step k if $k > \tau^{(L)}$. Thus it's something like $\max(k, \tau^{(L)})$. The variance is harder to characterize analytically than the cost; we refer to Glynn and Rhee [2014], Vihola [2018] or to Section 3 in Jacob et al. [2020]. In practice the variance is estimated from independent runs; indeed the ease of estimation of the variance is an appeal of these estimators. Let's simply comment here on the fact that if k was a large quantile of $\tau^{(L)}$, then with large probability we would observe $H_k^{(L)} = h(X_k)$, and thus the variance of $H_k^{(L)}$ might be comparable to the variance of $h(X_k)$. If k is large, the latter would be approximately equal to the variance of $h(X)$ under the target π. In other words, whatever the distribution of the meeting time $\tau^{(L)}$ might be, if we choose k adequately the variance of $H_k^{(L)}$ might be comparable to the variance of a perfect sample from π.

Yet at this point, the estimator $H_k^{(L)}$ should seem quite wasteful. Indeed it requires running chains for $\max(k, \tau^{(L)})$ iterations but it only uses the last iterations to approximate $\pi(h)$. If k is very large this sounds like a lot of wasted efforts. Following this intuition we can indeed obtain drastic gains by averaging $H_k^{(L)}$ over consecutive choices of k. We next provide more details. Consider a range of integers $k, \ldots, m$ for $k \leq m$ and associated estimators $H_k^{(L)}, \ldots, H_m^{(L)}$ obtained from the same trajectories $(X_t, Y_t)_{t \geq 0}$. All of these estimators $(H_t^{(L)})_{t=k}^m$ are unbiased, so their average is also unbiased, and can be written

$$H_{k:m}^{(L)} = \frac{1}{m-k+1} \sum_{t=k}^{m} H_t^{(L)}.$$

With some rearrangement and using the formula (4.9), we can obtain a more interesting representation of $H_{k:m}^{(L)}$ as

$$H_{k:m}^{(L)} = \frac{1}{m-k+1} \sum_{t=k}^{m} h(X_t) + \mathrm{BC}_{k:m}^{(L)}, \tag{4.10}$$

where $\mathrm{BC}_{k:m}^{(L)}$ refers to a "bias correction", and equals

$$\mathrm{BC}_{k:m}^{(L)} = \sum_{t=k+L}^{\tau^{(L)}-1} \min\left\{1, \frac{\lceil (t-k)/L \rceil}{m-k+1}\right\} \{h(X_t) - h(Y_{t-L})\}. \tag{4.11}$$

The estimator in (4.10) is appealing because it involves the standard MCMC ergodic average $(m-k+1)^{-1} \sum_{t=k}^{m} h(X_t)$, that would computed from the original chain $(X_t)_{t \geq 0}$. The bias correction term $\mathrm{BC}_{k:m}^{(L)}$ in (4.11), computed from the coupled chains $(X_t)_{t \geq 0}$ and $(Y_t)_{t \geq 0}$ up to the meeting time $\tau^{(L)}$, an expectation that exactly cancels out the "burn-in" bias. The estimator in (4.10) can be

computed by running 2 for $\max(\tau^{(L)}, m)$ steps. If we count the cost of sampling from the kernel K as one unit, and the cost of sampling from $\bar{K}$ as one unit if the chains have met already, and as two units otherwise, then the cost of obtaining (4.10) equals $\max(L, m + L - \tau^{(L)}) + 2(\tau^{(L)} - L)$ units.

4.3 Upper bounding the burn-in bias

4.3.1 Using lagged chains

As described in details in Biswas et al. [2019], it is also possible to use the bias correction term in (4.11) to quantify the burn-in bias, rather than removing it. For a class of functions $h \in \mathcal{H}$ we have

$$\pi(h) - \pi_k(h) = \mathbb{E}\left[\sum_{j=1}^{\infty} \{h(X_{k+jL}) - h(Y_{k+(j-1)L})\}\right],$$

and thus we can write

$$\sup_{h\in\mathcal{H}} |\pi(h) - \pi_k(h)| = \sup_{h\in\mathcal{H}} \left| \mathbb{E}\left[\sum_{j=1}^{\infty} \{h(X_{k+jL}) - h(Y_{k+(j-1)L})\}\right] \right|,$$

which allows to obtain both lower and upper bounds on distances between π and π_h that can be written as "integral probability metrics", i.e. supremum of differences of expectations over some class of functions. For the total variation distance, $\mathcal{H}$ is the class of functions bounded by $1/2$, so that $h \in \mathcal{H}$ implies $|h(x) - h(y)| \leq 1$ for all x, y. Then, using the triangular inequality, we can informally write

$$\sup_{h:|h|\leq 1/2} |\pi(h) - \pi_k(h)| \leq \mathbb{E}\left[\sum_{j=1}^{\infty} \mathbb{1}(k + jL < \tau^{(L)})\right] \leq \mathbb{E}\left[\max(0, \lceil \frac{\tau^{(L)} - L - k}{L} \rceil)\right].$$

This leads to the inequality, for all $k \geq 0$, and $L \geq 1$,

$$\|\pi_k - \pi\|_{\mathrm{TV}} \leq \mathbb{E}\left[\max(0, \lceil \frac{\tau^{(L)} - L - k}{L} \rceil)\right]. \tag{4.12}$$

The inequality is reminiscent of the coupling inequality. The left-hand side involves a distance between marginal distributions, while the right-hand side depends on a choice of coupling between Markov chains. The right-hand side is almost always unavailable in closed form. However since it is written as an expectation, it can be estimated from independent replications of $\tau^{(L)}$. That is, by running Algorithm 2 in parallel as many times as possible, we can obtain realizations of $\tau^{(L)}$ and use them to upper bound the right-hand side $\|\pi_k - \pi\|_{\mathrm{TV}}$, for a range of values of k. In passing note that the right-hand side goes to one when L increases for fixed k, and to zero as k increases for fixed L. Intuitively

when L is large the variable X_L is itself at stationarity, and thus the quantity $\tau^{(L)} - L$ becomes the number of iterations before a stationary and a non-stationary chains meet.

At a more superficial level inequality (4.12) provides a way of formalizing the link between "mixing times" of Markov chains and "meeting times" associated with particular coupling of chains. The meeting times depend on the transition kernel of the chain but also on the coupling employed, and thus might not always provide a sharp upper bound on the mixing time; it necessarily depends on the performance of the coupling. An unfortunate design of coupling can lead to a vacuous inequality for practical values of k and L.

[What about lower bounds?] The integral probability metric representation is amenable to lower bounds by selecting a particular choice of function h in the class $\mathcal{H}$ under consideration.

Upper bounds can also be estimated for Wasserstein or Prokhorov distances. For this, it is convenient to employ the coupling representation of such distances, rather than the "integral" representation used above. We write

$$\mathcal{D}(\pi, \pi_k) = \inf_{X, X_k \sim \text{Couplings}(\pi, \pi_k)} \text{expr}(X, X_k),$$

for some distance $\mathcal{D}$ and some "expression"; for instance, in the case of Wasserstein distances $\text{exp}(X, X_k) = \mathbb{E}[c(X, Y_k)]$ for some cost c function. With the triangular inequality applied infinitely many times one gets

$$\mathcal{D}(\pi, \pi_k) \leq \sum_{j=1}^{\infty} \mathcal{D}(\pi_{k+jL}, \pi_{k+(j-1)L}) = \sum_{j=1}^{\infty} \text{expr}(X_{k+jL}, Y_{k+(j-1)L})$$

for an arbitrary expression. Here $(X_{k+jL}, Y_{k+(j-1)L})$ is an arbitrary coupling of $\pi_{k+jL}, \pi_{k+(j-1)L}$, within the set $\text{Couplings}(\pi_{k+jL}, \pi_{k+(j-1L)L})$, provided that these series are well defined. In fact if the series are infinite, the above inequality still makes sense, but might not be terribly useful.

[I am not sure what Li and Wang [2020] is about, or Dobson et al. [2019], but it looks relevant in this section.]

An appealing aspect of the coupling-based diagnostics described above lies in their applicability to varied problems; the meeting time $\tau^{(L)}$ is an integer-valued random variable irrespective of the space $\mathbb{X}$ on which the Markov chain is defined. Figure 4.6 illustrates this.

4.3.2 Using many chains

A use of coupled chains follows Reutter and Johnson [1995], Johnson [1996]. There C chains start from π_0 marginally, and evolve according to K in a coupled way so that there exists a meeting time $\tau(C)$ such that for all $t \geq \tau(C)$, the C chains are identical: $X_t^{(c)} = X_t^{(c')}$ for all c, c'. Here comes the big assumption.

> In a rejection sampler, a draw X_0 from π_0 would be accepted as a draw from π with probability at least $1 - r$, where $r \in [0, 1)$.

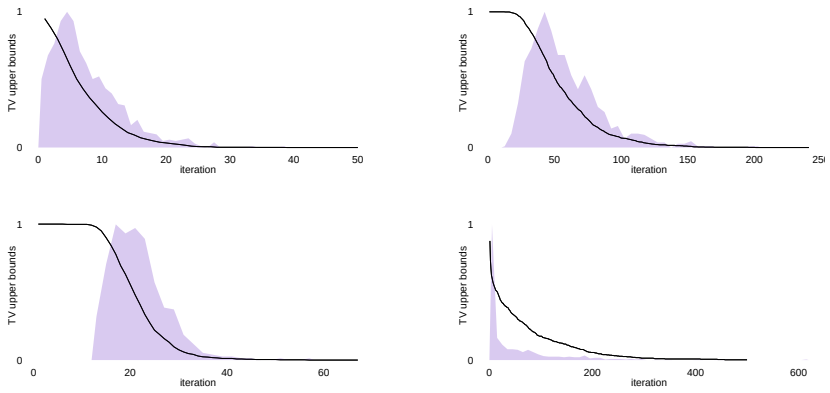

Figure 4.6: TV upper bounds, for a chain on 3-subsets out of 10; a Markov chain on graph colourings; a Hamiltonian Monte Carlo chain on logistic regression titanic data set; a Gibbs sampler for the dyestuff data set. Areas indicate the distribution of $\tau^{(L)} - L$.

The assumption might be quite realistic in some settings, where one can define π_0 to be a good approximation of π. For instance various efficient approximations of the posterior distribution have been proposed in Bayesian statistics, although usually they do not come with guarantees that are strong enough for the above assumption to be taken with confidence.

Under this assumption, the key result of Johnson [1996] can be rephrased this way:

$$\|\pi_t - \pi\|_{\mathrm{TV}} \leq \mathbb{P}(\tau(C) > t)(1 - r^C)^{-1}. \tag{4.13}$$

Proof. Let A be the event that at least one of the C chains starts from π. Under the assumption that event occurs with probability at least $1 - r^C$. Given A, we denote by s the index of a stationary chain among the C chains, and we can write

$$\begin{aligned}
\mathbb{P}(\tau(C) > t) &= \mathbb{P}(\tau(C) > t|A)\mathbb{P}(A) + \mathbb{P}(\tau(C) > t|A^c)\mathbb{P}(A^c) \\
&\geq \mathbb{P}(\tau(C) > t|A)\mathbb{P}(A) \\
&\geq \mathbb{P}(X_t^{(c)} \neq X_t^{(s)}|A)\left(1 - r^C\right) \quad \text{for any } c \neq s.
\end{aligned}$$

The last inequality comes from the fact that $\{X_t^{(c)} \neq X_t^{(s)} > t\}$ for any $c \neq s$ implies $\{\tau(C) > t\}$. We can then conclude with the coupling representation of the total variation distance. □

Note that the assumption is used to control the probability that at least one of the chains would be "as if" started at stationarity. Even though the value of r should probably be very close to one for the assumption to hold in most cases, the bound in (4.13) could be meaningful for moderate values of C thanks

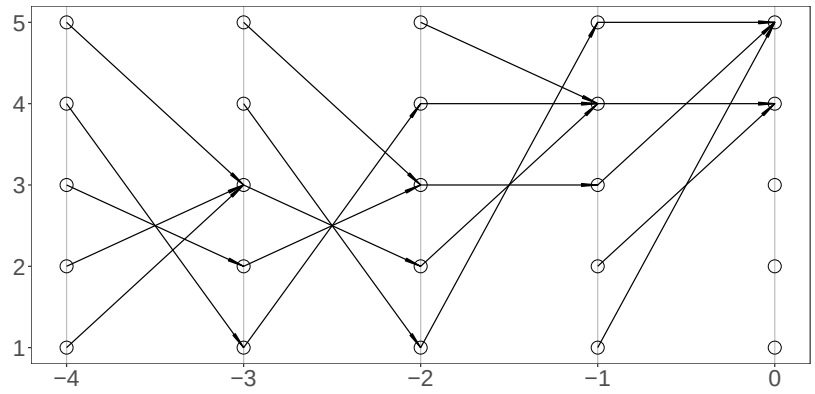

Figure 4.7: Coalescence of the maps $\Psi_1 \circ \Psi_2 \circ \ldots \circ \Psi_t$.

to the geometric decrease of $C \mapsto r^C$. As always we also see that the method is sensitive to the quality of the coupling.

4.4 Coupling from the past

4.4.1 Main idea

In this section we describe the basic idea of "coupling from the past" or CFTP, introduced in Propp and Wilson [1996], undoubtedly the most striking use of couplings in Monte Carlo. See Propp and Wilson [1998] for various references to related methods. We start with a finite state space $\mathbb{X} = \{1, \ldots, n\}$. We will employ the iterated random function representation of the chain, where $X_t = \psi(X_{t-1}, U_t)$, and we will write $\psi(\cdot, U_t)$ as Ψ_t. We assume that (X_t) is aperiodic and irreducible and thus converges to π in distribution. It might help to imagine the infinite sequence of (U_t) variables to be fixed. It will also help to imagine all processes to be indexed by $t \in \mathbb{Z}$ rather than $t \in \mathbb{N}$.

The first thing to recall is that the chain X_t, started from x_0, has the same marginal distribution as

$$Y_t = \Psi_1 \circ \Psi_2 \circ \ldots \circ \Psi_t(x_0),$$

where the random functions $(\Psi_t)_{t \geq 0}$ are applied in a reverse order. The process (Y_t) might reach some value $Y_{T^\star}$ at some "coalescence time" $T^\star$, such that for all $x_0 \in \mathbb{X}$, $\Psi_1 \circ \Psi_2 \circ \ldots \circ \Psi_{T^\star}(x_0) = Y_{T^\star}$. Indeed, by composing with increasingly many functions, the sequence $(\Psi_1 \circ \ldots \circ \Psi_t(\mathbb{X}))$ of sets obtained by applying $\Psi_1 \circ \ldots \circ \Psi_t$ to any $x_0 \in \mathbb{X}$ has a decreasing cardinality, which might eventually reach one.

Furthermore, noting again the order in which functions (Ψ_t) are applied in the construction of (Y_t), the value $Y_{T^\star}$ is also the value of Y_t for all times $t \geq T^\star$.

Thus $Y_{T^\star}$ is equal to $\lim_{t\to\infty} Y_t$. But since Y_t has the same distribution as X_t and since $\lim_{t\to\infty} X_t = \pi$, then $Y_{T^\star}$ must follow the stationary distribution π. In other words $Y_{T^\star}$ is an "exact sample" or "perfect sample" from π. The algorithm is described in Algorithm 3.

Algorithm 3 Coupling from the past.

Set $t \leftarrow 0$.
Set $\Psi_0 = \Psi_{0\to 0} = \mathrm{Id}$, the identity function.
Repeat:

- $t \leftarrow t - 1$,
- draw U_t and set $\Psi_t = \psi(\cdot, U_t)$,
- set $\Psi_{t\to 0} = \Psi_{t+1\to 0} \circ \Psi_t$,

Until $\Psi_{t\to 0}$ is a constant function, i.e. $|\Psi_{t\to 0}(\mathbb{X})| = 1$.
Output $T^\star = t$ and the unique value in the range of that function denoted by $\Psi_{T^\star\to 0}(\mathbb{X})$.

After the above informal description, we need to check two things more seriously: 1) Algorithm 3 terminates with probability one, i.e. $\mathbb{P}(T^\star < \infty) = 1$ with $T^\star = \sup\{t \leq 0 : |\Psi_{t\to 0}(\mathbb{X})| = 1\}$, and 2) the output $\Psi_{T^\star\to 0}(\mathbb{X})$ is indeed distributed according to π.

Proof. Let us first check that the algorithm terminates almost surely. Let's assume that $K^k(x \to y) > 0$ for some $k \geq 1$ and all $x, y \in \mathbb{X}$, in other words the chain is irreducible. If it was not the case (X_t) could not converge to π. Then the composition of k random functions, $\Psi_1 \circ \ldots \circ \Psi_k$, might have a positive chance of "coalescence": $|\Psi_1 \circ \ldots \circ \Psi_k(\mathbb{X})| = 1$ with positive probability. In that case coalescence will occur with Geometric tails, exactly as the first success in a sequence of independent coin flips. However note that this must be an assumption put on the random functions $\psi(\cdot, U)$: for the same kernel K some random functions might coalesce when composed, and some might not. Regarding the correctness of the output, we can write succintely

$$\begin{aligned}\|\mathcal{L}(Y_{T^\star}) - \pi\|_{\mathrm{TV}} &= \lim_{t\to\infty} \|\mathcal{L}(Y_t) - \pi\|_{\mathrm{TV}} \\ &= \lim_{t\to\infty} \|\mathcal{L}(X_t) - \pi\|_{\mathrm{TV}} = 0,\end{aligned}$$

provided the forward chain (X_t) converges to π. □

We provide an example of a Markov chain that converges to its invariant distribution π but a natural random function representation does not coalesce. Suppose $\mathbb{X} = \{1, 2\}$ and $K(1 \to 2) = K(2 \to 1) = 1/2$. The Markov chain converges to the uniform distribution on $\mathbb{X}$. Let $\psi(x, u) = \mathbb{1}(u \leq 1/2)x + \mathbb{1}(u > 1/2)(3 - x)$. In other words, $\psi(x, u) = x$ if $u < 1/2$ and otherwise it switches to the other state. Then the state space $\mathbb{X}$ is mapped to itself for all u, and thus

the cardinality of the sequence $(\Psi_1 \circ \ldots \circ \Psi_t(\mathbb{X}))$ is always equal to two. Can you find another random function representation, for the same Markov chain, that would lead to the coalescence of $\Psi_1 \circ \ldots \circ \Psi_t$?

A natural question is why the construction requires backward compositions of random functions, and not forward ones. After all, the forward compositions $\Psi_k \circ \ldots \circ \Psi_1$ have exactly the same probability of coalescing than the backward ones. Thus the time to coalescence would have the same distribution. The issue would be about the distribution of the output not being π. To see this, imagine a Markov kernel K under which some state $x' \in \mathbb{X}$ has a unique predecessor, i.e. $K(x \to x') > 0$ for only one $x \in \mathbb{X}$. Thus, for any random function representation of the kernel, forward coalescence could not occur at the state x', and thus x' could not be the output of the algorithm, even though we might have $\pi(x') > 0$. One can check that with the backward construction of Algorithm 3, such a state x' can be generated as an output.

Note that in Section 4.3.2 an assumption was added so that we could treat at least one out of C starting states as a stationary draw with some probability. With CFTP that assumption was not formulated, but the method terminates only when meeting occurs from all possible starting states. Thus, there is no magic here: these methods compare the chains to stationary ones either by assumption or by exhaustive calculation.

4.4.2 Implementation and partially ordered spaces

Algorithm 3 might not look directly obvious to implement since it involves functions from $\mathbb{X}$ to $\mathbb{X}$. If $\mathbb{X} = \{1, \ldots, n\}$ is small then functions can be stored explicitly e.g. via storing the image of $\mathbb{X}$ in an array of size n. The identity function corresponds to $\{1, \ldots, n\}$. At each step of the algorithm, some function Ψ_t is drawn which amounts to a set of n values $\{\Psi_t(1), \ldots, \Psi_t(n)\}$. Given the available function $\Psi_{t+1 \to 0}$, also stored as some set $\{\Psi_{t+1 \to 0}(1), \ldots, \Psi_{t+1 \to 0}(n)\}$, the composition is obtained as the set $\{\Psi_{t \to 0}(j) = \Psi_{t+1 \to 0}(\Psi_t(j)), j \in \{1, \ldots, n\}\}$, for a cost of the order of n operations.

The case of small finite state spaces is clearly not the most relevant in the practice of MCMC. The examples of finite state spaces encountered so far in these notes involve huge cardinalities (all possible subsets of a set, possible colourings of a graph, Ising models). Thus we cannot realistically assume that we can store $|\mathbb{X}|$ objects in memory. If we could, then we might probably be able to avoid any MCMC calculations in the first place.

The application of CFTP to non-trivial problems in Propp and Wilson [1996] involves the use of a partial ordering on the state space. We have seen before such an ordering in the example of image restoration [Gibbs, 2004]. Assume that there is a minimal element $\hat{0}$ and a maximal element $\hat{1}$, such that, according to an order $\leq$, for all $x \in \mathbb{X}$ we have both $\hat{0} \leq x$ and $x \leq \hat{1}$. In the case of the Ising model for instance, where $\mathbb{X} = \{-1, +1\}^n$, we can define $x \leq y$ when $x_{(i,j)} \leq y_{(i,j)}$ for all sites (i, j). The ordering is partial because for some $x, x' \in \mathbb{X}$ neither $x \leq x'$ or $x' \leq x$ might hold. In the Ising model, $\hat{0}$ is the state with all

spins equal to -1, while $\hat{1}$ is the state with all spins equal to $+1$. Such a partial ordering is used in Propp and Wilson [1996] as follows.

Suppose that we can find a random function representation $\psi(\cdot, U)$ of the kernel such that $x \leq y$ implies $\psi(x, U) \leq \psi(y, U)$ almost surely: the partial ordering is preserved by application of $\psi(\cdot, u)$ for almost all u. Then, the state $\Psi_{t\to 0}(x_0)$ for some arbitrary $x_0 \in \mathbb{X}$ remains "larger" than $\Psi_{t\to 0}(\hat{0})$, and smaller than $\Psi_{t\to 0}(\hat{1})$, for all times t. Therefore, if for some t we observe the equality $\Psi_{t\to 0}(\hat{0}) = \Psi_{t\to 0}(\hat{1})$, we can guarantee that $\Psi_{t\to 0}(x_0)$ would also be equal to that state for all $x_0 \in \mathbb{X}$, and thus that t is a coalescence time. We do not have to compute the function values $\Psi_{t\to 0}(x)$ for all $x \in \mathbb{X}$; we just need to compute $\Psi_{t\to 0}(\hat{0})$ and $\Psi_{t\to 0}(\hat{1})$. On the other hand, we cannot recursively compute $\Psi_{t\to 0}(\hat{0})$ and $\Psi_{t\to 0}(\hat{1})$ given $\Psi_{t+1\to 0}(\hat{0})$ and $\Psi_{t+1\to 0}(\hat{1})$ similarly to the recursion in Algorithm 3. For example $\Psi_{t\to 0}(\hat{0})$ is equal to $\Psi_{t+1\to 0}(\Psi_t(\hat{0}))$, and unless $\Psi_t(\hat{0}) = \hat{0}$, we do have access to $\Psi_{t+1\to 0}(\Psi_t(\hat{0}))$ from the previous iteration. Computing $\Psi_{t\to 0}(\hat{0})$ from scratch costs of the order of t operations at step t. Thus a naive implementation of the algorithm would lead to a cost that would be quadratic in the coalescence time $T^\star$. As should be hopefully clear by now, Propp and Wilson [1996] are not the naive type. Thus they propose to compute $\Psi_{t\to 0}(\hat{0})$ and $\Psi_{t\to 0}(\hat{1})$ not at every step t until coalescence, but rather at steps $t = 2^k$ for $k \geq 0$, "doubling the time" at each iteration. The procedure is described in Algorithm 4. The algorithm involves a little bit of apparent "waste" because the coalescence could have occurred earlier than the outputted $T^\star$; it could have been somewhere between $T^\star/2$ and $T^\star$. Propp and Wilson [Section 5, 1996] in fact provides some formal justification of this doubling procedure by showing that the waste is negligible. The section also describes strong link between the mixing time of the chain and the coalescence time obtained with Algorithm 4. Note that the latter is called "monotone" in Propp and Wilson [1996] to emphasize that it relies on an ordering of the state space and on a random function representation that preserves the ordering.

Algorithm 4 Coupling from the past on partially ordered state spaces, with minimal/maximal elements $\hat{0}/\hat{1}$, and time doubling at each iteration.

Set $t = 1$
Repeat:

- draw $U_1, \ldots, U_t$
- compute $\Psi_{t\to 0}(\hat{0})$ and $\Psi_{t\to 0}(\hat{1})$ with $\Psi_{t\to 0} = \Psi_1 \circ \ldots \circ \Psi_t$,
- if $\Psi_{t\to 0}(\hat{0}) = \Psi_{t\to 0}(\hat{1})$ output $T^\star$ as the coalesence time, and $\Psi_{t\to 0}(\hat{0})$ as a draw from π,
- otherwise set $t \leftarrow 2t$.

4.4.3 Some developments

Perfect sampling is now a sub-field of Monte Carlo. Huber [2016] is a book on the topic. Some of the developments could be described here, including Fill [1997], Foss and Tweedie [1998], Häggström and Nelander [1998], Huber [2007], Whiteley et al. [2016] or Blanchet and Chen [2019], Blanchet et al. [2019] for recent, challenging examples.

4.4.4 Perfect sampling and unbiased estimation

We have seen in the previous chapter that unbiased estimators can be obtained in a wide range of MCMC settings, following the pioneering work of Glynn and Rhee [2014]. Above we have seen that perfect sampling can also be achieved in some settings. The previous section describes CFTP, which has inspired many techniques, e.g. Blanchet et al. [2019], Blanchet and Chen [2019] for recent examples. There are other general, active directions towards perfect sampling, see Deligiannidis et al. [2020] for a recent example or Guo and Jerrum [2018], Guo et al. [2019]. Clearly, a perfect sample X from π yields unbiased estimators $h(X)$ of $\pi(h)$ for any π-integrable functions h. Thus perfect sampling appears strictly superior as unbiased estimation.

Thus it is worth wondering about the relation between perfect sampling and unbiased estimation; see Glynn [2016]. At the moment we have some MCMC settings where we know how to obtain unbiased estimators; and some other settings where we know how to get perfect samples. What can we do with a perfect sample that we could not do with an unbiased estimator?

[To be developed!]

[Random variables with bounded range versus unbounded range and how it matters for the computation of error bars.]

4.5 Summary and references

Coupled Markov chains can be implemented in various ways, to obtain pairs of trajectories that "contract" or that exactly coincide after some number of iterations. In turn these can be employed to obtain unbiased estimator of $\pi(h)$, or to obtain upper bounds on $\|\pi - \pi_k\|$ for various distances that have integral probability metric or coupling representations. The methodological use of couplings of Markov chains seems to have started in the 1990s, with the parallel contributions of Propp and Wilson [1996], Johnson [1996], Reutter and Johnson [1995], Johnson [1998], Neal [1999]... who else?

Beyond what's covered here, there are other uses for coupled Markov chains. In a "Markov chain equivalent to control variates", Neal and Pinto [2001] employ such couplings with a chain that targets the distribution π, and another chain that targets an analytically tractable approximation of it. Nüsken and Pavliotis [2019] in the same vein discuss variance reduction with coupled chains. See also Piponi et al. [2020]. What about Nicholls et al. [2012].

Instead of pairs of Markov chains with a time lag, the pioneering work of Neal [1999] employed "circularly coupled Markov chains". Johnson [1998] proposes a diagnostics that employs a pair of chains, one assumed at stationarity and one that regularly restarts from a fixed point in the state space. [expand].

The 1990s have seen the development of Monte Carlo methods that implement couplings of Markov chains to diagnose convergence, and obtain perfect samples and more. This has turned into research directions that continue to bear fruit. These methods involve couplings either many chains, or two chains that are sufficient to yield guarantees on the behavior of many chains.

On the other hand, the current practice of MCMC has not yet been impacted by these ideas as much as one might have hoped; at least in the field of statistics. The currently popular software packages appear to implement methods developed in the MCMC literature before 1990. Has all the recent stuff been mostly useless?

Bibliography

S. Agapiou, G. O. Roberts, and S. J. Vollmer. Unbiased Monte Carlo: Posterior estimation for intractable/infinite-dimensional models. *Bernoulli*, 24(3):1726–1786, 2018. 11

N. Biswas, P. E. Jacob, and P. Vanetti. Estimating convergence of Markov chains with L-lag couplings. In *Advances in Neural Information Processing Systems 32*, pages 7391–7401. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8958-estimating-convergence-of-markov-chains-with-l-lag-couplings.pdf. 12, 14

J. Blanchet and X. Chen. Perfect sampling of generalized jackson networks. *Mathematics of Operations Research*, 44(2):693–714, 2019. 21

J. Blanchet, J. Dong, and Z. Liu. Exact sampling of the infinite horizon maximum of a random walk over a nonlinear boundary. *Journal of Applied Probability*, 56(1):116–138, 2019. 21

A. E. Brockwell and J. B. Kadane. Identification of regeneration times in mcmc simulation, with application to adaptive schemes. *Journal of Computational and Graphical Statistics*, 14(2):436–458, 2005. 5

G. Deligiannidis, A. Doucet, and S. Rubenthaler. Ensemble rejection sampling. *arXiv preprint arXiv:2001.09188*, 2020. 21

M. Dobson, J. Zhai, and Y. Li. Using coupling methods to estimate sample quality for stochastic differential equations. *arXiv preprint arXiv:1912.10339*, 2019. 15

J. A. Fill. An interruptible algorithm for perfect sampling via markov chains. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 688–695, 1997. 21

G. E. Forsythe and R. A. Leibler. Matrix inversion by a Monte Carlo method. *Mathematical Tables and Other Aids to Computation*, 4(31):127–129, 1950. ISSN 08916837. URL http://www.jstor.org/stable/2002508. 8, 9

S. G. Foss and R. L. Tweedie. Perfect simulation and backward coupling. *Stochastic models*, 14(1-2):187–203, 1998. 21

L. Gajek. Note on unbiased estimability of the larger of two mean values. *Applicationes Mathematicae*, 23(2):239–245, 1995. 10

C. J. Geyer. Practical Markov chain Monte Carlo. *Statistical science*, pages 473–483, 1992. 6

A. L. Gibbs. Convergence in the Wasserstein metric for Markov chain Monte Carlo algorithms with applications to image restoration. *Stochastic Models*, 20(4):473–492, 2004. doi: 10.1081/STM-200033117. URL https://doi.org/10.1081/STM-200033117. 19

M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015. 10

P. W. Glynn. Randomized estimators for time integrals. Technical report, Wisconsin University Madison Mathematics Research Center, 1983. 10

P. W. Glynn. Exact simulation vs exact estimation. In *2016 Winter Simulation Conference (WSC)*, pages 193–205. IEEE, 2016. 21

P. W. Glynn and P. Heidelberger. Bias properties of budget constraint simulations. *Operations Research*, 38(5):801–814, 1990. 8

P. W. Glynn and P. Heidelberger. Analysis of parallel replicated simulations under a completion time constraint. *ACM Transactions on Modeling and Computer Simulations*, 1(1):3–23, 1991. 8

P. W. Glynn and C.-h. Rhee. Exact estimation for Markov chain equilibrium expectations. *Journal of Applied Probability*, 51(A):377–389, 2014. 10, 11, 13, 21

P. W. Glynn and W. Whitt. The asymptotic efficiency of simulation estimators. *Operations Research*, 40(3):505–520, 1992. 9

H. Guo and M. Jerrum. Perfect simulation of the hard disks model by partial rejection sampling. *arXiv preprint arXiv:1801.07342*, 2018. 21

H. Guo, M. Jerrum, and J. Liu. Uniform sampling through the lovász local lemma. *Journal of the ACM (JACM)*, 66(3):1–31, 2019. 21

O. Häggström and K. Nelander. Exact sampling from anti-monotone systems. *Statistica Neerlandica*, 52(3):360–380, 1998. 21

J. Heng and P. E. Jacob. Unbiased Hamiltonian Monte Carlo with couplings. *Biometrika*, 106(2):287–302, 2019. 4

M. Huber. Perfect simulation for image restoration. *Stochastic models*, 23(3): 475–487, 2007. 21

M. L. Huber. *Perfect simulation*. Chapman and Hall/CRC, 2016. 21

P. E. Jacob, F. Lindsten, and T. B. Schön. Smoothing with couplings of conditional particle filters. *Journal of the American Statistical Association*, 0 (0):1–20, 2019. doi: 10.1080/01621459.2018.1548856. URL https://doi.org/10.1080/01621459.2018.1548856. 12

P. E. Jacob, J. O'Leary, and Y. F. Atchadé. Unbiased Markov chain Monte Carlo with couplings. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2020. 4, 8, 12, 13

V. E. Johnson. Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths. *Journal of the American Statistical Association*, 91(433):154–166, 1996. 3, 7, 15, 16, 21

V. E. Johnson. A coupling-regeneration scheme for diagnosing convergence in Markov chain Monte Carlo algorithms. *Journal of the American Statistical Association*, 93(441):238–248, 1998. 3, 21, 22

N. Kahale. Optimal unbiased estimators via convex hulls. *arXiv preprint arXiv:1909.02876*, 2019. 10

J. Kuti. Stochastic method for the numerical study of lattice fermions. *Physical Review Letters*, 49(3):183, 1982. 8

Y. Li and S. Wang. Numerical computations of geometric ergodicity for stochastic dynamics. *arXiv preprint arXiv:2001.02294*, 2020. 15

D. McLeish. A general method for debiasing a Monte Carlo estimator. *Monte Carlo methods and applications*, 17(4):301–315, 2011. 10

R. M. Neal. Circularly-coupled Markov chain sampling. Technical report, Department of Statistics, University of Toronto, 1999. 3, 4, 21, 22

R. M. Neal and R. L. Pinto. Improving Markov chain Monte Carlo estimators by coupling to an approximating chain. 2001. 21

G. K. Nicholls, C. Fox, and A. M. Watt. Coupled MCMC with a randomized acceptance probability. *arXiv preprint arXiv:1205.6857*, 2012. 21

N. Nüsken and G. Pavliotis. Constructing sampling schemes via coupling: Markov semigroups and optimal transport. *SIAM/ASA Journal on Uncertainty Quantification*, 7(1):324–382, 2019. 21

D. Piponi, M. D. Hoffman, and P. Sountsov. Hamiltonian Monte Carlo swindles. *arXiv preprint arXiv:2001.05033*, 2020. 21

J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures & Algorithms*, 9 (1-2):223–252, 1996. 17, 19, 20, 21

J. G. Propp and D. B. Wilson. How to get a perfectly random sample from a generic markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms*, 27(2):170–217, 1998. 17

A. Reutter and V. E. Johnson. *General strategies for assessing convergence of MCMC algorithms using coupled sample paths*, volume 2. Citeseer, 1995. 7, 15, 21

C.-H. Rhee. *Unbiased Estimation with Biased Samplers*. PhD thesis, Stanford University, 2013. URL http://purl.stanford.edu/nf154yt1415. 9

C.-h. Rhee and P. W. Glynn. A new approach to unbiased estimation for SDE's. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pages 1–7. IEEE, 2012. 10

C.-h. Rhee and P. W. Glynn. Unbiased estimation with square root convergence for SDE models. *Operations Research*, 63(5):1026–1043, 2015. 10

J. S. Rosenthal. Parallel computing and Monte Carlo algorithms. *Far east journal of theoretical statistics*, 4(2):207–236, 2000. 8

T. Rychlik. Unbiased nonparametric estimation of the derivative of the mean. *Statistics & probability letters*, 10(4):329–333, 1990. 10

M. Vihola. Unbiased estimators and multilevel monte carlo. *Operations Research*, 66(2):448–462, 2018. doi: 10.1287/opre.2017.1670. URL https://doi.org/10.1287/opre.2017.1670. 10, 13

W. Wagner. Unbiased Monte Carlo evaluation of certain functional integrals. *Journal of Computational Physics*, 71(1):21–33, 1987. 10

W. R. Wasow. A note on the inversion of matrices by random walks. *Mathematical Tables and Other Aids to Computation*, 6(38):78–81, 1952. 8, 9

N. Whiteley, A. Lee, et al. Perfect sampling for nonhomogeneous markov chains and hidden markov models. *The Annals of Applied Probability*, 26(5): 3044–3077, 2016. 21