Symmetry-exploiting lifted MCMC algorithms

STAT 520A Project Report

Kenny Chiu, Tae Yoon (Harry) Lee

Abstract

The idea of lifted inference is based on improving algorithm efficiency by exploiting structure in the probabilistic model. MCMC algorithms that exploit symmetry structure in the model can be collected under the general family of lifted MCMC algorithms. However, there are still notable differences in how symmetry is exploited across these lifted MCMC algorithms. We conduct a literature review of lifted MCMC and identify two distinct classes of lifted MCMC algorithms—orbit MCMC and coset MCMC—which we describe under a group theoretic framework. We also implement, evaluate, and compare a number of lifted MCMC algorithms on a toy example.

1 Introduction

Many problems in statistical inference involve some form of symmetry. For example, the joint distribution of exchangeable observations is invariant to permutations of the observations, and a distribution defined over a sphere may be invariant to 3D rotations. Lifted inference (Poole, 2003) algorithms explicitly exploit these symmetries and are often observed to be more efficient than those that do not. In this project, we conduct a survey of lifted MCMC algorithms that explicitly take advantage of the symmetries to which the target distribution π is invariant. From our review of the literature, we identify two distinct classes of symmetry-exploiting MCMC algorithms which we describe under a group theoretic framework. We examine how each class exploits symmetry and discuss how the classes differ from one another in their approach. We also implement and evaluate two of these lifted MCMC algorithms on a toy example.

The organization of the report is as follows: Section 2 provides a brief background of the group theoretic concepts that we use to unify the different algorithms under a single mathematical framework; Section 3 introduces the two classes of symmetry-exploiting algorithms that we have identified and provides several examples from the literature that fall under each; Section 4 discusses the results of our evaluation of two lifted MCMC algorithms on a toy example; and Section 5 summarizes our findings and discusses possible directions of future work. The appendix contains more details about the MCMC algorithms that we reference, as well as additional experimental results.

2 Group theory

It is convenient to describe symmetry in the language of group theory. A group is a set $\mathcal{G}$ with a binary operator $\circ$ such that the following conditions hold:

1. Associativity. For $g_1, g_2, g_3 \in \mathcal{G}$, $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$.
2. Identity. There is an element $e \in \mathcal{G}$ such that for $g \in \mathcal{G}$, $g \circ e = g$.
3. Inverse. For $g \in \mathcal{G}$, there is an element $g^{-1} \in \mathcal{G}$ such that $g \circ g^{-1} = g^{-1} \circ g = e$ where e is the identity.

Let $\mathcal{X}$ denote a set. $\mathcal{G}$ acts on $\mathcal{X}$ if for $g \in \mathcal{G}$ and $x \in \mathcal{X}$, $g \circ x \in \mathcal{X}$. In the context of MCMC and symmetry, $\mathcal{X}$ denotes the (topological) state space and $\mathcal{G}$ denotes a (topological) symmetry group acting on $\mathcal{X}$. A particular symmetry group is of interest if it induces a structure on $\mathcal{X}$ such that $\mathcal{X}$ can be projected onto a smaller space. For example, if $\mathcal{X}$ is a state space defined over the sphere S^2 and the distribution of $\mathcal{X}$ is invariant to horizontal rotations around a fixed point (i.e., invariant to SO(2), the set of 2D rotations), then it may be possible to collapse $\mathcal{X}$ into a smaller state space defined over the vertical rotation of the sphere to which the distribution is not invariant. Figure 1 illustrates this idea.

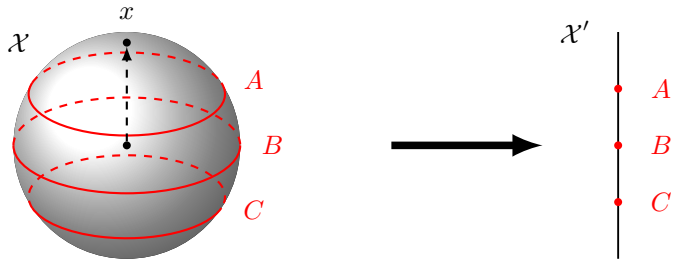

Figure 1: Example state space $\mathcal{X}$ defined over the sphere S^2. If $\mathcal{X}$ is constant within each set A, B, and C of states lying on a particular latitude over the surface of S^2 (red), then it may be possible to collapse $\mathcal{X}$ into a smaller state space $\mathcal{X}'$ defined over the latitude of S^2.

For $x \in \mathcal{X}$, let $\text{Orb}(x) = \{g \circ x \in \mathcal{X} \,|\, g \in \mathcal{G}\}$ denote the *orbit* of x by applying $\mathcal{G}$. Each orbit forms an equivalence class where for $x_1, x_2 \in \mathcal{X}$, $x_1 \sim x_2$ if and only if $x_1 = g \circ x_2$ for some $g \in \mathcal{G}$. Hence the set of $\mathcal{G}$-orbits forms a partition of $\mathcal{X}$, which we denote as $\mathcal{X}/\mathcal{G}$. If $\mathcal{X}$ has only one $\mathcal{G}$-orbit, i.e., $\text{Orb}(x) = \mathcal{X}$ for all $x \in \mathcal{X}$, then $\mathcal{X}$ is a *homogeneous space* for $\mathcal{G}$. Continuing with the sphere example, S^2 is a homogeneous space for SO(3) (the set of 3D rotations), but it is partitioned into an infinite set of parallel orbits over the sphere's surface by SO(2). The sets A, B, and C in Figure 1 show three of the orbits induced by SO(2).

Let $\mathcal{H}$ be a subgroup of $\mathcal{G}$ (a subset $\mathcal{H} \subset \mathcal{G}$ that is also a group). The *quotient space* $\mathcal{G}/\mathcal{H}$ is the set of *left cosets* $\{g \circ \mathcal{H} \,|\, g \in \mathcal{G}\}$ where $g \circ \mathcal{H} = \{g \circ h | h \in \mathcal{H}\}$. Each coset forms an equivalence class where for $g_1, g_2 \in \mathcal{G}$, $g_1 \sim g_2$ if and only if $g_1 = g_2 \circ h$ for some $h \in \mathcal{H}$. Hence the set of left cosets forms a partition of $\mathcal{G}$. One notable subgroup is the *stabilizer subgroup* $\text{Stab}(x) = \{g \in \mathcal{G} \,|\, g \circ x = x\}$ that leaves $x \in \mathcal{X}$ unchanged. In the context of the sphere, if the action of SO(2) does not change $\mathcal{X}$, then the quotient space SO(3)/SO(2) is a partition of SO(3). The sets A, B, and C in Figure 1 can each be mapped to a coset in SO(3)/SO(2).

Whenever necessary, we assume that a group of interest is locally compact so that the group has a left Haar measure which can be normalized to define a probability measure. Discrete groups and Lie groups always have a left Haar measure, but a continuous group may not in general (Folland, 2016).

3 Lifted MCMC

A variety of MCMC algorithms in the literature exploit symmetries in some form and could be considered as lifted MCMC algorithms (Lüdtke et al., 2018). Across these algorithms, we identify two broad classes based on how they take advantage of symmetry. The first class, which we call *orbit MCMC*, takes advantage of the orbit structure of $\mathcal{X}/\mathcal{G}$. The second class, which we call *coset MCMC*, takes advantage of the coset structure of $\mathcal{G}/\mathcal{H}$ induced by some subgroup $\mathcal{H}$. We explain the characteristics of each class in more detail and discuss their differing properties as a result.

3.1 Orbit MCMC

Orbit MCMC algorithms exploit the idea that the space $\mathcal{X}/\mathcal{G}$ of $\mathcal{G}$-orbits of $\mathcal{X}$ can be a potentially much smaller space than $\mathcal{X}$ itself. By reducing the size of the state space, a Markov chain on the space of orbits could have a faster mixing time than a Markov chain on $\mathcal{X}$. We discuss several MCMC algorithms in the literature that explore some variation of this idea.

For permutation groups acting on discrete state spaces, Niepert (2012) introduced a lifted MCMC algorithm that makes use of *orbital Markov chains* [A]. The algorithm augments the model with an auxiliary variable that represents the orbit, and alternates between the kernel of a standard Markov chain and the kernel of an orbital Markov chain. In the first step, the standard Markov kernel samples an $x'_{i+1} \in \mathcal{X}$ given x_i, which could be thought of as sampling a $\mathcal{G}$-orbit in $\mathcal{X}/\mathcal{G}$. The orbital Markov kernel then samples an

$x_{i+1} \in \text{Orb}(x'_{i+1})$ uniformly at random in the following step. Only the samples from the orbital Markov chain are retained. Niepert (2012) showed that if the standard Markov chain is ergodic and the distribution of the states is constant on each $\mathcal{G}$-orbit, the resulting chain has a stationary distribution and is reversible.

den Broeck and Niepert (2014) later extended orbital Markov chains to *orbital Metropolis chains* [B] where after a proposal x_{i+1} is sampled from the orbit $\text{Orb}(x'_{i+1})$, the new proposal is accepted with probability given by a Metropolis ratio. Using the orbital Metropolis chains, den Broeck and Niepert (2014) then introduced *Lifted Metropolis-Hastings* (LMH) [C]. LMH depends on an ergodic base chain similar to how the orbital Markov chain was used in (Niepert, 2012). However, whereas the orbital Markov chain algorithm uses an alternation of the base and orbital Markov kernels, LMH instead uses a mixture of the base kernel and the orbital Metropolis kernel. The mixture in LMH could also be extended to support multiple orbital Metropolis kernels. den Broeck and Niepert (2014) showed that the mixture of an ergodic base chain and an orbital Metropolis chain is ergodic and has a stationary distribution.

While the above algorithms ensure the existence of a stationary distribution, there are no guarantees about the mixing time of the chains. For state spaces where $\mathcal{X}/\mathcal{G}$ and its $\mathcal{G}$-orbits are finite, Holtzen et al. (2019) proposed the *orbit-jump MCMC* (OJ-MCMC) algorithm [D] that has mixing time guarantees bounded by the number of orbits $|\mathcal{X}/\mathcal{G}|$. OJ-MCMC achieves this by producing provably good quality samples at the cost of more expensive iterations. OJ-MCMC is a Metropolis-Hastings algorithm that uses the *uniform orbit distribution* proposal, which is based on the probability of selecting a $\mathcal{G}$-orbit $\text{Orb}(x)$ uniformly at random and then selecting an element $x' \in \text{Orb}(x)$ uniformly at random (given the size of $|\text{Orb}(x)|$). To sample a proposal from the uniform orbit distribution, the algorithm uses the Burnside process (Jerrum, 1992). This process requires computing specific subgroups in each iteration, and is the source of the computational overhead in OJ-MCMC. Holtzen et al. (2019) showed that the total variation distance between the distribution given by the OJ-MCMC chain and the target distribution is proportional to $|\mathcal{X}/\mathcal{G}|$.

We have categorized the above algorithms under orbit MCMC because of how they exploit the structure of $\mathcal{X}/\mathcal{G}$, where the MCMC resembles a random walk over the $\mathcal{G}$-orbits of $\mathcal{X}$. Note that when $\mathcal{X}$ is a homogeneous space for $\mathcal{G}$ (i.e., when there is only one $\mathcal{G}$-orbit), the algorithms reduce to standard MCMC methods on $\mathcal{X}$. Also, while the algorithms are conceptually simple, there is a computational bottleneck in computing $\text{Orb}(x)$ given a general $\mathcal{G}$. The three algorithms that we discussed in this section use the *product replacement algorithm* (Pak, 2000), which computes orbits on discrete spaces in polynomial time given only a generating set of $\mathcal{G}$ (a subset $G \subset \mathcal{G}$ from which any $g \in \mathcal{G}$ can be expressed in terms of combinations of $g' \in G$).

3.2 Coset MCMC

When the state space $\mathcal{X}$ is a homogeneous space for a symmetry group $\mathcal{G}$, coset MCMC algorithms look to construct a smaller quotient space $\mathcal{G}/\mathcal{H}$ generated by some non-trivial subgroup $\mathcal{H}$. As in orbit MCMC, the motivation is to reduce the size of the state space to speed up the mixing time of the chain. However, the group theoretic details tend to be more technical than in orbit MCMC as it is no longer sufficient to assume only the existence of a left Haar measure on $\mathcal{G}$. The space $\mathcal{G}/\mathcal{H}$ must be carefully constructed in order to ensure that it also has a usable measure. We discuss two algorithms in the literature that exploit the quotient space generated by a specific choice of $\mathcal{H}$.

Shariff et al. (2015) introduced a variant of the Metropolis-Hastings algorithm called *Metropolis-Hastings with Group Moves* (MH-GM) [E]. In each iteration, MH-GM samples a group action $g_{i+1} \in \mathcal{G}$ and applies it to the current state x_i to obtain the proposal $g_{i+1} \circ x_i$. The novelty of the proposed algorithm is the acceptance probability of the proposal, which is not derived from a measure over $\mathcal{X}$ but from a measure over the quotient space $\mathcal{G}/\mathcal{H}$ generated by the stabilizer subgroup $\mathcal{H} = \text{Stab}(x_i)$. A generalized version of MH-GM is also introduced for state spaces that are acted on by multiple symmetry groups $\mathcal{G}_1$, $\mathcal{G}_2$, ..., $\mathcal{G}_M$, $M \in \mathbb{N}$. The generalized algorithm simply introduces a mixture component to the original MH-GM algorithm where a group $\mathcal{G}_j$ is first sampled before sampling an action $g_{i+1} \in \mathcal{G}_j$. For both versions of MH-GM, Shariff et al. (2015) showed that the resulting kernel satisfies detailed balance under certain conditions on $\mathcal{G}$, $\mathcal{G}_1$, ..., $\mathcal{G}_M$.

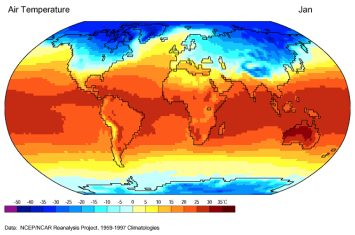

Figure 2: Mean air temperature of the Earth's surface in January, 1959-1997 (Pidwirny and Jones, 2009).

Barp et al. (2019) proposed the *Hamiltonian Monte Carlo on a homogeneous space* (HMC-HS) [F] for a Lie group $\mathcal{G}$. Unlike in MH-GM, HMC-HS considers the quotient space $\mathcal{G}/\mathcal{H}$ generated by the stabilizer subgroup $\mathcal{H} = \text{Stab}(x_0)$ of a fixed point $x_0 \in \mathcal{X}$. When $\mathcal{X}$ is a homogeneous space for $\mathcal{G}$, $x \in \mathcal{X}$ is identifiable only up to the coset $g \circ \mathcal{H} \in \mathcal{G}/\mathcal{H}$ where $x = g \circ x_0$ for $g \in \mathcal{G}$. The key insight is that when $\mathcal{G}/\mathcal{H}$ is reductive (i.e., there is a decomposition of its Lie algebra into orthogonal components), a Hamiltonian system on $\mathcal{G}/\mathcal{H}$ can be constructed from a Hamiltonian system on $\mathcal{G}$ by setting the momentum over $\mathcal{H}$ to be zero. In each HMC-HS iteration, the current "state" $g_i \in \mathcal{G}$ (such that $x_i = g_i \circ x_0$) and a sampled momentum is fed into the reversible symplectic integrator *leapfrog*, which then returns a trajectory of a fixed number of samples. The last sample in the trajectory g_{i+1}^M, $M \in \mathbb{N}$, is taken as the proposal, which the algorithm then accepts with probability given by the Hamiltonian system. Assuming numerical stability, HMC-HS produces a reversible chain as long as the integrator is reversible (Barp et al., 2019).

We have listed the above algorithms under coset MCMC due to how they exploit the structure of $\mathcal{G}/\mathcal{H}$ generated by some specific subgroup $\mathcal{H}$. Note that if $\mathcal{H}$ is taken as the trivial subgroup (the set containing only the identity element), these algorithms reduce to standard MCMC algorithms on $\mathcal{X}$. One detail that is not discussed in the literature is computing $\text{Stab}(x)$ for $x \in \mathcal{X}$ given a general group $\mathcal{G}$. This may be because the coset MCMC algorithms above were introduced for general continuous spaces whereas the orbit MCMC algorithms were introduced for specific discrete spaces, and that there does not appear to be a general algorithm for computing the stabilizer of any continuous group. In this case, computing the stabilizer will depend on the specific problem at hand.

4 Empirical study

Motivated by Figure 2, we construct a toy example where we infer the distribution of the surface temperature defined over a sphere using a naive MCMC algorithm and two lifted MCMC algorithms. We evaluate performance through (1) the effective sample size per number of target evaluations (Flegal et al., 2010) and (2) the RMSE of the density estimated by a histogram across sample sizes. The code for the experiment can be found in our GitHub repository[1].

4.1 Experimental setup

Let $\mathcal{X}$ be a state space defined over the unit sphere S^2 that represents the temperature (in Kelvin) over the Earth's surface. To simplify what is observed in Figure 2, suppose that the temperature is constant on each latitude, i.e., that it is invariant to 2D rotations around the North Pole. We take $\mathcal{G} = \text{SO}(2)$ acting on S^2 as the symmetry group of interest and so each latitude defines a $\mathcal{G}$-orbit.

The objective is to estimate the distribution of the temperature and the mean temperature by using MCMC algorithms to efficiently sample locations (x, y, z) over the surface of the unit sphere. Each location will be mapped to temperature through a deterministic function $f : S^2 \to \mathcal{X}$ defined as $f(x, y, z) = 310 - 80z^2$. The true distribution of the temperature is $310 - 80 \times \text{Beta}(0.5, 1)$, and thus the mean temperature is 283.33.

[1] https://github.com/tyhlee/W20_STAT520A

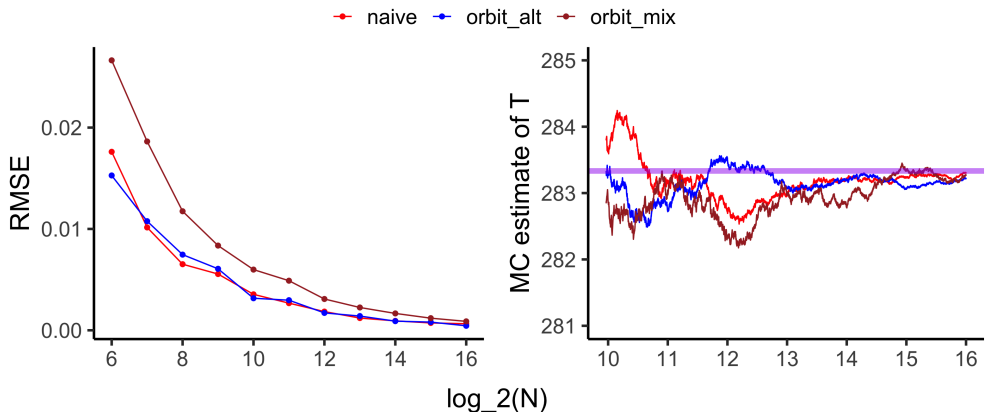

Figure 3: **(Left)** RMSE of the density estimated by a histogram with 100 bins across the sample size. **(Right)** MC estimate of the mean temperature across the sample size. The true mean is 283.33 (purple).

4.2 MCMC algorithms

We compare and evaluate three algorithms. The first algorithm called `naive` is based on sampling uniformly from SO(3). The second algorithm denoted `orbit_alt` uses the orbital Markov chain (Algorithm 1). We use the naive sampling scheme for the base kernel and uniform sampling within-orbit for the orbit kernel. The third algorithm denoted `orbit_mix` is a mixture version of the orbital Markov chain (similar to that of Lifted Metropolis-Hastings but with deterministic acceptance) with a mixture probability of 0.5. The two latter algorithms fall under the category of the orbit MCMC algorithms. For this example, coset MCMC with $\mathcal{G} = \mathrm{SO}(3)$ would look similar to the orbit MCMC algorithms given that the measure over cosets (latitude orbits) is uniform, and hence we do not consider coset MCMC algorithms.

4.3 Experimental results

Figure 3 summarizes the main results. The left plot shows that the RMSE between the true distribution and the histogram estimate of the distribution converges to 0 as the sample size increases for each algorithm. `orbit_mix` has the slowest convergence rate by roughly a factor of 2 compared to that of `naive` and `orbit_alt`. The right plot demonstrates that the mean temperature is well estimated once the chains stabilize after approximately 2^{15} samples. With 2^{16} samples, MC estimates of the mean temperature are 283.34, 283.25, and 283.32 for `naive`, `orbit_alt`, and `orbit_mix`, respectively, with MC standard errors of 21.35, 23.36, and 44.69. Based on the same samples, the ESSs per number of target evaluations of the `naive` algorithm are 65,674, 32,768, and 18,514 for `naive`, `orbit_alt`, and `orbit_mix`, respectively. The `naive` algorithms appears to outperform the lifted MCMC algorithms for this example. Additional results from the experiment can be found in [G].

5 Discussion

We have reviewed several lifted MCMC algorithms in the literature and described each under a framework based on group theory. Using this framework, we identified two distinct classes that aim to improve MCMC efficiency by exploiting symmetry in different ways. The class of orbit MCMC algorithms looks to reduce the size of the state space by working with the space $\mathcal{X}/\mathcal{G}$ of $\mathcal{G}$-orbits. The class of coset MCMC algorithms looks to reduce the size of the state space by working with the quotient space $\mathcal{G}/\mathcal{H}$ generated by some carefully chosen subgroup $\mathcal{H}$. Although the algorithms in both classes tend to look fairly simple when described in terms of group theoretic concepts, the implementation of these algorithms is generally not as straightforward due to the necessary computation of specific (sub)groups.

The results of our empirical study show that the orbital Markov chain and the mixture version do not outperform a naive algorithm on our toy example. These results differ from the original findings of Niepert

(2012) and den Broeck and Niepert (2014). We suspect that the differing context is the main factor, as these lifted MCMC algorithms were originally introduced for inference on discrete factor graphs with permutation groups, whereas our example involves a continuous state space and symmetry group. Our results suggest that in this case, it is more efficient to sample more orbits than to explore each orbit more thoroughly. The orbital Markov chain is less efficient than the naive algorithm as it stays in an orbit for two iterations as opposed to one. The mixture version performs worse, likely due to the mixture kernel allowing the chain to spend possibly multiple iterations within a single orbit.

Note that while we have identified two classes of lifted MCMC, the distinction between the two classes may not be as clear in practice. When working on the space $\mathcal{X}/\mathcal{G}$ in orbit MCMC, each $\mathcal{G}$-orbit is a homogeneous space for $\mathcal{G}$ and so one may then consider a subgroup $\mathcal{H}$ that further partitions the $\mathcal{G}$-orbit by $\mathcal{G}/\mathcal{H}$. When working on the space $\mathcal{G}/\mathcal{H}$ in coset MCMC, each coset could be considered as an $\mathcal{H}$-orbit of $\mathcal{X}$ and so one may think of the space as $\mathcal{X}/\mathcal{H}$. Hence, the distinction between the two classes may ultimately be a difference in perspective of what the "reference" symmetry group $\mathcal{G}$ is. The duality of these two classes may also lead to interesting directions of future work in lifted MCMC where new algorithms combine the two approaches.

Note: please see [H] for a brief discussion about orbital MCMC (Neklyudov and Welling, 2020).

References

Barp, A., A. Kennedy, and M. Girolami (2019). Hamiltonian monte carlo on symmetric and homogeneous spaces via symplectic reduction.

den Broeck, G. V. and M. Niepert (2014). Lifted probabilistic inference for asymmetric graphical models.

Flegal, J. M., G. L. Jones, et al. (2010). Batch means and spectral variance estimators in markov chain monte carlo. *The Annals of Statistics 38*(2), 1034–1070.

Folland, G. (2016). *A Course in Abstract Harmonic Analysis*. Textbooks in Mathematics. CRC Press.

Holtzen, S., T. Millstein, and G. V. den Broeck (2019). Generating and sampling orbits for lifted probabilistic inference.

Jerrum, M. (1992). Uniform sampling modulo a group of symmetries using markov chain simulation. In *Expanding Graphs*.

Lüdtke, S., M. Schröder, F. Krüger, S. Bader, and T. Kirste (2018). State-space abstractions for probabilistic inference: a systematic review. *Journal of Artificial Intelligence Research 63*, 789–848.

Neklyudov, K. and M. Welling (2020). Orbital mcmc.

Neklyudov, K., M. Welling, E. Egorov, and D. Vetrov (2020). Involutive mcmc: a unifying framework.

Niepert, M. (2012). Markov chains on orbits of permutation groups.

Pak, I. (2000). The product replacement algorithm is polynomial. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 476–485.

Pidwirny, M. and S. Jones (2009). Global Surface Temperature Distribution. http://www.physicalgeography.net/fundamentals/7m.html. Accessed: 2021-03-20.

Poole, D. (2003, 10). First-order probabilistic inference. *Proc. IJCAI-03*.

Shariff, R., A. György, and C. Szepesvari (2015, 09–12 May). Exploiting Symmetries to Construct Efficient MCMC Algorithms With an Application to SLAM. In G. Lebanon and S. V. N. Vishwanathan (Eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, Volume 38 of *Proceedings of Machine Learning Research*, San Diego, California, USA, pp. 866–874. PMLR.

Appendix

A Orbital Markov chains (Niepert, 2012)

Let $\mathcal{X}$ be a discrete state space and $\mathcal{G}$ a permutation group acting on $\mathcal{X}$. Let K' be the kernel of an aperiodic and irreducible Markov chain with a stationary distribution, and let K be the kernel of an orbital Markov chain, i.e., a kernel that samples uniformly at random from $\text{Orb}(x)$ when given x. The orbital Markov chain algorithm is then as follows:

Algorithm 1: Lifted MCMC using an orbital Markov chain

input : kernel K' of ergodic Markov chain
input : kernel K of orbital Markov chain
initialize x_0
samples $= \{x_0\}$
for $i = 0 \dots N-1$ **do**
sample $x'_{i+1} \sim K'(x'_{i+1}|x_i)$
sample $x_{i+1} \sim K(x_{i+1}|x'_{i+1})$
samples $\leftarrow$ samples $\cup\, x_{i+1}$
end
output : samples

Note that the orbit sampler $K(x_{i+1}|x'_{i+1})$ only needs to be able to sample from the orbit of x'_{i+1}. The product replacement algorithm Pak (2000) is able to do so from only the generating sets of the permutation group.

B Orbital Metropolis chains (den Broeck and Niepert, 2014)

Let $\mathcal{X}$ be a discrete state space and $\mathcal{G}$ a symmetry group acting on $\mathcal{X}$. Let π be a given distribution and let $Q(x'|x)$ be the conditional proposal distribution that samples uniformly at random from the orbit of a given state. The algorithm for an orbital Metropolis chain is then as follows:

Algorithm 2: Orbital Metropolis chain

input : proposal distribution Q
input : target distribution π
initialize x_0
samples $= \{x_0\}$
for $i = 0 \dots N-1$ **do**
sample $x'_{i+1} \sim Q(x'_{i+1}|x_i)$
$\alpha(x_i, x'_{i+1}) = \min\left\{1, \frac{\pi(x'_{i+1})}{\pi(x_i)}\right\}$
$x_{i+1} = \begin{cases} x'_{i+1} & \text{with probability } \alpha(x_i, x'_{i+1}) \\ x_i & \text{with probability } 1 - \alpha(x_i, x'_{i+1}) \end{cases}$
samples $\leftarrow$ samples $\cup\, x_{i+1}$
end
output : samples

C Lifted Metropolis-Hastings (den Broeck and Niepert, 2014)

Let K' be the kernel of an aperiodic and irreducible Markov chain with a stationary distribution, and let K be the kernel of an orbital Metropolis chain. Let $0 \leq p \leq 1$ be a mixture probability. The Lifted Metropolis-Hastings algorithm is then as follows:

Algorithm 3: Lifted Metropolis-Hastings

input : kernel K' of ergodic Markov chain
input : kernel K of orbital Metropolis chain
input : mixture probability p
initialize x_0
samples $= \{x_0\}$
for $i = 0 \dots N-1$ **do**
sample $x'_{i+1} \sim K'(x'_{i+1}|x_i)$
sample $x''_{i+1} \sim K(x''_{i+1}|x_i)$
$x_{i+1} = \begin{cases} x'_{i+1} & \text{with probability } p \\ x''_{i+1} & \text{with probability } 1-p \end{cases}$
samples $\leftarrow$ samples $\cup\, x_{i+1}$
end
output : samples

D Orbit-jump MCMC (Holtzen et al., 2019)

Let $\mathcal{X}$ be a state space and $\mathcal{G}$ a group acting on $\mathcal{X}$ such that the following conditions hold:

1. The set of $\mathcal{G}$-orbits is finite, i.e., $|\mathcal{X}/\mathcal{G}| < \infty$.
2. Every $\mathcal{G}$-orbit is finite, i.e., $|\mathrm{Orb}(x)| < \infty$ for all $x \in \mathcal{X}$.

The uniform orbit distribution $Q_{\mathcal{X}/\mathcal{G}}$ over $\mathcal{X}$ is defined as the probability of sampling a $\mathcal{G}$-orbit uniformly at random, and then sampling an element from the orbit uniformly at random. The probability of $x \in \mathcal{X}$ under the uniform orbit distribution is given by

$$Q_{\mathcal{X}/\mathcal{G}}(x) = \frac{1}{|\mathcal{X}/\mathcal{G}| \times |\mathrm{Orb}(x)|}$$

The Burnside process is a MCMC method that can be used to sample from $Q_{\mathcal{X}/\mathcal{G}}$. Let $\mathrm{Stab}_{\mathcal{X}}(g) = \{x \in \mathcal{X} \,|\, g \circ x = x\}$ be the set of $x \in \mathcal{X}$ that is fixed by $g \in \mathcal{G}$. A new sample is obtained by repeating the following steps multiple times:

1. Sample $g_{j+1} \sim \mathrm{Stab}(x_j)$ uniformly at random.
2. Sample $x_{j+1} \sim \mathrm{Stab}_{\mathcal{X}}(g_{j+1})$ uniformly at random.

Jerrum (1992) proved that the stationary distribution of the Burnside process is $Q_{\mathcal{X}/\mathcal{G}}$. Note that $\mathrm{Stab}(x)$ and $\mathrm{Stab}_{\mathcal{X}}(g)$ need to be recomputed every iteration. The Burnside process is said to produce good quality samples at the cost of expensive computation.

For a $\mathcal{G}$-invariant target distribution π, the orbit-jump MCMC algorithm is then as follows:

Algorithm 4: Orbit-jump MCMC

input : proposal Q' over $\text{Stab}(x)$
input : proposal Q over $\text{Stab}_{\mathcal{X}}(g)$
initialize x_0
samples $= \{x_0\}$
for $i = 0 \dots N-1$ **do**
  $x_{i+1}^0 = x_i$
  for $j = 0 \dots M-1$ **do**
    sample $g_{i+1}^{j+1} \sim Q'(g_{i+1}^{j+1} | x_{i+1}^j)$
    sample $x_{i+1}^{j+1} \sim Q(x_{i+1}^{j+1} | g_{i+1}^{j+1})$
  end
  $\alpha(x_i, x_{i+1}^M) = \min\left\{1, \frac{\pi(x_{i=1}^M) \times |\text{Orb}(x_{i+1}^M)|}{\pi(x_i) \times |\text{Orb}(x_i)|}\right\}$
  $x_{i+1} = \begin{cases} x_{i+1}^M & \text{with probability } \alpha(x_i, x_{i+1}^M) \\ x_i & \text{with probability } 1 - \alpha(x_i, x_{i+1}^M) \end{cases}$
  samples $\leftarrow$ samples $\cup\, x_{i+1}$
end
output : samples

E Metropolis-Hastings with Group Moves (Shariff et al., 2015)

Let $\mathcal{X}$ be a topological state space that is homogeneous for a topological group $\mathcal{G}$. Then there exists a left Haar measure μ on $\mathcal{G}$ that is left invariant and relatively right invariant, i.e., for $g \in \mathcal{G}$ and subgroup $\mathcal{H} \leq \mathcal{G}$,

$$\mu(g \circ \mathcal{H}) = \mu(\mathcal{H})$$
$$\mu(\mathcal{H} \circ g) = \Delta_{\mathcal{G}}(g)\mu(\mathcal{H})$$

where $\Delta_{\mathcal{G}} : \mathcal{G} \to \mathbb{R}_+$ is the *right modular function* of G. Note that $\Delta_{\mathcal{G}} = 1$ when the subgroup is both left and right invariant, e.g., when $\mathcal{G}$ is compact, discrete, or abelian. There is also a relatively invariant measure λ on $\mathcal{X}$ such that for $\mathcal{V} \subset \mathcal{X}$,

$$\lambda(g \circ \mathcal{V}) = \chi(g)\lambda(\mathcal{V})$$

where $\chi : \mathcal{G} \to \mathbb{R}_+$. In addition, if $\mathcal{G}$ is locally compact, there exists a unique Haar measure β_x on $\text{Stab}(x)$ for $x \in \mathcal{X}$ with $\beta_x(\text{Stab}(x)) = 1$.

Define a group proposal kernel over $\mathcal{G}$ given $x \in \mathcal{X}$ as

$$Q_{\mathcal{G}}(dg|x) = q(g|x)\mu(dg)$$

for some density q, and define q' as the density over the stabilizer of the current state, i.e.,

$$q'(g|x) = \int_{\text{Stab}(x)} q(g \circ h|x)\beta_x(dh)$$

Assume that the target distribution satisfies

$$\pi(dx) = p(x)\lambda(dx)$$

for some density p. The Metropolis-Hastings with Group Moves algorithm is then as follows:

Algorithm 5: Metropolis-Hastings with Group Moves

input : proposal distribution $Q_{\mathcal{G}}$
input : densities q', p
input : functions χ, $\Delta_{\mathcal{G}}$
initialize x_0
samples $= \{x_0\}$
for $i = 0 \dots N-1$ **do**
sample $g_{i+1} \sim Q_{\mathcal{G}}(g_{i+1}|x_i)$
$\alpha(x_i, g_{i+1}) = \min\left\{1, \frac{\chi(g_{i+1})p(g_{i+1} \circ x_i)q'(g_{i+1}^{-1}|g_{i+1} \circ x_i)}{\Delta_{\mathcal{G}}(g_{i+1})p(x_i)q'(g_{i+1}|x_i)}\right\}$
$x_{i+1} = \begin{cases} g_{i+1} \circ x_i & \text{with probability } \alpha(x_i, g_{i+1}) \\ x_i & \text{with probability } 1 - \alpha(x_i, g_{i+1}) \end{cases}$
samples $\leftarrow$ samples $\cup\, x_{i+1}$
end
output : samples

Note that this algorithm generalizes to standard Metropolis-Hastings when $g \circ x = g + x$.

Let $\mathcal{G}_1$, $\mathcal{G}_2$, ..., $\mathcal{G}_M$, $M \in \mathbb{N}$, be groups acting on $\mathcal{X}$, each with a proposal kernel $Q_j(dg_j|x)$, $j \in \{1, 2, ..., M\}$. Let λ be a measure on $\mathcal{X}$ that is simultaneously relatively invariant under all groups, i.e., χ_j-relatively invariant under each $\mathcal{G}_j$. Let $a(j|x) > 0$ be mixture coefficients such that $\sum_{j=1}^{M} a(j|x) = 1$ for all $x \in \mathcal{X}$. The Metropolis-Hastings with Group Moves algorithm generalized to multiple groups is then as follows:

Algorithm 6: Metropolis-Hastings with Group Moves for multiple groups

input : densities q', p
input : functions χ, $\Delta_{\mathcal{G}}$
initialize x_0
samples $= \{x_0\}$
for $i = 0 \dots N-1$ **do**
sample $j_{i+1} \sim a(j_{i+1}|x_i)$
sample $g_{i+1} \sim Q_{j_{i+1}}(g_{i+1}|x_i)$
$\alpha(x_i, g_{i+1}) = \min\left\{1, \frac{\chi_{j_{i+1}}(g_{i+1})a(j_{i+1}|g_{i+1} \circ x_i)p(g_{i+1} \circ x_i)q'_{j_{i+1}}(g_{i+1}^{-1}|g_{i+1} \circ x_i)}{\Delta_{\mathcal{G}}a(j_{i+1}|x_i)(g_{i+1})p(x_i)q'_{j_{i+1}}(g_{i+1}|x_i)}\right\}$
$x_{i+1} = \begin{cases} g_{i+1} \circ x_i & \text{with probability } \alpha(x_i, g_{i+1}) \\ x_i & \text{with probability } 1 - \alpha(x_i, g_{i+1}) \end{cases}$
samples $\leftarrow$ samples $\cup\, x_{i+1}$
end
output : samples

The kernel for the Metropolis-Hastings with Group Moves algorithm satisfies detailed balance under assumptions 1 and 2 below, and the kernel for the generalized algorithm satisfies detailed balance under all three. The assumptions are as follows:

1. $\mathcal{X}$, $\mathcal{G}$, and $\mathcal{G}_1$, ..., $\mathcal{G}_M$ are locally compact and Hausdorff.
2. The action of $\mathcal{G}$, $\mathcal{G}_j$ on $\mathcal{X}$ preserves compactness.
3. The image of $\mathcal{X}$ under $\mathcal{G}_{j_1}$, $\mathcal{G}_{j_2}$, $j_1 \neq j_2$, overlap negligibly. In other words, for $x \in \mathcal{X}$, the condition

$$p(x) \int \mathbb{1}\left[g \in (\mathcal{G}_{j_1} \cap \mathcal{G}_{j_2}) \circ \mathrm{Stab}_j(x)\right] q'(g|x)\mu_j(dg) = 0$$

is satisfied for either $j = j_1$ or $j = j_2$.

F Hamiltonian Monte Carlo on homogeneous spaces (Barp et al., 2019)

Let $\mathcal{X}$ be a manifold that is a homogeneous space for a Lie group (differentiable manifold) $\mathcal{G}$. Denote the Lie algebra (vector space) of $\mathcal{G}$ as $\mathfrak{g}$ and denote its adjoint representation as $\mathrm{Ad}_{\mathcal{G}}(\cdot)$. Let $\mathcal{H} = \mathrm{Stab}(x)$ be the stabilizer subgroup of $\mathcal{G}$ for some specific point $x \in \mathcal{X}$. Assume that the quotient space $\mathcal{G}/\mathcal{H}$ is reductive, i.e., that there is a decomposition $\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{k}$ such that $\mathrm{Ad}_{\mathcal{G}}(\mathcal{H}) \circ \mathfrak{k} = \mathfrak{k}$. Let $\phi : \mathcal{G} \to \mathcal{G}/\mathcal{H}$ be the canonical projection function that maps $g \in \mathcal{G}$ to its coset $g \circ \mathcal{H} \in \mathcal{G}/\mathcal{H}$.

Let $\tilde{V} : \mathcal{X} \to \mathbb{R}$ be a potential energy over $\mathcal{X} \cong \mathcal{G}/\mathcal{H}$, and define $V = \tilde{V} \circ \phi$ as an extended potential over $\mathcal{G}/\mathcal{H}$. Consider a basis for $\mathfrak{k}$ such that the matrix associated to the $\mathrm{Ad}_{\mathcal{G}}(\cdot)$ $\mathcal{H}$-invariant inner product on $\mathfrak{k}$ is $T = \frac{1}{2}I$ where I is the identity matrix. Define the Hamiltonian on $\mathcal{G} \times \mathfrak{g}$ as $H = V + T$; V corresponds to the density of the target distribution on $\mathcal{G}/\mathcal{H}$, while T (kinetic energy) corresponds to the density over the auxiliary variable, momentum P. Let $\rho : \mathcal{G} \to \mathrm{GL}(n)$ be an injective representation of $g \in \mathcal{G}$ as non-singular n-dimensional matrices. Let p be a distribution with respect to the measure $e^{-T}\lambda$ where λ is the Lebesgue measure.

The leapfrog integrator is used as a reversible symplectic integrator to approximate the Hamiltonian trajectories. The Hamiltonian Monte Carlo algorithm on homogeneous spaces is then as follows:

Algorithm 7: Hamiltonian Monte Carlo on homogeneous spaces

input : canonical projection ϕ
input : distribution p
input : Hamiltonian H
input : time step size δt
input : trajectory length L

initialize ρ_0
sample $P_0 \sim p$
$x_0 = \phi(\rho_0)$
samples $= \{x_0\}$
$M = \lfloor L/\delta t \rfloor$
for $i = 0 \dots N-1$ **do**
  $\rho_{i+1}^0 = \rho_i$
  sample $P_{i+1}^0 \sim p$
  $\{(\rho_{i+1}^j, P_{i+1}^j)\}_{j=1}^M = \mathrm{Leapfrog}(\rho_{i+1}^0, P_{i+1}^0, \delta t, L)$
  $\alpha((\rho_i, P_i), (\rho_{i+1}^M, P_{i+1}^M)) = \min\left\{1, \exp\left(-H(\rho_{i+1}^M, P_{i+1}^M) + H(\rho_i, P_i)\right)\right\}$
  $(\rho_{i+1}, P_{i+1}) = \begin{cases} (\rho_{i+1}^M, P_{i+1}^M) & \text{with probability } \alpha((\rho_i, P_i), (\rho_{i+1}^M, P_{i+1}^M)) \\ (\rho_i, P_i) & \text{with probability } 1 - \alpha((\rho_i, P_i), (\rho_{i+1}^M, P_{i+1}^M)) \end{cases}$
  $x_{i+1} = \phi(\rho_{i+1})$
  samples $\leftarrow$ samples $\cup\, x_{i+1}$
end

output : samples

G Additional experimental results

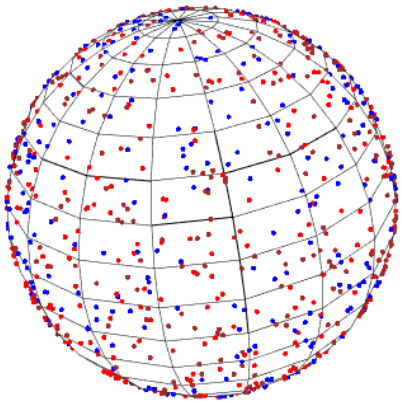

Figure 4: Samples from the naive (red), orbital Markov chain (blue), and mixture orbital Markov chain (brown) algorithms.

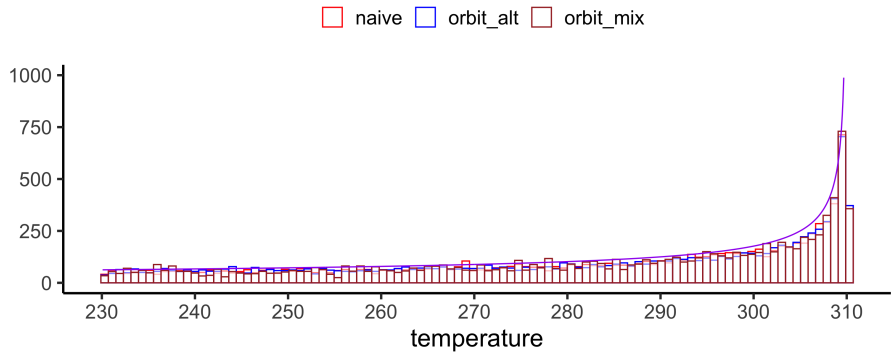

Figure 5: Histogram of the temperature based on 10,000 samples from the naive (red), orbital Markov chain (blue), and mixture orbital Markov chain (brown) algorithms. The true distribution is shown in purple.

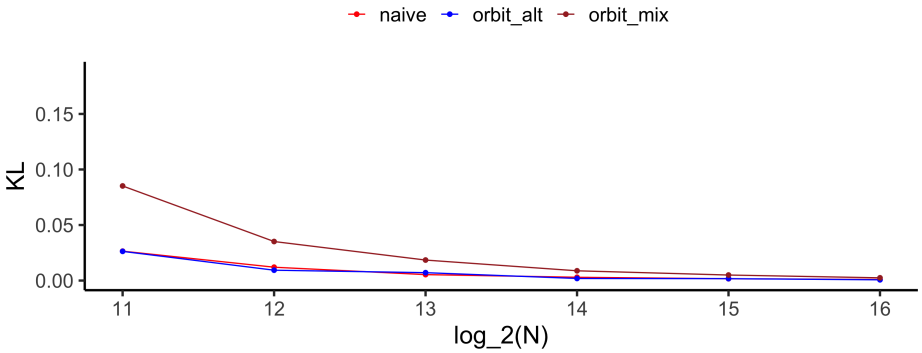

Figure 6: Forward KL divergence of the distribution estimated by the naive (red), orbital Markov chain (blue) and mixture orbital Markov chain (brown) algorithms.

H Orbital MCMC (Neklyudov and Welling, 2020)

We include a brief discussion of *Orbital MCMC* (Neklyudov and Welling, 2020) (oMCMC) which is not referenced in our report. This is because despite the name *orbital* MCMC, oMCMC does not actually exploit symmetry in the way that we have defined it under our group theoretic framework, and hence we do not consider it to be a lifted MCMC algorithm.

oMCMC extends the idea of *involutive MCMC* (Neklyudov et al., 2020) (iMCMC), which is a framework that is able to describe many MCMC algorithms by specifying (1) a deterministic function f that outputs a proposal based on the current state and an auxiliary variable, and (2) an acceptance probability for the proposal. The main advantage of including a deterministic function f is that it can be learned, and that deterministic Markov chains can have high mixing rates. However, Neklyudov et al. (2020) showed that in order for the resulting chain to be reversible, f must be involution satisfying $f = f^{-1}$. oMCMC extends iMCMC by allowing f to simply be a bijection, and instead samples proposals by repeated application of f given the current state and a sampled auxiliary variable. This can be viewed as exploring the orbits of f, and the auxiliary variables are used to transition between orbits.

The original oMCMC algorithm that uses the *escaping orbital kernel* (Neklyudov and Welling, 2020) is as follows:

Algorithm 8: Escaping oMCMC

input : target density $p(x)$
input : density $p(v|x)$ and a sampler from $p(v|x)$
input : continuous bijection $f(x, v)$
initialize x
for $i = 0 \dots N-1$ **do**
sample $v_{i+1} \sim p(v_{i+1}|x_i)$
propose $(x'_{i+1}, v'_{i+1}) = f(x_i, v_{i+1})$
$\alpha(x_i, v_{i+1}) = \inf_{k \in \mathbb{Z}} \left\{ \frac{p(f^k(x_i, v_{i+1}))}{p(x_i, v_{i+1})} \left| \frac{\partial f^k}{\partial [x, v]} \right| \right\}$
$x_{i+1} = \begin{cases} x'_{i+1} & \text{with probability } \alpha(x_i, v_{i+1}) \\ x_i & \text{with probability } 1 - \alpha(x_i, v_{i+1}) \end{cases}$
samples $\leftarrow$ samples $\cup\, x_{i+1}$
end
output : samples

Although oMCMC does not fall under our framework, oMCMC is able to describe a general MCMC algorithm that resembles coset MCMC. If we take the space of v to be $\mathcal{G}$ for which $\mathcal{X}$ is a homogeneous space and let $f(x, v) = v \circ x$, then exploring the f-orbits reduces to exploring the cosets generated by some stabilizer subgroup, and the algorithm reduces to a coset MCMC algorithm with a particular choice for the acceptance probability. Note that it may also be possible to frame orbit MCMC under oMCMC by specifying f to traverse within $\mathcal{G}$-orbits, and taking the space of v as $\mathcal{X}$ to traverse between orbits. However, this is less intuitive than coset MCMC as the within-orbit sampling would then be deterministic.