Teleport annealing

Andrew R. Barron, W. D. Brinda, Jason M. Klusowski,
and Dylan O'Connell

1 Introduction

Markov Chain Monte Carlo (MCMC) algorithms generally have a target distribution as a steady state; they are guaranteed to approach their steady states and thus produce approximate draws from the desired distribution *eventually*. The problem is that there are typically no guarantees on how long it will take before the process is approximately distributed according to the target. In particular, in high-dimensional multimodal distributions, it can be practically impossible for common MCMC techniques to move from one mode to another. Simulated annealing can help a chain explore a variety of modes, but each chain still ends up stuck in some mode. One can build both large and small steps into a technique, but in high dimensions large random steps are unlikely to land anywhere desirable.

The *evolutionary sampling* algorithm of Xie et al. [2015] runs multiple chains simultaneously and gives each chain opportunities to teleport to the location of another. In that algorithm, the teleportations are governed by Metropolis-Hastings probabilities. Our proposed algorithm also runs simultaneous chains, but we prescribe novel teleportation probabilities that guide the chains' distribution along a planned annealing path from an initializing distribution to the target.

Theorem 1.1 provides the key insight.

Theorem 1.1. *Suppose V and V' are independent $\mathcal{V}$-valued random variables both draw from probability density q, and let r be a probability density on $\mathcal{V}$. Given V and V', generate $B \sim Bern(a + \frac{r(V)-q(V)}{q(V)})$ assuming $a + \frac{r(V)-q(V)}{q(V)} \in [0,1]$. Then the random variable $\widetilde{V} := BV + (1-B)V'$ has marginal distribution R.*

Proof. The proof is easiest to understand for discrete $\mathcal{V}$ with q and r as proba-

bility mass functions.

$$\begin{aligned}
\mathbb{P}\{\widetilde{V} = v\} &= \mathbb{P}\{V = v \cap B = 1\} + \mathbb{P}\{V' = v \cap B = 0\} \\
&= \mathbb{P}\{V = v\}\mathbb{P}\{B = 1 | V = v\} + \mathbb{P}\{V' = v\}\mathbb{P}\{B = 0\} \\
&= q(v)(a + \tfrac{r(v)-q(v)}{q(v)}) + q(v) \sum_{v' \in \mathcal{V}} q(v')(1 - a - \tfrac{r(v')-q(v')}{q(v')}) \\
&= aq(v) + r(v) - q(v) + (1-a)q(v) - q(v) \sum_{v' \in \mathcal{V}} (r(v') - q(v')) \\
&= r(v)
\end{aligned}$$

The logic extends beyond the case of discrete $\mathcal{V}$ if mathematical care is taken. □

If the current and target densities are in a smoothly time-parametrized family $\{p_t\}$, we can approximate the crucial Bernoulli quantity by using

$$\frac{p_{t+h}(v) - p_t(v)}{p_t(v)} \approx h \underbrace{\tfrac{\partial}{\partial t} \log p_t(v)}_{\text{“}\delta_v(t)\text{”}} \tag{1}$$

for small enough h. Assume p_t is a probability mass function and that w_t is proportional to it.

$$\begin{aligned}
\delta_v(t) &:= \tfrac{\partial}{\partial t} \log p_t(v) \\
&= \tfrac{\partial}{\partial t} \log \frac{w_t(v)}{\sum_{v' \in \mathcal{V}} w_t(v')} \\
&= \tfrac{\partial}{\partial t} \log w_t(v) - \frac{1}{\sum_{v' \in \mathcal{V}} w_t(v')} \sum_{v'' \in \mathcal{V}} \tfrac{\partial}{\partial t} w_t(v'') \\
&= \tfrac{\partial}{\partial t} \log w_t(v) - \sum_{v'' \in \mathcal{V}} \frac{w_t(v'')}{\sum_{v' \in \mathcal{V}} w_t(v')} \tfrac{\partial}{\partial t} \log w_t(v'')
\end{aligned}$$

The second term is the expectation of the first according to the weights p_t. This reveals the advantage of the approximation (1): rather than calculating the normalizing factor, we can estimate the derivative of its logarithm. This trick is not limited to discrete distributions, as long as the derivative and integral can be interchanged.

Section 2 spells out the teleport annealing algorithm based on these ideas. Next, Section 3 provides simulations to compare the algorithm to Metropolis-Hastings sampling in a generic context. Section 4 provides an example of the algorithm using Gibbs sampling with internal annealing to approximate the posterior in Gaussian mixture modeling. Finally, in Section 5 we discuss practical issues and indicate how teleport annealing might yield provable statistical guarantees in some contexts.

2 The algorithm

Let $p : \mathcal{V} \to \mathbb{R}$ be a probability density with respect to μ, and suppose that q is proportional to p.

The first task is to devise a w_t such that w_0 is a probability density with respect to μ that can be sampled exactly and $w_1 = q$. Additionally, the logarithm of w_t must be differentiable for each $v \in \mathcal{V}$. Define $p_t := \frac{w_t}{\int_{\mathcal{V}} w_t(v) d\mu(v)}$. Draw M elements independently according to w_0 (which equals p_0) to serve as initializations for M separate chains. For each chain, calculate the derivative with respect to t of $\log w_t$ at that chain's current location. Evaluate these M functions at $t = 0$, compute the average of these M quantities, then subtract that average from each of the quantities. The result is an estimate of the chains' δ quantities [defined in (1)] at $t = 0$. Each chain should be kept as is with probability .5 plus h times that chain's estimated δ; otherwise, the chain's value should be overwritten with the value of another chain that is chosen uniformly at random. The new values represent the chains at time $t = h$. Continue to alternate between estimating δ values and teleporting chains until $t = 1$, while inserting MCMC steps along the way to weaken dependence among chains and to counteract the error that arises from the approximation (1) and from estimating the expected derivative of $\log w_t$.

With $1/h \in \mathbb{N}$, the algorithm can be stated more formally as follows.

Algorithm 1: Teleport annealing

$t \leftarrow 0$;
generate $V_1^{(0)}, \ldots, V_M^{(0)} \overset{iid}{\sim} w_0$;
while $t < 1$ **do**
  for i *in* $\{1, \ldots M\}$ **do**
    $d_i \leftarrow \frac{\partial}{\partial \tau} \log w_\tau(V_i^{(t)})$ evaluated at t;
  $\bar{d} \leftarrow \frac{1}{M} \sum_i d_i$;
  for i *in* $\{1, \ldots M\}$ **do**
    $\hat{\delta} \leftarrow d_i - \bar{d}$;
    generate $U \sim \text{Unif}[0, 1]$;
    if $U \leq .5 + h\hat{\delta}$ **then**
      $V_i^{(t+h)} \leftarrow V_i^{(t)}$;
    else
      generate Z uniformly from $\{1, \ldots, i-1, i+1, \ldots, M\}$;
      $V_i^{(t+h)} \leftarrow V_Z^{(t)}$;
  take MCMC steps toward p_t with each chain;
  $t \leftarrow t + h$;
return $V_1^{(1)}, \ldots, V_M^{(1)}$

Next, we will describe two specific instances of the algorithm and see how they perform in simulations.

3 Use with Metropolis-Hastings sampling

Let q be proportional to a target density p. Define $w_t := f^{1-t}q^t$ where f is a density that can be easily sampled.[1] Then $w_0 = f$ and $w_1 = q$. A simulated annealing version of Metropolis-Hastings would draw an initializer according to f, then gradually increase t from 0 to 1 while taking steps with probability equal to the ratio of proposed points' w_t values to current points' w_t values.

For teleport annealing, the following quantity is needed to calculate teleportation probabilities.

$$\frac{\partial}{\partial t} \log w_t = \log q - \log f$$

Let us demonstrate by sampling from a "spiraling" Gaussian mixture in $\mathbb{R}^d$ with density proportional to

$$q(x) = \sum_{j=1}^{d} j e^{-\frac{1}{d^2/18}\|x - je_j\|^2}. \tag{2}$$

where e_j represents the jth canonical basis unit vector.[2] Notice that the weights increase as the components spiral outward. This example is designed to be challenging for ordinary MCMC methods: we will initialize with $N(0, d/2)$, so initialization points will land in the low-mass modes more often than they land in the high-mass modes. As a result, a disproportionate number of chains are likely to get caught in low-mass modes. We will see if teleport annealing can overcome the problem.

The teleport annealing with have $10,000$ chains, use $h = .01$, and take at each time increment, one teleportation step followed by one Metropolis-Hastings step. Another $10,000$ ordinary Metropolis-Hastings chains will also follow an annealing path. The only difference between these two algorithms is the inclusion of a teleportation step. We also perform an addition run Metropolis-Hastings for 400 steps from $t = 0$ to $t = 1$ rather than 100 in order to approximately equalize the time taken by teleport annealing and ordinary Metropolis-Hastings annealing. The size of Metropolis-Hastings proposal move is s times $N(0, I_d)$, and we will look at a range of s values.

To compare the algorithms, we calculate the proportion of the $10,000$ chains that end up closest to each component mean. This is approximately the proportion of chains that have ended up stuck in that mode.

For dimension 2, Figure 1 shows a true sample, a teleport annealing sample with 100 annealing steps, a Metropolis-Hastings sample with 100 annealing steps, and a Metropolis-Hastings sample with 400 annealing steps, the latter three algorithms using $s = 1$. (For visual clarity, only 500 out of the $10,000$ chains are plotted.) Boxplots in Figure REF show the results of 40 replications of this experiment; it indicates that teleport annealing tends to have about

[1] The family of distributions proportional to w_t is the one-dimensional exponential family that passes through f and p.

[2] Of course, it is easy to sample from this mixture directly.

the right number of chains in each mode, while ordinary Metropolis-Hastings without teleporting puts too few chains in the larger mode. When Metropolis-Hastings takes extra steps to run for as long as teleport annealing, it improves but still falls short.

More generally, we quantify the mismatch between each sample and the true component weights of approximately $\frac{1}{\sum_{j=1}^{d} j}(1, \ldots, d)$ by calculating a χ^2-statistic. For each $s \in \{.5, 1, 1.5, 2, 2.5\}$, the experiment is repeated forty times with $d \in \{2, 3, 4\}$, and the quartiles of the results are plotted in Figure REF. The χ^2-statistic quartiles for true samples of size $10,000$ are included as a baseline for comparison.

4 Use with Gibbs sampling

Consider the k-component Gaussian mixture model in which each component has covariance $\sigma^2 I_d$. With data modeled as iid, the likelihood can be written as a sum of k^n product terms.

$$\begin{aligned}
\prod_{i=1}^{n} p_\theta(X_i) &\propto \prod_{i=1}^{n} \sum_{j=1}^{k} e^{-\frac{1}{2\sigma^2}\|X_i - \mu_j\|^2} \\
&= \sum_{v \in \mathcal{V}} \prod_{i=1}^{n} e^{-\frac{1}{2\sigma^2}\|X_i - \mu_{v_i}\|^2} \\
&= \sum_{v \in \mathcal{V}} \prod_{j=1}^{k} e^{-\frac{1}{2\sigma^2}\left[n_{v,j}\|\mu_j - \overline{X}_{v,j}\|^2 + \sum_{i:v_i=j}\|X_i - \overline{X}_{v,j}\|^2\right]}
\end{aligned}$$

where $\mathcal{V} = \{1, \ldots, k\}^n$ indexes the set of all k^n possible assignments of labels, $v = (v_1, \ldots, v_n)$ denotes a labeling by having each $v_i \in \{1, \ldots, k\}$, $n_{v,j}$ denotes the number of observations that labeling v assigns to to label j, and $\overline{X}_{v,j}$ is the mean of the observations with label j according to labeling v.

In a Bayesian analysis with independent Normal priors $\mu_j \sim N(\alpha_j, \frac{\sigma^2}{\beta_j} I_d)$,

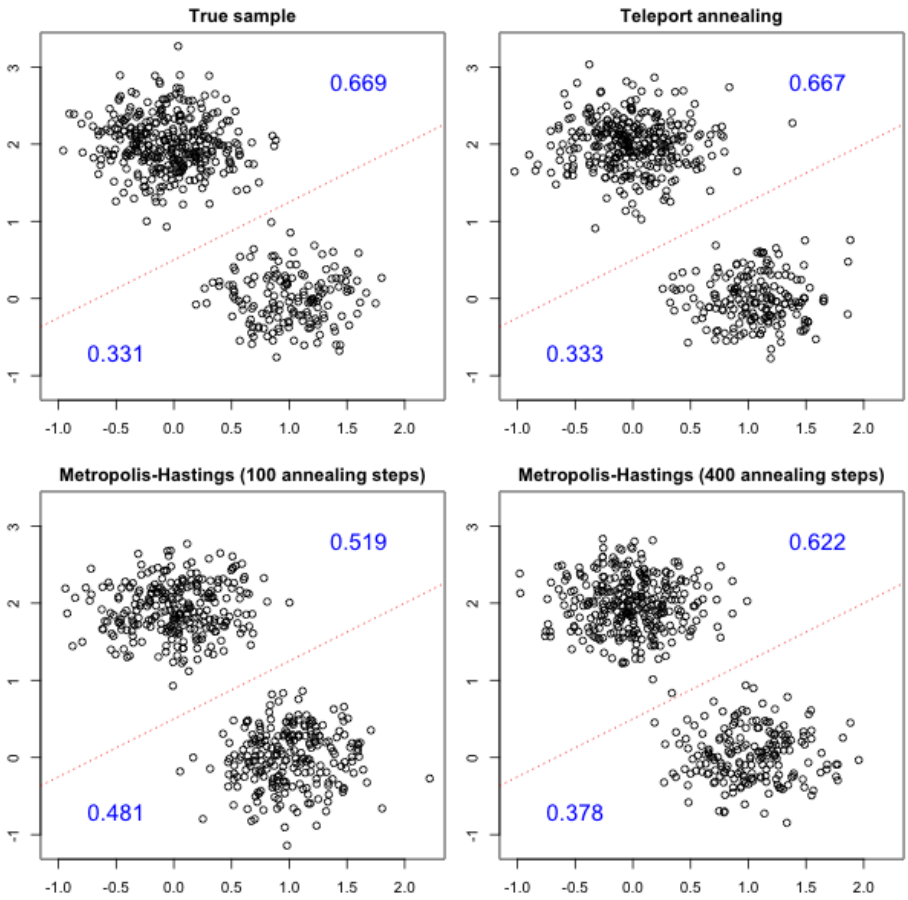

Figure 1: A sample of size $10,000$ was taken from the spiraling Gaussian mixture (2). Another $10,000$ draws were generated using teleport annealing with 100 annealing steps, Metropolis-Hastings with 100 annealing steps, and Metropolis-Hastings with 400 annealing steps. For each method, the proportion of points closer to each mode is reported, and 500 of the points are plotted.

the posterior can be represented as a Gaussian mixture with k^n components.

$$\prod_{j=1}^{k} e^{-\frac{\beta_j}{2\sigma^2}\|\mu_j-\alpha_j\|^2} \prod_{i=1}^{n} p_\theta(X_i) \propto \sum_{v\in\mathcal{V}} \prod_{j=1}^{k} e^{-\frac{1}{2\sigma^2}[n_{v,j}\|\mu_j-\overline{X}_{v,j}\|^2+\sum_{i:v_i=j}\|X_i-\overline{X}_{v,j}\|^2+\beta_j\|\mu_j-\alpha_j\|^2]}$$

$$= \sum_{v\in\mathcal{V}} \underbrace{\left(\prod_{j=1}^{k} \frac{1}{(\beta_j+n_{v,j})^{d/2}} e^{-\frac{1}{2\sigma^2}[\frac{\beta_j n_{v,j}}{\beta_j+n_{v,j}}\|\overline{X}_{v,j}-\alpha_j\|^2+\sum_{i:v_i=j}\|X_i-\overline{X}_{v,j}\|^2]}\right)}_{w(v)}$$

$$\times \underbrace{\left(\prod_{j=1}^{k} (\beta_j+n_{v,j})^{d/2} e^{-\frac{(\beta_j+n_{v,j})}{2\sigma^2}\|\mu_j-\widetilde{\mu}_{v,j}\|^2}\right)}_{f_v(\theta)} \tag{3}$$

with $\widetilde{\mu}_{v,j} := \frac{\beta_j}{\beta_j+n_{v,j}}\alpha_j + \frac{n_{v,j}}{\beta_j+n_{v,j}}\overline{X}_{v,j}$. A draw from (3) would be achieved by drawing a labeling according to the weights proportional to $\{w(v) : v \in \mathcal{V}\}$ then drawing $\theta = (\mu_1, \ldots, \mu_k)$ from the Gaussian density proportional to f_v.

We will use a parallel annealing path sampling algorithm to guide weights toward those in (3). First, we need to devise a time-parameterization. In order to be able to do appropriate Gibbs sampling along the way, we use an *internal annealing* approach. Consider the family of Gaussian mixture models with density proportional to

$$p_\theta^{(t)}(X_i) = \sum_{j=1}^{k} e^{-\frac{1}{2\sigma^2}[(1-t)\|X_i\|^2+t\|X_i-\mu_j\|^2]}.$$

As in (3), we express the resulting posterior as a mixture

$$\prod_{j=1}^{k} e^{-\frac{\beta_j}{2\sigma^2}\|\mu_j-\alpha_j\|^2} \prod_{i=1}^{n} p_\theta^{(t)}(X_i) \propto \sum_{v\in\mathcal{V}} \prod_{j=1}^{k} e^{-\frac{1}{2\sigma^2}[tn_{v,j}\|\mu_j-\overline{X}_{v,j}\|^2+t\sum_{i:v_i=j}\|X_i-\overline{X}_{v,j}\|^2+\beta_j\|\mu_j-\alpha_j\|^2]}$$

$$= \sum_{v\in\mathcal{V}} \underbrace{\left(\prod_{j=1}^{k} \frac{1}{(\beta_j+tn_{v,j})^{d/2}} e^{-\frac{1}{2\sigma^2}[\frac{\beta_j t n_{v,j}}{\beta_j+tn_{v,j}}\|\overline{X}_{v,j}-\alpha_j\|^2+t\sum_{i:v_i=j}\|X_i-\overline{X}_{v,j}\|^2]}\right)}_{w_t(v)}$$

$$\times \underbrace{\left(\prod_{j=1}^{k} (\beta_j+tn_{v,j})^{d/2} e^{-\frac{(\beta_j+tn_{v,j})}{2\sigma^2}\|\mu_j-\widetilde{\mu}_{v,j}^{(t)}\|^2}\right)}_{f_v^{(t)}(\theta)}$$

with $\widetilde{\mu}_{v,j}^{(t)} := \frac{\beta_j}{\beta_j+tn_{v,j}}\alpha_j + \frac{tn_{v,j}}{\beta_j+tn_{v,j}}\overline{X}_{v,j}$. At $t = 1$, this is equal to the target distribution (3), while at $t = 0$ it assigns equal probability to all labelings which makes it easy to initialize.

Our algorithm begins by drawing M uniformly random labelings for the data, which will serve as the chains' initializations. For each chain, we calculate the $t = 0$ version of

$$\frac{\partial}{\partial t} \log w_t(v) = -\sum_{j=1}^{k} \left[\frac{d}{2} \frac{n_{v,j}}{\beta_j + t n_{v,j}} + \frac{1}{2\sigma^2} \frac{\beta_j n_{v,j}}{(\beta_j + t n_{v,j})^2} \|\overline{X}_{v,j} - \alpha_j\|^2 + \frac{1}{2\sigma^2} \sum_{i:v_i=j} \|X_i - \overline{X}_{v,j}\|^2 \right]. \tag{4}$$

We then estimate the expectation of this quantity by averaging these values over the M chains. For each labeling v and time t, let $\hat{\delta}_v(t)$ denote (4) minus the average. Ideally, h is small enough that .5 times the largest absolute value of $\hat{\delta}_v(t)$ is no greater than .5, so that the coin-flips of Theorem 1.1 will be possible. For each chain, generate a coin-flip with heads probability $.5 + h\hat{\delta}_v(t)$ where v is the labeling of the chain in question. If heads, then leave the chain alone. If tails, then the labeling of this chain is replaced by the current labeling of another chain chosen uniformly at random from the $M - 1$ others.

Since $h\hat{\delta}_v(t)$ is only an estimate of an approximation of the required quantity from Theorem 1.1, we will intermittently follow teleportation steps with steps of Gibbs sampling to move the distribution of labels and parameters closer to the time t version of the posterior. Given a labeling v, the Gibbs sampling procedure draws independent Gaussian component means according to the density proportional to $f_v^{(t)}$. Then given component means, the label for the ith observation is assigned to label j with probability

$$\frac{e^{-\frac{1}{2\sigma^2} t \|X_i - \mu_j\|^2}}{\sum_{j'=1}^{k} e^{-\frac{1}{2\sigma^2} t \|X_i - \mu_{j'}\|^2}}.$$

These Gibbs sampling steps also help weaken the dependence among the chains.

This algorithm is used in [Brinda, 2018, Chap 5] to provide initializers to EM for optimizing log likelihood. HOW did it do in the simulations?

5 Discussion

-importance.

-implications for other problems including function estimation.

-more analysis and hopefully statistical guarantees will be in a future paper.

References

W. D. Brinda. *Adaptive Estimation with Gaussian Radial Basis Mixtures*. PhD thesis, Yale University, 2018.

Zhenping Xie, Jun Sun, Vasile Palade, Shitong Wang, and Yuan Liu. Evolutionary sampling: A novel way of machine learning within a probabilistic framework. *Information Sciences*, 299:262 – 282, 2015.