Summary of Monte Carlo Sampling Methods

Tianpei Xie

Sep. 28th., 2022

Contents

1 Basic Concepts

1.1 Random variable generation

- The most fundamental methods for random variable generation is the ***uniform pseudo-random number generator***.

  Definition A *uniform pseudo-random number generator* is an algorithm which, starting from an initial value u_0 and a transformation D, produces a sequence $(u_i) = (D^i(u_0))$ of values in $[0,1]$. For all n, the values $(u_1, \ldots, u_n)$ reproduce the behavior of an *iid* sample $(V_1, \ldots, V_n)$ of uniform random variables when compared through a usual set of tests.

- Given uniform distribution $\mathcal{U}[0,1]$, and a known c.d.f. $F(x)$, we can generate samples via the ***inverse transform*** method. [Robert and Casella, 1999]

  Definition For a ***non-decreasing*** function F on $\mathbb{R}$, the ***generalized inverse*** of F, F^{-}, is the function defined by

  $$F^{-}(u) = \inf\{x \in \mathcal{R}(F) : F(x) \geq u\}. \tag{1}$$

  where $\mathcal{R}(F)$ is the range of F. If F is the cumulative distribution function $F_X(x) = P(X \leq x)$, then F_X^{-1} is called ***quantile function***.

- We then have the following lemma, sometimes known as the **probability integral transform**, which gives us a representation of any random variable as a transform of a uniform random variable.

  Lemma 1.1 *[Robert and Casella, 1999]*
  If $U \sim \mathcal{U}[0,1]$, then the random variable $F^{-}(U)$ has the distribution F.

- Theoretically, any random variable can be generated using the inverse transform. But in practice, it is only available when we can compute the c.d.f and its inverse F^{-} explicitly.

1.2 Why Monte Carlo ?

- The most fundamental operations in probabilistic modeling and statistics involve ***integration*** on **high dimensional spaces**. For example,

  - ***Expectation***:

    $$\mathbb{E}_{\boldsymbol{X}\sim p}\left[h(\boldsymbol{X})\right] = \int_{\mathcal{X}} h(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \tag{2}$$

    where $\boldsymbol{x} := [x_1, \ldots, x_d]$

  - ***Marginalization***:

    $$P(x_1, \ldots, x_{i-1}, x_{i+1} \ldots, x_d) = \int_{x_i \in \mathcal{X}_i} P(x_1, \ldots, x_{i-1}, x_i, x_{i+1} \ldots, x_d) dx_i$$

  - ***Conditioning***:

    $$P(x_1, \ldots, x_{i-1}, x_{i+1} \ldots, x_d \,|\, X_i = x_i) = \frac{P(x_1, \ldots, x_{i-1}, x_i, x_{i+1} \ldots, x_d)}{\int_{\boldsymbol{x}_{-i} \in \boldsymbol{X}_{-i}} P(x_1, \ldots, x_{i-1}, x_i, x_{i+1} \ldots, x_d) d\boldsymbol{x}_{-i}}$$

    where $\boldsymbol{x}_{-i} = [x_1, \ldots, x_{i-1}, x_{i+1} \ldots, x_d]$.

- In ***Bayesian inference*** and analysis, we need to compute *posterior distribution* given data via **the Bayes rule**:

$$\begin{aligned} P(\boldsymbol{\Theta}|\mathcal{D}) &= \frac{P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})}{P(\mathcal{D})} \\ &= \frac{P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})}{\int_{\boldsymbol{\Theta}} P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})d\boldsymbol{\Theta}} \\ \Rightarrow \log P(\boldsymbol{\Theta}|\mathcal{D}) &= \log P(\mathcal{D}|\boldsymbol{\Theta}) + \log P(\boldsymbol{\Theta}) - \log \int_{\boldsymbol{\Theta}} P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})d\boldsymbol{\Theta} \end{aligned}$$

where $\mathcal{D} := \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ are i.i.d. samples and $\boldsymbol{\Theta}$ is a set of parameters for model $p(\mathcal{D}|\boldsymbol{\Theta})$.

The main challenge for Bayesian methods in high dimensional space is to compute the normalization factor and conditional expectation

$$\begin{aligned} Z &= \int_{\boldsymbol{\Theta}} P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})d\boldsymbol{\Theta}. \\ \mathbb{E}\left[\boldsymbol{\Theta}|\mathcal{D}\right] &= \int \boldsymbol{\Theta} P(\boldsymbol{\Theta}|\mathcal{D})d\boldsymbol{\Theta} \end{aligned}$$

- The ***log-partition function*** in **exponential families** play a critical role since it defines a *bijective mapping* from *natural parameters* to *mean parameters* and is related to **negative entropy** via *variational principles.* The log-partition function for exponential families is a convex function as well.

$$A(\boldsymbol{\eta}) = \log \int_{\Omega} \exp\left(\langle \boldsymbol{\eta}\,,\, \boldsymbol{\phi}(\boldsymbol{x})\rangle\right) d\boldsymbol{x}$$

- The ***<u>Monte Carlo methods</u>*** ***approximate integration*** by the ***empirical average*** over a set of samples $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m)$ generated from the density p. Specifically from (2),

$$\overline{h}_m = \frac{1}{m}\sum_{i=1}^{m} h(\boldsymbol{X}_i). \tag{3}$$

since $\overline{h}_m$ converges almost surely to $\mathbb{E}_{\boldsymbol{X}\sim p}\left[h(\boldsymbol{X})\right]$ by the *Strong Law of Large Numbers.* Moreover, when $\overline{h}_m$ has a finite expectation under p, the speed of convergence of $\overline{h}_m$ can be assessed since the variance

$$\mathrm{Var}\left(\overline{h}_m\right) = \frac{1}{m}\int_{\mathcal{X}} \left(h(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{X}\sim p}\left[h(\boldsymbol{X})\right]\right)^2 p(\boldsymbol{x})d\boldsymbol{x}$$

can also be estimated from the sample $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m)$ through

$$v_m = \frac{1}{m}\sum_{i=1}^{m}\left(h(\boldsymbol{X}_i) - \overline{h}_m\right)^2$$

For m large,

$$\frac{h(\boldsymbol{X}) - \overline{h}_m}{\sqrt{v_m}} \to \mathcal{N}(0,1).$$

This leads to the construction of a ***convergence test*** and of ***confidence bounds*** on the approximation of $\mathbb{E}_{\boldsymbol{X}\sim p}\left[h(\boldsymbol{X})\right]$.

1.3 Advantages and disadvantages of Monte Carlo Methods

The ***advantages*** of Monte Carlo Methods are

- ***Easy to implement***: Since the Monte Carlo methods rely on empirical averages to approximate the high dimensional integration, it is easier to implement compared to solving the integration analytically.
- ***Not need for exact knowledge***: As an approximation method, it is not required for Monte Carlo sampler to have exact knowledge on the underlying target distributions. Many algorithms allow the agent to produce samples from target distribution given only the envelop of the distributions, the local factors/conditional distributions, or the dynamic of system.
- ***Suitable for parallel and distributed computing***: Since Monte Carlo methods only require averaging, it is suitable for distributed computing algorithms such as MapReduce and Spark as well as parallel computing based on matrix operations.
- ***Widely available for inference and learning***: When computing expectations, sampling methods are always available due to *the Strong Law of Large Numbers.* This is especially true when the target joint distributions are complicated, such as probabilistic graphical models, mixture models, truncated models, non-parametric Bayesian models etc. Many efficient Monte Carlo methods are proposed based on additional knowledge on the target distribution such as lower/upper bounds, factorization, gradients as well as dynamics of system.

The ***disadvantages*** include:

- ***High variance***: Compared to deterministic methods such as variational inference, dynamic programming etc., many Monte Carlo methods have low bias but high variance due to the randomness introduced by sampling. Poor choice of sampling algorithm with high rejection and lacks of knowledge on target distribution would lead to high variance as well. It is thus critical to ***explore as much information as possible*** regarding target distribution when designing the sampling method.
- ***Slow convergence***: Monte Carlo methods would suffer slow convergence if not designed properly. In MCMC, the rate of convergence depends on the characteristic of the Markov chain as well as sample efficiency. In general, the number of samples required to attain a given accuracy level increased when the dimensionality and model complexity increased. In reinforcement learning, the agents need to trial-and-error many episodes in order to learn how to react which is prohibitive expensive in real world applications.
- ***Low sample efficiency***: Monte Carlo methods rely on **trial-and-reject mechanism** to produce samples, which would waste a lot samples in the process. Low sample efficiency means that the agents spend too much time generating irrelevant samples without updating its knowledge. The sample efficiency is critical for small-scale, high cost or online applications since the total amount of samples is limited and it is very costly to drop samples.
- ***Slow to update***: In online setting, Monte Carlo methods may need to wait until the end of episode to update the agent. In Hamiltonian Monte Carlo, each sample is generated after a period of time of dynamic evolution.

2 Rejection and Weighting

2.1 Reject Sampling

- Suppose $l(\boldsymbol{x}) = c\, f(\boldsymbol{x})$ is computable, where f is the probability density and c is unknown. If we can find a sampling distribution $g(\boldsymbol{x})$ and a ***covering constant*** M so that the envelop property (i.e. $Mg(\boldsymbol{x}) \geq l(\boldsymbol{x})$) is satisfied for all $\boldsymbol{x}$. We can apply the following procedure:

  Accept-Reject sampling sampling [Liu, 2001]

  1. Draw sample $\boldsymbol{X}$ from $g(\cdot)$ and compute the **ratio**
  $$r = \frac{l(\boldsymbol{x})}{M\, g(\boldsymbol{x})} \quad (\leq 1)$$
  2. Draw $U \sim \mathcal{U}[0,1]$. If $U \leq r$, accept $\boldsymbol{X}$; otherwise, reject $\boldsymbol{X}$.

  Then the accepted samples follow the distribution $f(\boldsymbol{x})$.

- **Lemma 2.1** *If there exist a density $g_m(\boldsymbol{x})$, a function $g_l(\boldsymbol{x})$ and a constant M such that*
$$g_l(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq M\, g_m(\boldsymbol{x})$$
  then the algorithm **Envelope Accept-Reject**

  1. *Draw sample $\boldsymbol{X}$ from $g_m(\boldsymbol{x})$ and U from $\mathcal{U}[0,1]$*
  2. *Accept $\boldsymbol{X}$ if $U \leq g_l(\boldsymbol{x})/(M\, g_m(\boldsymbol{x}))$;*
  3. *Otherwise, accept $\boldsymbol{X}$ if $U \leq f(\boldsymbol{x})/(M\, g_m(\boldsymbol{x}))$ else return to step 1.*

  produces random variables that are distributed according to f.

2.2 Variance Reduction

We introduce several techniques that reduce variance while maintain the unbiasness of the estimator.

- **Stratified Sampling**: We partition the region $\mathcal{X}$ into k sub-regions $\mathcal{X}^i$ and suppose that the probability distribution f in each sub-regions is ***relative homogeneous***. Then we can generate samples from each region $\mathcal{X}^i$ and compute the region-wise sample average. The final result is the average of region-wise sample average and the variance is reduced from original.

  Note that if the probability distribution is not homogeneous within each region, the stratified sampling would increase the bias and makes the estimate less accurate.

- **Rao-Blackwellization**: An approach to reduce the variance of an estimator is to use the *conditioning inequality*
$$\text{var}(\mathbb{E}\left[\delta(\boldsymbol{X})|\boldsymbol{Z}\right]) \leq \text{var}(\mathbb{E}\left[\delta(\boldsymbol{X})\right]) \tag{4}$$
  sometimes called ***Rao-Blackwellization*** [Liu, 2001]. This method reflects a **basic principle**: ***one should carry out analytical computation as much as possible***. The more information available for sampler, the less variance it would have.

Suppose the sample can be decomposed into two parts $(\boldsymbol{X}, \boldsymbol{Z})$ and the conditional expectation can be carried out explicitly $\mathbb{E}\left[h(\boldsymbol{X})|\boldsymbol{Z}\right]$ for each sub-population $(\boldsymbol{X}, \boldsymbol{Z}_i)$, then the estimator

$$\hat{I}_m = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\left[h(\boldsymbol{X})|\boldsymbol{Z}_i\right]$$

has lower variance than the direct sample average of $h(\boldsymbol{X}_j)$. In statistic, $\hat{I}_m$ is often called ***mixture estimator*** since it combine a mixture of distributions for each sub-population.

- **Control Variates Methods**: In this method, one uses a **control variate** C that is highly correlated with sample $\boldsymbol{X}$ to reduce the variance. Suppose that $\mu_C = \mathbb{E}\left[C\right]$ is known and the sample estimator

  $$X(b) = X + b\left(C - \mu_C\right),$$

  which has the same mean as X. If we choose optimal $b^* = \text{Cov}(X\,C)/\text{var}(C)$, the variance of $X(b)$ is smaller than X.

  Note that the technique of control variates is manageable only in very specific cases: the control function $\mu_C = \mathbb{E}\left[C\right]$ must be available, as well as the optimal weight b^*.

- **Antithetic Variates Methods**: This method describes a way of creating ***negative correlated*** samples so that the average of two negative correlated samples have lower variance than the variance of two independent samples. More generally, let g be a monotone function on $[0, 1]$, so that for any $u_1, u_2 \in [0, 1]$

  $$\left(g(u_1) - g(u_2)\right)\left(g(1-u_1) - g(1-u_2)\right) \leq 0.$$

  Then for two i.i.d random variables $U_1, U_2 \sim \mathcal{U}[0, 1]$, $X_1 = g(U)$ and $X_2 = g(1-U)$ we have

  $$\mathbb{E}\left[\left(g(U_1) - g(U_2)\right)\left(g(1-U_1) - g(1-U_2)\right)\right] = \text{Cov}(X_1\,X_2) \leq 0$$

  Thus the variance of $\frac{1}{2}(X_1 + X_2)$ is less than the variance of two independent samples from Monte Carlo simulator. Here X_2 is called **antithetic variables**.

2.3 Importance Sampling

- The vanilla rejection sampling method suffers from ***low sample efficiency and slow convergence*** since it wasted a lot effort evaluating random samples located in regions where the target function value $h(\boldsymbol{x})p(\boldsymbol{x})$ is almost zero.

- The ***importance sampling*** idea suggests that one should **focus on regions of "importance"** so as to save computational resources. This is esp. important for ***high dimensional probability***, since in high dimensional space, the ***support*** *of target distribution is exponentially small* as compared to the entire region $\mathcal{X}$. In high dimensional setting, the **vanilla Monte Carlo schemes are bound to fail**.

- Consider the expectation estimation

  $$\begin{aligned}\mu = \mathbb{E}_{\boldsymbol{X}\sim p}\left[h(\boldsymbol{X})\right] &= \int_{\mathcal{X}} h(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \\ &= \mathbb{E}_{\boldsymbol{X}\sim g}\left[\frac{p(\boldsymbol{X})}{g(\boldsymbol{X})}h(\boldsymbol{X})\right] := \mathbb{E}_{\boldsymbol{X}\sim g}\left[w(\boldsymbol{X})h(\boldsymbol{X})\right] \end{aligned} \tag{5}$$

$$= \int_{\mathcal{X}} w(\boldsymbol{x}) h(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x}$$

where $w(\boldsymbol{x}) := \dfrac{p(\boldsymbol{x})}{g(\boldsymbol{x})}$ is the ***importance weight.***

- The ***Importance Sampling Algorithm*** [Liu, 2001, Robert and Casella, 1999] is as below:
  1. Draw $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m \sim g(\boldsymbol{x})$ where g is trial distribution;
  2. Calculate *the importance weight* $w(\boldsymbol{x})$:
  $$w_i := w(\boldsymbol{X}_i) = \frac{p(\boldsymbol{X}_i)}{g(\boldsymbol{X}_i)}, \quad i = 1, \ldots, m$$
  3. Approximate μ by a ***biased*** *estimator* (***asymptotic*** *unbiased*)
  $$\hat{\mu} = \frac{\sum_{i=1}^{m} w_i\, h(\boldsymbol{X}_i)}{\sum_{i=1}^{m} w_i} \tag{6}$$
  Another way to approximate μ is via the ***unbiased*** *estimator*
  $$\hat{\mu} = \frac{1}{m} \sum_{i=1}^{m} w_i\, h(\boldsymbol{X}_i) \tag{7}$$
- The ***trial*** *distribution* g is also called the ***proposal*** *distribution* or ***instrumental*** *distribution* [Robert and Casella, 1999, Liu, 2001].
- The **variance** of importance sampler is
$$\mathbb{E}_g \left[\frac{p^2(\boldsymbol{X})}{g^2(\boldsymbol{X})} h^2(\boldsymbol{X}) \right] = \mathbb{E}_p \left[\frac{p(\boldsymbol{X})}{g(\boldsymbol{X})} h^2(\boldsymbol{X}) \right] = \int_{\mathcal{X}} \frac{p^2(\boldsymbol{x})}{g(\boldsymbol{x})} h^2(\boldsymbol{x}) d\boldsymbol{x}.$$
In practice, the importance sampler would work poorly (high-amplitude ***jumps***, ***instability*** of the path of the average, ***slow convergence***) if
$$\int_{\mathcal{X}} \frac{p^2(\boldsymbol{x})}{g(\boldsymbol{x})} d\boldsymbol{x} = \infty.$$
On the other hand, the convergence is guaranteed almost surely without condition on g.
- The ***relative efficiency*** between vanilla Monte Carlo estimator vs. the importance sampler (6) is approximated as
$$\text{RE} \approx \frac{1}{1 + \text{var}_g(w)} \tag{8}$$
The ***effective sample size (ESS)*** is thus
$$\text{ESS} \approx \frac{m}{1 + \text{var}_g(w)}.$$
- There are several **advantages** using importance sampling:

- For (6), one only need to know target distribution $p(\cdot)$ ***up to a constant*** when calculating the importance weight. This is very convenient for distributions such as the exponential families. Also (6) has lower variance with higher bias as compared to (7).
- Importance sampling allows us to **approximate** the expectation of an ***unknown*** or ***complex*** target distribution using **simple trial distribution** and then ***correcting the bias*** via ***reweighting***. Similar to rejection sampling, the performance of importance sampling depends on the **closeness** of $g(\boldsymbol{x})$ to the target $|h(\boldsymbol{x})|\,p(\boldsymbol{x})$. In particular, the trial density $g(\boldsymbol{x})$ should have longer tail than $p(\boldsymbol{x})$, i.e. $\text{supp}(g) \supset \text{supp}(p)$. In high dimensional setting, it would still be challenging to find such a good trial distribution g.

2.4 Sequential Importance Sampling (SIS)

- It is nontrivial to find a good trial distribution g in high dimensional space. One of the most useful strategies in these problems is to *build up the trial density* ***sequentially***.
- Denote $\boldsymbol{x} := [x_1, \ldots, x_d]$ and $\boldsymbol{x}_t = (x_1, \ldots, x_t)$. Then both the trial density g and target density p have the factorization

$$g(\boldsymbol{x}) = g(x_1) \prod_{t=2}^{d} g(x_t|\boldsymbol{x}_{t-1})$$
$$p(\boldsymbol{x}) = p(x_1) \prod_{t=2}^{d} p(x_t|\boldsymbol{x}_{t-1}).$$

So the importance weight has a **recursive update**

$$w_t(\boldsymbol{x}_t) = w_{t-1}(\boldsymbol{x}_{t-1}) \frac{p(x_t|\boldsymbol{x}_{t-1})}{g(x_t|\boldsymbol{x}_{t-1})}, \quad t = 2, \ldots, d \tag{9}$$
$$w_1(\boldsymbol{x}_1) = \frac{p(x_1)}{g(x_1)}$$

- In order to approximate the marginal target $p(\boldsymbol{x}_{t-1})$, we introduce a set of "auxiliary distributions" $p_t(\boldsymbol{x}_t), t = 1, \ldots, d$. In particle filtering, $p_t(\boldsymbol{x}_t)$ is chosen as the correct posterior of the true signals.
- The ***Sequential Importance Sampling (SIS)*** algorithm is described as below:
  1. Draw $X_t = x_t$ from $g(x_t|\boldsymbol{x}_{t-1})$ and let $\boldsymbol{x}_t \leftarrow [x_t, \boldsymbol{x}_{t-1}]$.
  2. Compute the ***incremental weight***:
  $$u_t = \frac{p_t(\boldsymbol{x}_t)}{p_{t-1}(\boldsymbol{x}_{t-1})\, g(x_t|\boldsymbol{x}_{t-1})} \tag{10}$$
  3. Compute $w_t \leftarrow w_{t-1}\, u_t$

  It is easy to show that $\boldsymbol{x}_t$ is properly weighted by w_t given that $\boldsymbol{x}_{t-1}$ is properly weighted by w_{t-1}. Thus the whole sample $\boldsymbol{x}$ obtained sequentially is properly weighted by the final importance w_d w.r.t. to target $p(\boldsymbol{x})$.
- The **benefits** for using sequential importance sampling include:

  - We can make use of the **characteristics of local factor** $p(x_t|\boldsymbol{x}_{t-1})$ in target density when designing trial density $g(x_t|\boldsymbol{x}_{t-1})$. An example is the *local Markov property* $p(x_t|\boldsymbol{x}_{t-1}) = p(x_t|x_{t-1})$ for probabilistic graphical models. By sequential sampling, we break the complex problem into smaller pieces.
  - We can **stop** generating further components of $\boldsymbol{x}$ if the ***partial weights*** w_k that derived from the sequentially generated the ***partial samples*** $\boldsymbol{x}_k$ are *too small*. We can also **reject sample with small weight** and restart again. This way we avoid wasting effort generating samples with little effect on final estimation. This rejection process would introduce additional bias which should be corrected [Robert and Casella, 1999].
  - The SIS algorithm is **attractive** since we can use a sequence of auxiliary distributions to construct more efficient sampling algorithm.

- Consider the state-space model

$$\begin{aligned} \boldsymbol{\xi}_t &\sim q_t(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1}, \theta) \quad \text{(state equation)} \\ \boldsymbol{y}_t &\sim f_t(\boldsymbol{y}_t|\boldsymbol{\xi}_t, \phi) \quad \text{(observation equation)}, \end{aligned}$$

  where $\boldsymbol{\xi}_k \in \mathbb{R}^d$ is the state at time k and $\boldsymbol{y}_k$ is the observation at time k. We can apply the sequential importance sampling to approximate the online conditional mean estimator $\widehat{\boldsymbol{\xi}}_t = \mathbb{E}\left[\boldsymbol{\xi}_t \,|\, \boldsymbol{y}_{1:t}\right]$. However, directly implementation of SIS will result in "*particle degeneracy*", i.e. some particle will be discarded during the sequential updates.

- The ***particle filter*** (or ***boostrap filter***) is proposed to fix this problem. It is described as:

  The ***Sampling-Importance-Resampling (SIR)***

  1. Draw $\boldsymbol{\xi}_{t+1}^{(*,j)}$ from the state equation $q_t(\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_t^{(j)}, \theta)$ for $j = 1, \ldots, m$
  2. Weight each draw by $w_{t+1}^{(j)} = f_t(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1}^{(*,j)}, \phi)\, \widetilde{w}_t^{(j)}$
  3. Normalize $w_{t+1}^{(j)}$ as $\widetilde{w}_{t+1}^{(j)}$
  4. ***Resample*** from $\left\{\boldsymbol{\xi}_{t+1}^{(*,1)}, \ldots, \boldsymbol{\xi}_{t+1}^{(*,m)}\right\}$ according to multinomial distribution with probability $\left\{\widetilde{w}_{t+1}^{(j)}\right\}_{j=1}^{m}$ to produce a random sample $\left\{\boldsymbol{\xi}_{t+1}^{(1)}, \ldots, \boldsymbol{\xi}_{t+1}^{(m)}\right\}$ for time $t+1$

  Averaging $\left\{\boldsymbol{\xi}_{t+1}^{(1)}, \ldots, \boldsymbol{\xi}_{t+1}^{(m)}\right\}$ will obtain the approximate conditional posterior mean estimator at $t+1$.

3 Markov Chain Monte Carlo

3.1 From Vanilla Monte Carlo to MCMC

- **Definition** A ***Markov Chain Monte Carlo (MCMC) method*** for the simulation of a distribution f is any method producing an ergodic Markov chain $(X_t)_t$ whose stationary distribution is f.
- Compared to vanilla Monte Carlo (e.g. inverse transformation, reject sampling, importance sampling), MCMC has the following **characteristics**:

  - Unlike vanilla Monte Carlo, ***Markov Chain Monte Carlo (MCMC) methods*** generate ***dependent* samples** via Markov chain.
  - The MCMC updates ***preserve the probability measure*** *π at convergence*. That is, *when the Markov chain* ***converges***, the distribution of X_t is ***invariant***, i.e. it is the same as the distribution of $X_{t+1}, \ldots$. Thus we have obtained a sequence of ***identically distributed (but dependent) samples***. When Markov chain converges (***mixing***), we can use samples the same way as we did in vanilla Monte Carlo to approximate the expectation. In particular, an MCMC estimator is

$$J_T = \widehat{\mathbb{E}}_{\boldsymbol{\pi}}\left[h(X)\right] = \frac{1}{T}\sum_{t=0}^{T} h(X_t). \tag{11}$$

    The ergodic theorem guarantees the (almost sure) convergence of the empirical average to $\mathbb{E}_{\boldsymbol{\pi}}\left[h(X)\right]$ where $\boldsymbol{\pi}$ is the stationary distribution. A sequence $(X_t)_t$ produced by a Markov chain Monte Carlo algorithm can thus be employed just as an iid sample.
  - Similar to importance sampling, we approximate the expectation (2) using an alternative proposal distribution $\boldsymbol{\pi}$ which is the stationary distribution of an ergodic Markov chain. This is the idea behind Metropolis-Hastings algorithm.

- Markov Chain Monte Carlo (MCMC) methods are ***preferred*** in following situations:
  - The target distribution is ***high dimensional***. Due to *the curse of dimensionality*, the variance, which is a function of dimension d, will grow exponentially as the dimensionality increases. Moreover, many high dimensional joint distributions are usually not represented in explicit function form due to their complicated partition functions. In this situation, finding a proposal distribution that is close to the target distribution in high dimensional space is also very challenging.
  - Some stochastic optimization algorithms naturally produce Markov chain structures. It is a general fact that the use of Markov chains allows for a greater scope than the methods presented in vanilla Monte Carlo.
  - Vanilla Monte Carlo and MCMC algorithms both satisfy the $O(1/\sqrt{n})$ convergence requirement for the approximation of J. There are thus many instances where a specific MCMC algorithm dominates, variance-wise, the corresponding Monte Carlo proposal.

3.2 Metropolis-Hastings Algorithm

- Consider the following energy-based model

$$\pi(\boldsymbol{x}) = \frac{1}{Z}\exp(-h(\boldsymbol{x})),$$
$$\text{where } Z = \int_{\mathcal{X}} \exp(-h(\boldsymbol{x}))d\boldsymbol{x}$$

  is the partition function in high dimensional space.

- The basic idea for ***Metropolis Algorithm*** is to simulate f using the stationary distribution g from a Markov chain. Compare to analysis of Markov chain itself, which often starts from a *known* transition kernel, the Metropolis algorithm starts from a *known* stationary distribution g and is interested in how to prescribe an efficient transition kernel to reach the equilibrium.

- Intuitively, the Metropolis Algorithm is based on a *"trial-and-error" strategy*: at each iteration, a random perturbation of $\boldsymbol{X}_t$ is generated by Markov chain. Then the gain is computed for this new sample. If the gain is large enough, it will be accepted by high probability. Otherwise, we keep using the old sample $\boldsymbol{X}_t$.

- The ***Metropolis-Hastings Algorithm*** is described as below:

  1. Given current configuration $\boldsymbol{X}_t$, draw $\boldsymbol{Y}$ from the proposal function $K(\boldsymbol{X}_t, \boldsymbol{Y})$.
  2. Compute the ***Hastings ratio*** (*acceptance function*):
  $$r(\boldsymbol{X}_t, \boldsymbol{Y}) := \frac{\pi(\boldsymbol{Y})\, K(\boldsymbol{Y}, \boldsymbol{X}_t)}{\pi(\boldsymbol{X}_t)\, K(\boldsymbol{X}_t, \boldsymbol{Y})} \tag{12}$$
  3. (***Metropolis Rejection***) Accept $\boldsymbol{X}_{t+1} = \boldsymbol{Y}$ with probability
  $$\alpha(\boldsymbol{X}_t, \boldsymbol{Y}) = \min\{1,\ r(\boldsymbol{X}_t, \boldsymbol{Y})\}$$
  4. Otherwise, accept $\boldsymbol{X}_{t+1} = \boldsymbol{X}_t$.

  Here $K(x, y)$ is called ***proposal function***, which is a *transition probability* $q(y|x)$ for Markov chain. The target distribution π is the *stationary distribution* for this Markov chain.

- We can check on the properties of the Markov chain $(\boldsymbol{X}_t)_t$ from Metropolis-Hastings algorithm:

  - (***Irreducibility***): To make sure the *irreducibility* to hold, the domain $\mathcal{X}$ of target distribution π need to be ***connected***. Otherwise, the Markov chain will not be able to reach other connected components. In other words, the stationary distribution π is ***not multimodal distribution***.
  - (***Positive recurrent***): A sufficient condition for both the *irreducibility* and *positive recurrence* to hold is the positivity of transition kernel:
  $$q(\boldsymbol{y}|\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{y}) > 0 \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}. \tag{13}$$
  This makes sure that every state can be visited in one step. Thus the Markov chain is positive recurrent. In fact, it can be shown in [Robert and Casella, 1999] that if $(\boldsymbol{X}_t)_t$ is π-*irreducible*, then it is *Harris recurrent.*

    Lemma 3.1 *[Robert and Casella, 1999]*
    If $(\boldsymbol{X}_t)_t$ from MH algorithm is π-irreducible, then it is Harris recurrent.
  - (***Aperiodic***): Unlike the Rejection Sampling, the Metropolis Rejection does not simply regenerate a new proposal repeatedly until some of them is accepted. Instead, it reuse the value from the old state. An attempt to make Metropolis Rejection like the Rejection sampling will destroy the property that **this update will preserve the distribution** π

    A *sufficient condition* for ***aperiodic*** property to hold is to have $P\{\boldsymbol{X}_{t+1} = \boldsymbol{X}_t\} > 0$, and thus
  $$P\{\pi(\boldsymbol{X}_t)\, K(\boldsymbol{X}_t, \boldsymbol{Y}) \le \pi(\boldsymbol{Y})\, K(\boldsymbol{Y}, \boldsymbol{X}_t)\} < 1 \tag{14}$$

This is **critical** to guarantee the aperiodic property of the Markov chain, which is essential for *Ergodic theorem* to hold. Note that this condition also make sure K is *not the transition kernel* for the Markov chain induced by Metropolis-Hastings algorithm.

  – (***Reversibility***): From the theorem, we see that $(\boldsymbol{X}_t)_t$ is ***time-reversible*** with π as its ***invariant distribution***.

- The *transition probability* of $(\boldsymbol{X}_t)_t$ from the Metropolis-Hastings algorithm is computed as

$$\begin{aligned} A(\boldsymbol{x}\,,\boldsymbol{y}) &= K(\boldsymbol{x}\,,\boldsymbol{y})\,\alpha(\boldsymbol{x}\,,\boldsymbol{y}) \\ &= K(\boldsymbol{x}\,,\boldsymbol{y})\ \min\left\{1,\ \frac{\pi(\boldsymbol{y})\,K(\boldsymbol{y},\boldsymbol{x})}{\pi(\boldsymbol{x})\,K(\boldsymbol{x}\,,\boldsymbol{y})}\right\} \\ &= \frac{\min\left\{\pi(\boldsymbol{x})\,K(\boldsymbol{x}\,,\boldsymbol{y})\,,\pi(\boldsymbol{y})\,K(\boldsymbol{y},\boldsymbol{x})\right\}}{\pi(\boldsymbol{x})} := \frac{\delta(\boldsymbol{x},\boldsymbol{y})}{\pi(\boldsymbol{x})}. \end{aligned} \tag{15}$$

Note that due to the **rejection rule**, $A(\boldsymbol{x}\,,\boldsymbol{y}) \neq K(\boldsymbol{x}\,,\boldsymbol{y})$. We can see that since $\delta(\boldsymbol{x},\boldsymbol{y}) = \delta(\boldsymbol{y},\boldsymbol{x})$, the *detailed balance equation* holds

$$\pi(\boldsymbol{x})A(\boldsymbol{x},\boldsymbol{y}) = \pi(\boldsymbol{y})A(\boldsymbol{y},\boldsymbol{x}). \tag{16}$$

From the theorem on detailed balance equation, we see that π is guaranteed to be the stationary distribution of the Markov chain and the Markov chain $(\boldsymbol{X}_t)_t$ is *time-reversible.*

- Finally, we see that the convergence of estimator (11) based on ergodic theorem.

Theorem 3.2 *[Robert and Casella, 1999]*
Suppose that the Metropolis-Hastings Markov chain $(\boldsymbol{X}_t)_t$ is π-irreducible.

  1. *If $h \in L_1(\pi)$, then*

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} h(\boldsymbol{X}_t) = \mathbb{E}_{\boldsymbol{\pi}}\left[h(\boldsymbol{X})\right] \tag{17}$$

  2. *If, in addition, $(\boldsymbol{X}_t)_t$ is **aperiodic**, then*

$$\lim_{t\to\infty} \left\| \int_{\mathcal{X}} K^t(\boldsymbol{x},\cdot)d\mu(\boldsymbol{x}) - \pi \right\|_{TV} = 0 \tag{18}$$

  *for **every initial distribution** μ, where $K^t(\boldsymbol{x},\cdot)$ denotes the kernel for t transitions.*

- **Corollary 3.3** *[Robert and Casella, 1999]*
The conclusions of Theorem 3.2 hold if the Metropolis-Hastings Markov chain $(\boldsymbol{X}_t)_t$ has conditional density $q(\boldsymbol{y}|\boldsymbol{x}) = K(\boldsymbol{x},\boldsymbol{y})$ that satisfies (13) *and* (14).

3.3 Some Special Metropolis-Hastings Algorithms

3.3.1 Random-walk Metropolis-Hastings

- A natural approach for the practical construction of a Metropolis-Hastings algorithm is to take into account the value previously simulated to generate the following value; that is, to consider a ***local exploration*** of the ***neighborhood*** of the current value of the Markov chain. This idea is already used in algorithms such as the *simulated annealing algorithm* and the *stochastic gradient method.*

- The *Random-walk Metropolis-Hastings algorithm* is very commonly used when ***no prior knowledge*** is available. In this case, the preferred proposal is just a *uniformly local* move, since it is ***uninformative*** and ***unbiased***.

- In general, define a ***symmetric distribution*** g_σ so that $q(\boldsymbol{y}|\boldsymbol{x}) = g_\sigma(\|\boldsymbol{y}-\boldsymbol{x}\|_2) > 0$ for all $\|\boldsymbol{y}-\boldsymbol{x}\|_2 < \delta$. Here σ controls the "***range***" of the exploration. The random walk with transition kernel as g will produce an ergodic Markov chain.

  The most common distributions g in this setup are the ***uniform distributions on spheres*** centered at the origin or standard distributions like the ***normal*** and the ***Student's t distributions***. *All these distributions usually need to be* ***scaled***.

- The ***Random-walk Metropolis-Hastings*** is described as below:

  1. Draw $\boldsymbol{\epsilon}_t$ from $g_\sigma(\boldsymbol{\epsilon})$ and set $\boldsymbol{Y} := \boldsymbol{X}_t + \boldsymbol{\epsilon}_t$;
  2. Compute the ratio:
  $$r(\boldsymbol{X}_t, \boldsymbol{Y}) := \frac{\pi(\boldsymbol{Y})}{\pi(\boldsymbol{X}_t)} \tag{19}$$
  3. Accept $\boldsymbol{X}_{t+1} = \boldsymbol{Y}$ with probability
  $$\alpha(\boldsymbol{X}_t, \boldsymbol{Y}) = \min\{1,\ r(\boldsymbol{X}_t, \boldsymbol{Y})\}$$
  4. Otherwise, accept $\boldsymbol{X}_{t+1} = \boldsymbol{X}_t$.

- Random-walk Metropolis is not only simple to implement, it also has a particularly nice **intuition**. The **proposol distribution** is biased towards ***large volumes*** ($g_\sigma(\cdot)$), and hence the *tails* of the target distribution, while the *Metropolis correction* **rejects** those proposals that jump into neighborhoods where ***the density is too small*** ($\pi(\boldsymbol{x}^{(t)})$). The combined procedure then preferentially selects out those proposals that fall into neighborhoods of high probability mass, concentrating towards the typical set as desired.

3.3.2 Independent Metropolis-Hastings (IMH) Algorithm

- ***Independent Metropolis-Hastings (IMH) Algorithm***:

  1. Generate $\boldsymbol{Y}_t$ from $g(\boldsymbol{y})$;
  2. Compute ratio:
  $$r(\boldsymbol{X}_t, \boldsymbol{Y}) := \frac{\pi(\boldsymbol{Y}_t)\, g(\boldsymbol{X}_t)}{\pi(\boldsymbol{X}_t)\, g(\boldsymbol{Y}_t)} = \frac{w(\boldsymbol{Y}_t)}{w(\boldsymbol{X}_t)} \tag{20}$$
  where $w(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{g(\boldsymbol{x})}$ is the *importance weight*.
  3. Accept $\boldsymbol{X}_{t+1} = \boldsymbol{Y}_t$ with probability
  $$\alpha(\boldsymbol{X}_t, \boldsymbol{Y}) = \min\{1,\ r(\boldsymbol{X}_t, \boldsymbol{Y})\}$$
  4. Otherwise, accept $\boldsymbol{X}_{t+1} = \boldsymbol{X}_t$.

- *Independent Metropolis-Hastings (IMH)* choose the proposal transition function $K(\boldsymbol{x}, \boldsymbol{y})$ as an ***independent*** *density* $g(\boldsymbol{y})$. That is, the proposal move $\boldsymbol{Y}_t$ is generated ***independently*** from previous sample $\boldsymbol{X}_t$.

- This method is an **alternative** to the *Rejection Sampling* and *Importance Sampling.* As with Rejection sampling, the performance of *IMH* depends on how close the proposal distribution g to the target distribution π. It is also suggested that g has longer tail than π for robustness of performance.

  It can be shown that if the envelop condition $\pi(\boldsymbol{x}) \le M\, g(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ holds, then the Metropolis-Hastings Markov chain $(\boldsymbol{X}_t)_t$ is ergodic, thus the convergence to $\mathbb{E}_\pi[h]$ is guaranteed.

- Compared to *Rejection Sampling* and *Importance Sampling*, the *Independent Metropolis-Hastings (IMH)* has ***larger expected acceptance probability*** (at least $1/M$ when the chain is stationary) [Robert and Casella, 1999]. Thus IMH is more **sample efficient**.

3.3.3 Configurational bias Monte Carlo

- The *Configurational bias Monte Carlo (CBMC)* can be seen as a ***SIS-based IMH***. Assume that we have auxiliary distributions $\pi_s(\boldsymbol{x}_{1:s})$ where $\boldsymbol{x}_{1:s} = [x_1, \ldots, x_s]$ at each time s. $\boldsymbol{x} := \boldsymbol{x}_{1:d}$. The proposal sampling probability $g_s(x_s|\boldsymbol{x}_{1:(s-1)})$ for all s. Here the proposal joint distribution can be computed sequentially

$$g(\boldsymbol{x}) = g_1(x_1) \prod_{s=2}^{d} g_s(x_s|\boldsymbol{x}_{1:(s-1)}) \tag{21}$$

- The ***Configurational bias Monte Carlo (CBMC)*** is described as below:

  1. Generate $\boldsymbol{Y}$ according to $g(\boldsymbol{y})$ via the sequential process in (21).
  2. Compute the *importance weight*:

$$w(\boldsymbol{Y}) = \frac{\pi(\boldsymbol{Y})}{g(\boldsymbol{Y})} = \frac{\pi_1(Y_1)}{g_1(Y_1)} \prod_{s=2}^{d} \frac{\pi_s(\boldsymbol{Y}_{1:s})}{\pi_s\left(\boldsymbol{Y}_{1:(s-1)}\right) g_s\left(Y_s \mid \boldsymbol{Y}_{1:(s-1)}\right)} \tag{22}$$

     Similarly compute the importance weight for $\boldsymbol{X}_t$ as $w(\boldsymbol{X}_t)$.
  3. Accept $\boldsymbol{X}_{t+1} = \boldsymbol{Y}$ with probability

$$\alpha(\boldsymbol{X}_t, \boldsymbol{Y}) = \min\left\{1,\ \frac{w(\boldsymbol{Y})}{w(\boldsymbol{X}_t)}\right\}$$

  4. Otherwise, accept $\boldsymbol{X}_{t+1} = \boldsymbol{X}_t$.

- We can modify this process to make **stage-wise acceptance decision**. Suppose that, at time $t-1$, $\boldsymbol{X}_t = (X_1^{(t)}, \ldots, X_d^{(t)})$ is accepted. Then at s-th stage, we accept $\boldsymbol{Y}_{1:s}$ with probability

$$\alpha_s(\boldsymbol{X}_{1:s}^{(t)}, \boldsymbol{Y}_{1:s}) = \min\left\{1,\ \frac{u_s(\boldsymbol{Y}_{1:s})}{u_s(\boldsymbol{X}_{1:s}^{(t)})}\right\} \tag{23}$$

where

$$u_s(\boldsymbol{Y}_{1:s}) = \frac{\pi_s\left(\boldsymbol{Y}_{1:s}\right)}{\pi_s\left(\boldsymbol{Y}_{1:(s-1)}\right)\, g_s\left(Y_s \,|\, \boldsymbol{Y}_{1:(s-1)}\right)}.$$

That is, the acceptance rate at each stage is equal to the *ratio of incremental weights* between proposed state and the old state.

If ***rejected***, we *go back to the* ***first stage*** *to rebuild the whole configuration*.

Note that it is **not necessary** to make accept-reject decision **for each stage**.

3.4 Gibbs Sampling

- **Theorem 3.4** *The* ***Gibbs sampling method*** *is equivalent to the* ***composition*** *of d Metropolis-Hastings algorithms, with acceptance probabilities* ***uniformly*** *equal to* 1*.*

  In particular, the j-th Metropolis-Hastings algorithms has proposal function

  $$K_j(\boldsymbol{x}, \boldsymbol{x}') = \delta_{\boldsymbol{x}_{-j}}(\boldsymbol{x}'_{-j}) \times \pi_j(x'_j|\boldsymbol{x}_{-j}), \quad j = 1, \ldots, d \tag{24}$$

  where $\delta_{\boldsymbol{x}_{-j}}(\boldsymbol{x}'_{-j}) = 1$ if $\boldsymbol{x}_{-j} = \boldsymbol{x}'_{-j}$, otherwise equal to 0.

- The basic update of ***Gibbs sampling*** involves two steps:

  1. Split the multi-dimensional sample $\boldsymbol{X} := (X_j, \boldsymbol{X}_{-j})$.
  2. Update X'_j by conditional distribution $\pi_j(x_j|\boldsymbol{x}_{-j})$; Then $\boldsymbol{X}' := (X'_j, \boldsymbol{X}_{-j})$.

- As compared to Metropolis-Hastings algorithm, Gibbs sampling has a number of distinct features:

  – The ***acceptance rate*** of the Gibbs sampler is ***uniformly*** **equal to** 1. Therefore, every simulated value is accepted and the suggestions on the optimal acceptance rates for Metropolis-Hastings do not apply in this setting. This also means that ***convergence assessment*** for this algorithm should be treated differently than for Metropolis-Hastings techniques.

  – The use of the Gibbs sampler implies limitations on the choice of instrumental distributions and requires a **prior knowledge** of some analytical or ***probabilistic properties*** of π.

  – The Gibbs sampler is, by construction, ***multidimensional***. Even though some components of the simulated vector may be artificial for the problem of interest, or unnecessary for the required inference, the construction is still at least two-dimensional.

  – The Gibbs sampler does not apply to problems where *the number of parameters* ***varies***, because of the obvious lack of irreducibility of the resulting chain.

- Gibbs sampling is **preferred** in following situations:

  – For ***high dimensional*** joint distribution $\pi(\boldsymbol{x})$, ***the local factor*** $\pi(x_k|\boldsymbol{x}_{-k})$ can be represented in ***explicit function form*** and is ***easy to simulate***. For instance, in many probabilistic graphical models such as *Ising model*, *Gaussian graphical models*, *non-parametric Bayesian models*, *Hidden Markov models* etc., the local factor can explicitly simulated. Similarly, in many complex system, people have ***more knowledge on the local dynamics*** as compared to the overall system.

  - The target domain $\mathcal{X}$ contains **certain structures** such as *clusters*, *low-dimensional subspaces/sub-manifolds* etc. A simple Metropolis-Hastings algorithm such as random walk is ***inefficient*** since it spends too much efforts on exploration of low density regions. In high dimensional spaces, this type of random walk is doomed to fail. Instead, Gibbs sampling use conditional distribution resulting from constraining the target distribution π on certain subspaces. As a result, no rejection is incurred at sampling.

- Finally, note that Gibbs sampling is heavily affected by the ***parameterization*** (or *decomposition*) of space of distributions (like *coordinate descent*).

3.4.1 Slice Sampling

- The simplest Gibbs sampling is the ***Slice Sampling*** [Robert and Casella, 1999], which use random walk to generate *uniformly distributed* samples from the subgraph $\mathfrak{L}(f) := \{(x,u) : 0 \le u \le f(x)\}$.

- The ***Slice Sampling*** for simple distribution f is described as below:

  1. Starting from a point $(x,u) \in \mathfrak{L}(f)$, move along u-axis, i.e. sampling according to the conditional distribution
  $$U|X = x \sim \mathcal{U}\left\{u : 0 \le u \le f(x)\right\}.$$
  This results in a point $(x,u') \in \mathfrak{L}(f)$.

  2. Then given $(x,u') \in \mathfrak{L}(f)$, move along x-axis, i.e. sampling according to the conditional distribution
  $$X|U = u' \sim \mathcal{U}\left\{x : u' \le f(x)\right\}.$$
  This results in a point $(x',u') \in \mathfrak{L}(f)$.

- If f can be decomposed into k components
$$f(x) \propto \prod_{i}^{k} f_i(x)$$
where $f_i(x)$ is some positive functions, e.g. likelihoods, this decomposition can then be associated with k ***auxiliary variables*** ω_i, so that
$$f_i(x) = \int \mathbb{1}\left\{0 \le \omega_i \le f(x)\right\} d\omega_i.$$
Therefore, f is the marginal distribution of $(f, \omega_1, \ldots, \omega_k)$. This is called *de-marginalization* of f.

- The ***General Slice Sampler*** will be

  1. For $i = 1, \ldots, k$, sample $\omega_i^{(t+1)} \sim \mathcal{U}[0, f_i(x^{(t)})]$;

  2. sample $x^{(t+1)}$ from $\mathcal{U}(A_{t+1})$ where $A_{t+1} := \left\{x : f_i(x) \ge \omega_i^{(t+1)},\ i = 1, \ldots, k\right\}$

- There is no accept-reject step in this sampling since the samples are drawn from conditional distribution of the target which is uniform on $\mathfrak{L}(f) := \{(x,u) : 0 \le u \le f(x)\}$.

3.4.2 Gibbs Sampling

- Suppose $\boldsymbol{X} = [X_1, \ldots, X_d]$ and for given k, the rest of variables $\boldsymbol{X}_{-k} = [X_j, \forall\, j \neq k]$. Let $\pi(\cdot|\boldsymbol{x}_{-k})$ be the conditional distribution of target π on rest of variables $\boldsymbol{x}_{-k}$.
- The ***Random Scan Gibbs Sampling*** is described as below at $t+1$ iteration:

  1. Given $\boldsymbol{X}^{(t)} = [X_1^{(t)}, \ldots, X_d^{(t)}]$ from iteration t, ***randomly*** select a coordinate i from $\{1, \ldots, d\}$ with probability $(\alpha_1, \ldots, \alpha_d)$ (usually uniform $(1/d, \ldots, 1/d)$).
  2. Draw $X_i^{(t+1)}$ from conditional distribution $\pi(x_i|\boldsymbol{X}_{-i}^{(t)})$ and *leave the rest of variable **unchanged***. That is
  $$\boldsymbol{X}_{-i}^{(t+1)} = \boldsymbol{X}_{-i}^{(t)}$$

- The ***Systematic Scan Gibbs Sampling*** is described as below at $t+1$ iteration:

  1. Given $\boldsymbol{X}^{(t)} = [X_1^{(t)}, \ldots, X_d^{(t)}]$ from iteration t, then
     for $i = 1, \ldots, d$:
     (a) draw $X_i^{(t+1)}$ from conditional distribution
     $$\pi\left(x_i \middle| \boldsymbol{X}_{1:(i-1)}^{(t+1)}, \boldsymbol{X}_{(i+1):d}^{(t)}\right)$$

- It is easy to check that *every* single conditional update for both *Random Scan Gibbs Sampling* and *Systematic Scan Gibbs Sampling* **preserve the joint probability** π. Suppose $\boldsymbol{X}^{(t)} \sim \pi$, then $\boldsymbol{X}_{-i}^{(t)} \sim \pi(\boldsymbol{X}_{-i}^{(t)})$ its marginal distribution. Then
  $$\pi(X_i^{(t+1)}|\boldsymbol{X}_{-i}^{(t)}) \times \pi(\boldsymbol{X}_{-i}^{(t)}) = \pi\left(X_i^{(t+1)}, \boldsymbol{X}_{-i}^{(t)}\right),$$
  which means that after one update the new configuration follows the same distribution π.
- The transition kernel for Markov chain induced by *systematic scan Gibbs sampling* is
  $$K(\boldsymbol{x}, \boldsymbol{x}') = \prod_{j=1}^{d} \pi_j(x'_j|\boldsymbol{x}'_{1:(j-1)}, \boldsymbol{x}_{(j+1):d}). \tag{25}$$
- It can be shown in [Robert and Casella, 1999] that if transition kernel associated with Gibbs sampling is ***absolutely continuous***, then the Markov chain is Harris recurrent. The ergodic theorem also holds under the same condition. If the Markov chain is aperiodic, it will have limit distribution equal to the stationary distribution.
- It can also be shown in [Robert and Casella, 1999] that each sub-sequence $(X_i^{(t)}), i = 1, \ldots, d$ is a **Markov chain** with stationary distribution as its ***marginal distribution*** $\pi_i(x_i)$. Moreover, all of these sub-chains have *interleaving properties* to form *duality* so that the ergodicity of one chain can be generalized to other chains.

3.4.3 The Hammersley-Clifford Theorem

- A most surprising feature of the Gibbs sampler is that the *conditional distributions* contain ***sufficient information*** to produce a sample from the *joint distribution*. Note that its counterpart in optimization, the coordinate ascent algorithm does not necessarily lead to the *global maximum*, but may end up in a saddlepoint. It is, therefore, somewhat remarkable that the full conditional distributions perfectly summarize the joint density, although the set of **marginal distributions obviously fails to do so**.

- **Definition** Let $(X_1, X_2, \ldots, X_d) \sim \pi(x_1, x_2, \ldots, x_d)$, where π_i denotes the marginal distribution of X_i. If $\pi_i(x_i) > 0$ for every $i = 1, \ldots, d$, implies that $\pi(x_1, x_2, \ldots, x_d) > 0$, then π satisfies the ***positivity condition***.

- **Theorem 3.5** ***(Hammersley-Clifford)***
Under the ***positivity condition****, the joint distribution π satisfies*

$$\pi(x_1, x_2, \ldots, x_d) = \prod_{j=1}^{d} \frac{\pi_{\ell_j}(x_{\ell_j} \,|\, x_{\ell_1}, x_{\ell_2}, \ldots, x_{\ell_{j-1}}, x'_{\ell_{j+1}}, \ldots, x'_{\ell_d})}{\pi_{\ell_j}(x'_{\ell_j} \,|\, x_{\ell_1}, x_{\ell_2}, \ldots, x_{\ell_{j-1}}, x'_{\ell_{j+1}}, \ldots, x'_{\ell_d})} \tag{26}$$

for every permutation ℓ on $\{1, 2, \ldots, d\}$ and every $\boldsymbol{x}' \in \mathcal{X}$.

- From Hammersley-Clifford theorem, we see that ***any transition rule*** $A(x_j \to x'_j)$ that leaves the ***conditional distribution*** $\pi_j(x_j \,|\, \boldsymbol{x}_{-j})$ ***invariant*** will leave the joint distribution π invariant.

4 Hamiltonian Monte Carlo

4.1 Hamiltonian Dynamics

- **Theorem 4.1** ***(Hamilton's equations)*** *[Gelfand et al., 2000]*
Let $J[y]$ be a functional of the forms:

$$J[x^1, \ldots, x^n] = \int_a^b \mathcal{L}(x^1, \ldots, x^n, \dot{x}^1, \ldots, \dot{x}^n, t)dt \tag{27}$$

defined on the set of functions $x^i(t), i = 1, \ldots, n$ which have continuous first derivatives in $[a, b]$ and satisfy the boundary conditions $\boldsymbol{x}(a) = [x^1(a), \ldots, x^n(a)] = A$, $\boldsymbol{x}(b) = [x^1(b), \ldots, x^n(b)] = B$. The Hamiltonian $\mathcal{H}$ corresponding to functional J is defined as

$$\mathcal{H}(x^1, \ldots, x^n, v_1, \ldots, v_n, t) = \sum_i v_i \dot{x}^i - \mathcal{L}(x^1, \ldots, x^n, \dot{x}^1, \ldots, \dot{x}^n, t).$$

Then a ***necessary condition*** *for $J[x^1, \ldots, x^n]$ to have an extremum for a given function $\boldsymbol{x}(t) = [x^1(t), \ldots, x^n(t)]$ is that $\boldsymbol{x}(t)$ and $\boldsymbol{v}(t)$ satisfy* ***Hamilton's equations***

$$\frac{\partial \mathcal{H}}{\partial v_i} = \dot{x}^i, \quad \forall\, i \tag{28}$$

$$\frac{\partial \mathcal{H}}{\partial x^i} = -\dot{v}_i, \quad \forall\, i \tag{29}$$

$$\frac{\partial \mathcal{H}}{\partial t} = -\frac{\partial \mathcal{L}}{\partial t} \tag{30}$$

- The equation (29) is the definition of velocity or momentum (up to a constant scale) and the equation (28) is the Newton's law. The Hamiltonian dynamic $(\boldsymbol{x}(t), \boldsymbol{v}(t))$ is described by $2n$

first-order differential equations:

$$\begin{aligned}
\frac{dx^i}{dt} &= \frac{\partial \mathcal{H}}{\partial v_i}, \quad i = 1, \dots, n \\
\frac{dv_i}{dt} &= -\frac{\partial \mathcal{H}}{\partial x^i}, \quad i = 1, \dots, n && (31) \\
\Leftrightarrow \frac{d\boldsymbol{z}}{dt} &= \boldsymbol{\Omega}\, \nabla_{\boldsymbol{z}} \mathcal{H} && (32) \\
\text{where } \boldsymbol{z} := (\boldsymbol{x}, \boldsymbol{v}) \quad \boldsymbol{\Omega} &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_n \\ -\boldsymbol{I}_n & \boldsymbol{0} \end{bmatrix} \text{ is called structure matrix.} && (33)
\end{aligned}$$

- Let the ***Hamiltonian flow map*** $\phi_t : (\boldsymbol{x}, \boldsymbol{v}) \mapsto (\boldsymbol{x}_t(\boldsymbol{x}, \boldsymbol{v}), \boldsymbol{v}_t(\boldsymbol{x}, \boldsymbol{v}))$ be the position and velocity after time t starting from $(\boldsymbol{x}, \boldsymbol{v})$. The Hamiltonian flow has the following properties:

  – ***Reversibility***: Hamiltonian dynamics of the form mentioned above are ***time reversible*** for $t \geq 0$ [Brooks et al., 2011]:

  $$\phi_t(\boldsymbol{x}_t(\boldsymbol{x}, \boldsymbol{v}), -\boldsymbol{v}_t(\boldsymbol{x}, \boldsymbol{v})) = (\boldsymbol{x}, -\boldsymbol{v}). \tag{34}$$

  The mapping $\phi_t : (\boldsymbol{x}, \boldsymbol{v}) \mapsto (\boldsymbol{x}_t(\boldsymbol{x}, \boldsymbol{v}), \boldsymbol{v}_t(\boldsymbol{x}, \boldsymbol{v}))$ is one-to-one. It is seen that ϕ_t has inverse mapping.

  – ***Conservation of the Hamiltonian***: A second property of the dynamics is that it keeps the ***Hamiltonian invariant*** (i.e. *conserved*) [Brooks et al., 2011].

  $$\begin{aligned}
  \frac{d\mathcal{H}}{dt} &= \sum_{i=1}^{n} \left[\frac{dx^i}{dt} \frac{\partial \mathcal{H}}{\partial x^i} + \frac{dv_i}{dt} \frac{\partial \mathcal{H}}{\partial v_i} \right] \\
  &= \sum_{i=1}^{n} \left[\frac{dx^i}{dt} \frac{dv_i}{dt} - \frac{dv_i}{dt} \frac{dx^i}{dt} \right] = 0 && (35)
  \end{aligned}$$

  To see this note that, since $\mathcal{H}$ does not depend on t explicitly (or $\frac{\partial \mathcal{H}}{\partial t} = 0$).

  – ***Volume Preservation***: A third fundamental property of Hamiltonian dynamics is that it ***preserves volume in phase space*** $(\boldsymbol{x}, \boldsymbol{v})$ space. Formally, let $\boldsymbol{F} = (\frac{d\boldsymbol{x}}{dt}, \frac{d\boldsymbol{v}}{dt})$ be the vector field associated to the Hamiltonian in the phase space $\mathbb{R}^n \times \mathbb{R}^n$. First note that the ***divergence of $\boldsymbol{F}$ is zero*** [Brooks et al., 2011]:

  $$\begin{aligned}
  \nabla \cdot \boldsymbol{F} = \operatorname{div} \boldsymbol{F} &= \sum_{i=1}^{n} \left[\frac{\partial}{\partial x^i} \frac{dx^i}{dt} + \frac{\partial}{\partial v_i} \frac{dv_i}{dt} \right] \\
  &= \sum_{i=1}^{n} \left[\frac{\partial}{\partial x^i} \frac{\partial \mathcal{H}}{\partial v_i} - \frac{\partial}{\partial v_i} \frac{\partial \mathcal{H}}{\partial x^i} \right] = 0 && (36)
  \end{aligned}$$

  Since divergence represents the volume density of the outward flux of a vector field from an infinitesimal volume around a given point, it being zero everywhere implies volume preservation.

  – ***Symplecticness***: Volume preservation is also a consequence of Hamiltonian dynamics being ***symplectic*** [Brooks et al., 2011]. Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{v})$, and define $\boldsymbol{\Omega}$ as in (33)

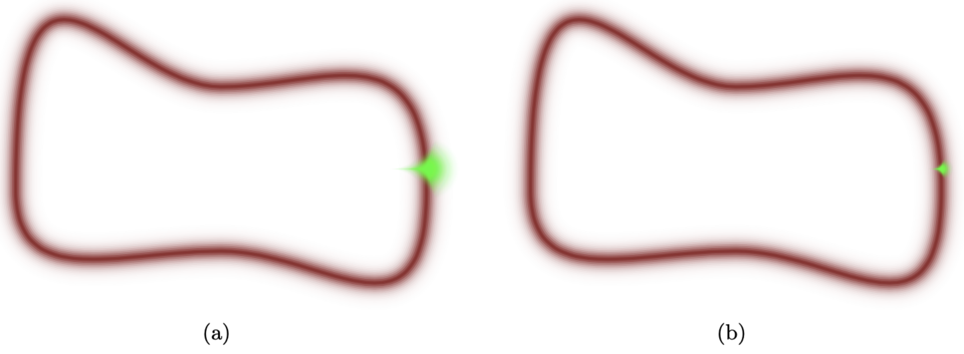

FIG 10. *In high dimensions, the Random Walk Metropolis proposal density (green) is strongly biased towards the outside of the typical set where the target density, and hence the Metropolis acceptance probability vanishes. (a) If the proposal variances are large then the proposals will stray too far away from the typical set and are rejected. (b) Smaller proposal variances stay within the typical set and hence are accepted, but the resulting transition density concentrates tightly around the initial point. Either way we end up with a Markov chain that explores the typical set very, very slowly.*

Figure 1: The challenge of random walk Metropolis-Hastings on high dimensional space. [Betancourt, 2017]

The ***symplecticness condition*** is that the Jacobian matrix, $(\partial_{\boldsymbol{z}} \boldsymbol{\phi}_t)$ of the mapping ϕ_t satisfies

$$(\partial_{\boldsymbol{z}} \boldsymbol{\phi}_t)^T \, \boldsymbol{\Omega}^{-1} \, (\partial_{\boldsymbol{z}} \boldsymbol{\phi}_t) = \boldsymbol{\Omega}^{-1} \tag{37}$$

Note $\boldsymbol{\Omega}^{-1} = \boldsymbol{\Omega}^T = -\boldsymbol{\Omega}$. This implies volume conservation i.e. $\det (\partial_{\boldsymbol{z}} \boldsymbol{\phi}_t)$ is one. When $n > 1$, the ***symplecticness condition is stronger than volume preservation***.

4.2 From MCMC to Hamilton Monte Carlo

4.2.1 Monte Carlo in high dimensional space

- One of the characteristic properties of **high-dimensional spaces** is that there is *much more volume outside any given neighborhood than inside of it.* On the other hand, the density is concentrated around the mode. Thus the region that contributes the most to the computation of expectation is not around mode nor in the tail. This set, referred as ***typical set***, is the focus on the simulation.

  As the dimension of parameter space increases, however, the tension between the density and the volume grows and the typical set becomes ***more singular*** [Betancourt, 2017].

- The **success** of Markov Chain Monte Carlo lies in its ability to explore the parameter space while ***preserving the target distribution***. The resulting Markov chain will *drift into and then across the typical set* regardless of its initial state, providing a powerful quantification of the typical set from which we can derive accurate expectation estimators.

- *Random-walk Metropolis-Hastings* is doomed to fail in high dimensional space due to its inefficiency and low acceptance rate. There are an exponential number of directions in which to guess but only a singular number of directions that stay within the typical set and pass the check.

- In order to make large jumps away from the initial point, and into new, unexplored regions of

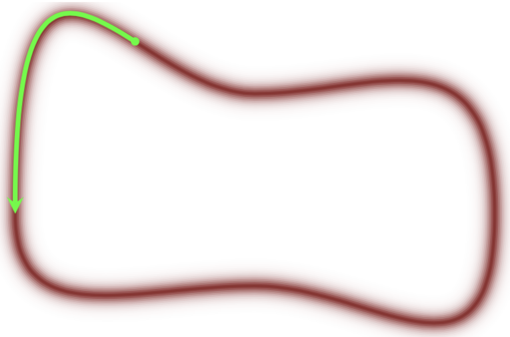

Fig 11. *Most Markov transitions are diffusive, concentrating around the initial point such that the corresponding Markov chains linger in small neighborhoods of the typical set for long periods of time. In order to maximize the utility of our computational resources we need* coherent *Markov transitions that are able to glide across the typical set towards new, unexplored neighborhoods.*

Figure 2: The Markov chain that is able to explore the contours of high probability mass is needed.

the typical set, we need to exploit information about the ***geometry*** of the *typical set* itself. Specifically, we need transitions that can follow those ***contours of high probability mass***, coherently gliding through the typical set.

- Hamiltonian Monte Carlo is the unique procedure for automatically generating this *coherent exploration* for sufficiently well-behaved target distributions.

4.2.2 Hamiltonian Monte Carlo

- Compare to MCMC, ***Hamiltonian Monte Carlo (HMC)*** [Brooks et al., 2011] has some unique characteristics:
  - HMC reply on the ***Hamiltonian dynamic*** to **explore** the parameter space. As oppose to the *stochastic process* from MCMC, the HMC exploration process is ***deterministic*** based on the *differential equations* (28), (29). HMC only need to simulate the **initial velocity/momentum** $\boldsymbol{v}(0)$ to start the Hamiltonian dynamic.
  - HMC ***preserves the target distribution*** π by *properties of Hamiltonian dynamic*, while the MCMC preserve the target distribution as the *stationary distribution* of Markov chain.
  - Unlike the *diffusion transition kernel* in MCMC, the transition process of HMC encodes ***geometrical information*** of the typical set via its associated ***vector field***. In other words, instead of fumbling around parameter space with random, uninformed jumps, we can follow the direction assigned to each at point for a small distance. Continuing this process traces out a coherent trajectory through the typical set that efficiently moves us far away from the initial point to new, unexplored regions of the typical set *as quickly as possible.*
  - The performance of MCMC and its special case, Gibbs sampling, depend on a particular ***parameterization*** of the target distribution. On the other hand, the design of Hamiltonian carefully remove the dependency on parameterization with additional geometric

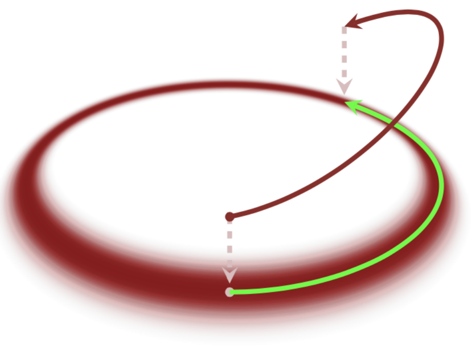

Fig 19. *By constructing a probability distribution on phase space that marginalizes to the target distribution, we ensure that the typical set on phase space projects to the typical set of the target distribution. In particular, if we can construct trajectories that efficiently explore the joint distribution (black) they will project to trajectories that efficiently explore the target distribution (green).*

Figure 3: By constructing a probability distribution on phase space that marginalizes to the target distribution, we ensure that the typical set on phase space projects to the typical set of the target distribution. [Betancourt, 2017]

constraints while twisting the directions to align with the typical set.

- Compared to Random-walk Metropolis-Hastings, HMC reduces the correlation between successive sampled states by proposing moves to distant states which maintain a high probability of acceptance.

- Inspired by classical mechanic systems, the **key idea** of ***Hamiltonian Monte Carlo (HMC)*** is introduce ***auxiliary momentum parameters*** in order to twist the gradient vector field into a vector field aligned with the typical set, and hence one capable of generating efficient exploration.

- Consider the target distribution as

$$\pi(\boldsymbol{x}, \boldsymbol{v}) \propto \frac{1}{Z} \exp\left(-\mathcal{H}(\boldsymbol{x}, \boldsymbol{v})\right), \tag{38}$$

  where $\boldsymbol{v}$ is the ***auxiliary momentum parameters***, $\boldsymbol{z} := (\boldsymbol{x}, \boldsymbol{v})$ is called ***phase-space parameterization***. The joint probability $\pi(\boldsymbol{x}, \boldsymbol{v})$ distribution on phase space is called the ***canonical distribution*** or phase-space distribution.

- Note that $\mathcal{H}(\boldsymbol{x}, \boldsymbol{v})$ is ***independent*** of the details of any ***parameterization***, so does the *canonical density* $\pi(\boldsymbol{x}, \boldsymbol{v})$ in (38). Moreover, $\mathcal{H}(\boldsymbol{x}, \boldsymbol{v})$ captures the ***invariant probabilistic structure*** of the phase-space distribution and, most importantly, the ***geometry*** of its typical set.

- We can factorize π into

$$\pi(\boldsymbol{x}, \boldsymbol{v}) = \pi(\boldsymbol{v}|\boldsymbol{x})\pi(\boldsymbol{x}) \tag{39}$$

$$\begin{aligned} -\log \pi(\boldsymbol{x}, \boldsymbol{v}) &= -\log \pi(\boldsymbol{v}|\boldsymbol{x}) - \log \pi(\boldsymbol{x}) \\ &= \mathcal{K}(\boldsymbol{x}, \boldsymbol{v}) + \mathcal{V}(\boldsymbol{x}) := \mathcal{H}(\boldsymbol{x}, \boldsymbol{v}). \end{aligned} \tag{40}$$

  where $\mathcal{K}(\boldsymbol{x}, \boldsymbol{v})$ is the *kinetic energy* and $\mathcal{V}(\boldsymbol{x})$ is the *potential energy*. The potential energy is completely determined by the target distribution while the *kinetic energy* is ***unconstrained*** and must be specified by the implementation.

This factorization guarantees that any trajectories exploring the typical set of the phase space distribution $\pi(\boldsymbol{x}, \boldsymbol{v})$ will project to trajectories exploring the typical set of the target distribution $\pi(\boldsymbol{x})$.

- Note that $\mathcal{H}(\boldsymbol{x}, \boldsymbol{v})$ is ***independent*** of the details of any ***parameterization***, so does the *canonical density* $\pi(\boldsymbol{x}, \boldsymbol{v})$ in (38). Moreover, $\mathcal{H}(\boldsymbol{x}, \boldsymbol{v})$ captures the ***invariant probabilistic structure*** of the phase-space distribution and, most importantly, the ***geometry*** of its typical set.
- Given (40), recall from (31) that

$$\begin{aligned} \frac{dx^i}{dt} &= \frac{\partial \mathcal{K}}{\partial v_i} = -\frac{\partial \log \pi(\boldsymbol{v}|\boldsymbol{x})}{\partial v_i}, \quad i = 1, \ldots, d \\ \frac{dv_i}{dt} &= -\frac{\partial \mathcal{K}}{\partial x^i} - \frac{\partial \mathcal{V}}{\partial x^i} = \frac{\partial \log \pi(\boldsymbol{v}|\boldsymbol{x})}{\partial x^i} + \frac{\partial \log \pi(\boldsymbol{x})}{\partial x^i}, \quad i = 1, \ldots, d \end{aligned} \tag{41}$$

4.3 Idealized Hamiltonian Monte Carlo

4.3.1 Formulation

- Let $\phi_t : (\boldsymbol{x}, \boldsymbol{v}) \mapsto (\boldsymbol{x}_t(\boldsymbol{x}, \boldsymbol{v}), \boldsymbol{v}_t(\boldsymbol{x}, \boldsymbol{v}))$ be the trajectory characterized by differential equations (41). $\phi_t(\boldsymbol{x}, \boldsymbol{v})$ is the position and velocity/momentum at time t starting from $(\boldsymbol{x}, \boldsymbol{v})$.
- The ***idealized Hamiltonian Monte Carlo*** [Betancourt, 2017, Vishnoi, 2021] is described as below:
  1. For $t = 1, 2, \ldots, k$:
     (a) Given $\boldsymbol{X}_{t-1}$, generate a momentum $\boldsymbol{V}_{t-1}$ from conditional distribution $\pi(\boldsymbol{v}|\boldsymbol{X}_{t-1})$;
     (b) Set $(\boldsymbol{X}_t, \boldsymbol{V}_t) = \phi_T(\boldsymbol{X}_{t-1}, \boldsymbol{V}_{t-1})$ by integrating Hamilton's equations for some time T
  2. Return $\boldsymbol{X}_k$ after projecting away the momentum.

  Composing these steps together yields a ***Hamiltonian Markov transition*** composed of *random trajectories* that rapidly explore the target distribution.
- Using energy level set for Hamiltonian,

$$\mathcal{H}^{-1}(E) = \{(\boldsymbol{x}, \boldsymbol{v}) : \mathcal{H}(\boldsymbol{x}, \boldsymbol{v}) = E\},$$

  which are all $(2\,d - 1)$-dimensional, ***compact*** surfaces in phase space. These ***concentric level sets*** form a ***foliation***. The joint distribution $\pi(\boldsymbol{x}, \boldsymbol{v})$ thus can be decomposed via conditionals on energy level set:

$$\pi(\boldsymbol{x}, \boldsymbol{v}) = \pi(\boldsymbol{\theta}_E \,|\, E)\pi(E),$$

  .
- We can see that the entire Hamiltonian Markov chain decouples into ***two distinct phases***:
  1. ***deterministic*** exploration of individual *level sets*

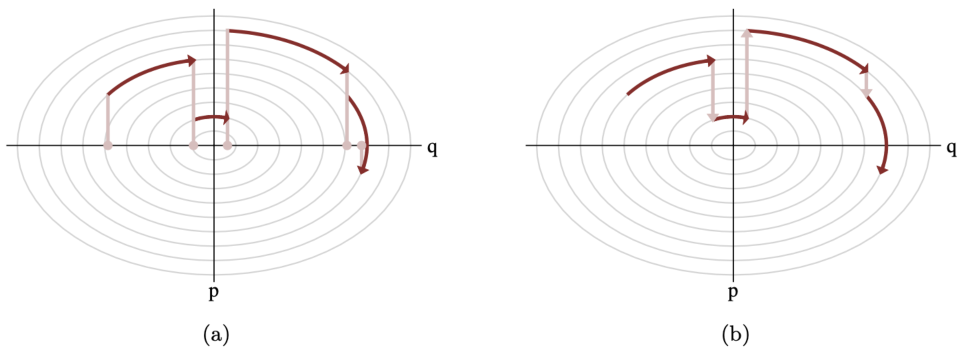

FIG 22. *(a) Each Hamiltonian Markov transition lifts the initial state onto a random level set of the Hamiltonian, which can then be explored with a Hamiltonian trajectory before projecting back down to the target parameter space. (b) If we consider the projection and random lift steps as a single momentum resampling step, then the Hamiltonian Markov chain alternates between deterministic trajectories along these level sets (dark red) and a random walk across the level sets (light red).*

Figure 4: Using level-sets, we see that HMC first explore within level set and then jump from one level set to another via random walk. [Betancourt, 2017]

2. a ***stochastic*** exploration ***between*** the level sets themselves

- **Theorem 4.2** ***(HMC Preserves the Target Density)*** *[Vishnoi, 2021] Suppose* $(\boldsymbol{X}, \boldsymbol{V})$ *is a sample from the density*

$$\pi(\boldsymbol{x}, \boldsymbol{v}) = \frac{1}{Z} \exp\left(-\mathcal{H}(\boldsymbol{x}, \boldsymbol{v})\right) d\mu(\boldsymbol{x}, \boldsymbol{v})$$

where $Z = \int \exp\left(-\mathcal{H}(\boldsymbol{x}, \boldsymbol{v})\right) d\mu(\boldsymbol{x}, \boldsymbol{v})$ *is the partition function. Let* $T > 0$ *be the step size of the HMC. Then the density of* $\phi_T(\boldsymbol{X}, \boldsymbol{V})$ *is* π *for any* $T \geq 0$. *Moreover the density of* $\phi_T(\boldsymbol{X}_{t-1}, \boldsymbol{V}_{t-1})$, *where* $\boldsymbol{V}_{t-1} \sim \pi(\boldsymbol{v}|\boldsymbol{X}_{t-1})$ *is also* π. *Thus, the* ***idealized HMC algorithms preserves*** π.

4.3.2 Euclidean-Gaussian Kinetic Energies

- The first substantial ***degree of freedom*** in the Hamiltonian Monte Carlo method that we can tune is the choice of the conditional probability distribution over the momentum or, equivalently, **the choice of a kinetic energy function**. Along with the target distribution, this choice completes the probabilistic structure on phase space which then determines the geometry of the microcanonical decomposition.

- The simplest choice is to allow velocity/momentum $\boldsymbol{V}_t$ being ***independent*** of $\boldsymbol{X}_t$, i.e. $\pi(\boldsymbol{v}|\boldsymbol{x}) := g(\boldsymbol{v})$. This is equivalent to assuming that *the phase-space geometry* is ***flat***. The Hamiltonian Monte Carlo corresponds to a *random-walk* across a foliation of the phase-space.

- A natural choice of $g(\boldsymbol{v})$ is Normal distribution $\mathcal{N}(\boldsymbol{v} \,|\, \boldsymbol{0}, \boldsymbol{\Sigma}_v)$. Here, the covariance matrix $\boldsymbol{\Sigma}_v$ encodes an Euclidean metric in the phase-space. The corresponding kinetic energy is a ***Euclidean-Gaussian kinetic energy***

$$\mathcal{K}(\boldsymbol{v}) = \frac{1}{2} \boldsymbol{v}^T \boldsymbol{\Sigma}_v^{-1} \boldsymbol{v} + \log\det \boldsymbol{\Sigma}_v + const.. \tag{42}$$

And the Hamilton's equation is simplified as

$$\frac{d\boldsymbol{x}}{dt} = \nabla_{\boldsymbol{v}}\mathcal{K} = \boldsymbol{\Sigma}_v^{-1}\boldsymbol{v}$$
$$\frac{d\boldsymbol{v}}{dt} = -\nabla_{\boldsymbol{x}}\mathcal{V} = \nabla_{\boldsymbol{x}}E \tag{43}$$

where the target distribution $\pi(\boldsymbol{x}) \propto \exp(-E(\boldsymbol{x}))$.

- Because the Euclidean structure over the momentum is dual to the Euclidean structure over the parameters, its interactions with the target distribution are straightforward to derive. Applying the transformation $\boldsymbol{p}' = \boldsymbol{\Sigma}_v^{-1/2}\boldsymbol{v}$ simplifies the kinetic energy, but remember that we have to apply the ***opposite transformation*** to the parameters, $\boldsymbol{x}' = \boldsymbol{\Sigma}_v^{1/2}\boldsymbol{x}$, to preserve the Hamiltonian geometry. Consequently, a choice of $\boldsymbol{\Sigma}_v^{-1}$ effectively ***rotates*** and then ***rescales*** the target **parameter** space, potentially ***correlating*** or ***de-correlating*** the target distribution and correspondingly ***warping*** the energy level sets.

  In particular, as the inverse Euclidean metric more closely resembles the covariance of the target distribution it de-correlates the target distribution, resulting in energy level sets that are more and more uniform and hence easier to explore.

4.4 Hamiltonian Monte Carlo in Practice

- The ***main obstruction*** to implementing the Hamiltonian Monte Carlo method is generating the Hamiltonian trajectories themselves. Aside from a few trivial examples, we *cannot solve Hamilton's equations exactly* and any implementation must instead solve them *numerically.* **Numerical inaccuracies**, however, can quickly compromise the utility of even the most well-tuned Hamiltonian transition.
- Common ODE solvers suffer from an issue of ***drift***. As we numerically solve longer and longer trajectories the error in the solvers adds coherently, pushing the approximate trajectory away from the true trajectory and the typical set that we want to explore. Moreover, the magnitude of this drift rapidly increases with the dimension of phase space.
- Fortunately, we can use the geometry of phase space itself to construct an extremely powerful family of numerical solvers, known as ***symplectic integrators*** [Liu, 2001, Leimkuhler and Reich, 2004, Haier et al., 2006, Betancourt, 2017, Vishnoi, 2021], that are robust to phenomena like drift and enable high-performance implementations of the Hamiltonian Monte Carlo method.

4.4.1 Symplectic Integrators

- Symplectic integrators are powerful because the numerical trajectories they generate exactly ***preserve phase space volume***, just like the Hamiltonian trajectories they are approximating. Consequently, the numerical trajectories cannot drift away from the exact energy level set, instead oscillating near it even for long integration times.
- For independent momentum distribution like the Euclidean-Gaussian kinetic energy, the symplectic integrator is called ***leapfrog integrator*** [Brooks et al., 2011].
- The ***Leapfrog Integrator*** is described as below, where ϵ is the time discretization, or step size:

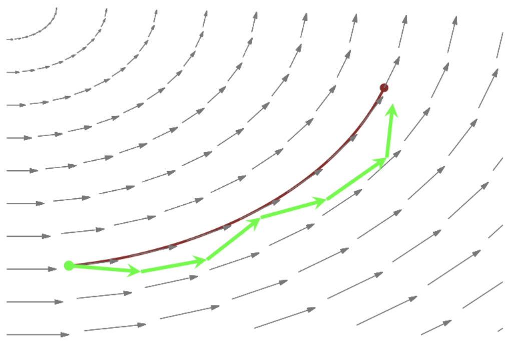

Fig 28. *Symplectic integrators generate numerical trajectories that are incompressible like the exact Hamiltonian trajectory they approximate. Consequently their approximation error cannot add up coherently to pull the numerical trajectories away from the exact trajectories. Instead the numerical trajectories oscillate around the exact level set, even as we integrate for longer and longer times.*

Figure 5: Symplectic integrator will oscillate near exact level set even for long integration times. [Betancourt, 2017]

1. Initialization: $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $\boldsymbol{v}_0 \leftarrow \boldsymbol{v}$.
2. For $0 \le t \le \lfloor \frac{T}{\epsilon} \rfloor$:

$$\begin{aligned}
&(a) \quad \boldsymbol{v}_{t+\frac{1}{2}} \leftarrow \boldsymbol{v}_t - \frac{\epsilon}{2}\nabla_{\boldsymbol{x}}\mathcal{V}(\boldsymbol{x}_t)\\
&(b) \quad \boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t + \epsilon\,\boldsymbol{\Sigma}_v^{-1}\boldsymbol{v}_{t+\frac{1}{2}}\\
&(c) \quad \boldsymbol{v}_{t+1} \leftarrow \boldsymbol{v}_{t+\frac{1}{2}} - \frac{\epsilon}{2}\nabla_{\boldsymbol{x}}\mathcal{V}(\boldsymbol{x}_{t+1})
\end{aligned}$$

This simple but precise interleaving of discrete momentum and position updates ensures exact volume preservation on phase space, and hence the accurate numerical trajectories we need to realize the potential of a Hamiltonian transition.

- Employing symplectic integrators provides the opportunity to translate the theoretical performance of the Hamiltonian Monte Carlo method into a practical implementation. There remain, however, two obstructions to realizing this translation.
  - First, even though symplectic integrators are highly accurate, the small errors they do introduce will bias the resulting Hamiltonian transitions without an exact correction.
  - Second, we have to be able to select a symplectic integrator well-suited to a given target distribution.

4.4.2 Correcting for Symplectic Integrator Error

- One particularly natural strategy for correcting the bias introduced by the error in a symplectic integrator is to treat the ***Hamiltonian transition as the proposal*** for a Metropolis-Hastings scheme on phase space.
- Suppose $(\boldsymbol{x}_L, \boldsymbol{v}_L)$ be the last state of L symplectic integrator steps starting from $(\boldsymbol{x}_0, \boldsymbol{v}_0)$. **One issue** with computing the Metropolis-Hastings acceptance rate is that the transition kernel for HMC is determined by the symplectic integrator which can only move forward *not backward*. In order to obtain the ***reversible chain***, we need to use the trick that ***flipping***

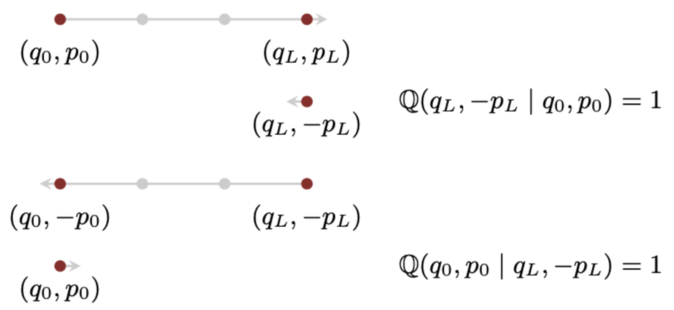
(q_0, p_0)
(q_L, p_L)
$(q_L, -p_L)$
$\mathbb{Q}(q_L, -p_L \mid q_0, p_0) = 1$
$(q_0, -p_0)$
$(q_L, -p_L)$
(q_0, p_0)
$\mathbb{Q}(q_0, p_0 \mid q_L, -p_L) = 1$

FIG 31. *Augmenting the numerical trajectory with a momentum flip defines a reversible Metropolis-Hastings proposal for which the forwards and backwards proposal probabilities are both well-behaved and we recover a valid correction scheme.*

Figure 6: Obtaining the reverse chain by flipping the sign of momentum. Note that the kinetic energy is symmetric to the change of sign in momentum. [Betancourt, 2017]

the sign of momentum *will not change the kinetic energy*, thus not changing the overall distribution. Thus we will have the Hastings ratio computed as

$$\begin{aligned} r(\boldsymbol{x}_0, \boldsymbol{v}_0; \boldsymbol{x}_L, -\boldsymbol{v}_L) &= \frac{\pi(\boldsymbol{x}_L, -\boldsymbol{v}_L)\, K((\boldsymbol{x}_L, -\boldsymbol{v}_L); (\boldsymbol{x}_0, \boldsymbol{v}_0))}{\pi(\boldsymbol{x}_0, \boldsymbol{v}_0) K((\boldsymbol{x}_0, \boldsymbol{v}_0); (\boldsymbol{x}_L, -\boldsymbol{v}_L))} \\ &= \frac{\pi(\boldsymbol{x}_L, -\boldsymbol{v}_L)}{\pi(\boldsymbol{x}_0, \boldsymbol{v}_0)} \\ &= \exp\left(-\mathcal{H}(\boldsymbol{x}_L, -\boldsymbol{v}_L) + \mathcal{H}(\boldsymbol{x}_0, \boldsymbol{v}_0)\right) \end{aligned} \tag{44}$$

- Finally, we have <u>***Metropolized Hamiltonian Monte Carlo***</u> [Brooks et al., 2011, Betancourt, 2017]:
  1. For $t = 1, 2, \ldots, k$:
     (a) Generate a momentum $\boldsymbol{V}_{t-1}$ from Normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_v)$;
     (b) Set $(\boldsymbol{X}', \boldsymbol{V}') = \widehat{\phi}_T(\boldsymbol{X}_{t-1}, \boldsymbol{V}_{t-1})$ as the last state of T symplectic integrator steps starting from $(\boldsymbol{X}_{t-1}, \boldsymbol{V}_{t-1})$.
     (c) Compute the Hastings ratio:
     $$r(\boldsymbol{X}_{t-1}, \boldsymbol{V}_{t-1}; \boldsymbol{X}', -\boldsymbol{V}') = \exp\left(-\mathcal{H}(\boldsymbol{X}', -\boldsymbol{V}') + \mathcal{H}(\boldsymbol{X}_{t-1}, \boldsymbol{V}_{t-1})\right)$$
     (d) Accept $\boldsymbol{X}_t = \boldsymbol{X}'$ with probability
     $$\alpha(\boldsymbol{X}_{t-1}, \boldsymbol{V}_{t-1}; \boldsymbol{X}', -\boldsymbol{V}') = \min\left\{1,\ r(\boldsymbol{X}_{t-1}, \boldsymbol{V}_{t-1}; \boldsymbol{X}', -\boldsymbol{V}')\right\}$$
     (e) Otherwise, accept $\boldsymbol{X}_t = \boldsymbol{X}_{t-1}$.
- Symplectic integrators are not exactly energy preserving, causing their numerical trajectories to deviate from the target energy level set. In particular, sampling a state uniformly from any numerical trajectory will not generate a sample from the canonical distribution. This error, however, can be exactly corrected by sampling from the trajectory not uniformly but rather with weights proportional to the desired canonical density function [Betancourt, 2017].

- Regarding the ***robustness*** of the HMC, preliminary results show that even simple implementations of the Hamiltonian Monte Carlo method are geometrically ergodic over a large class of target distributions, larger than the class for non-gradient based algorithms like Random-walk Metropolis-Hastings.
- There also some concerns in practice:
  - In practice, ***longer trajectory*** T is preferred so that the end state is less correlated with the initial state. However, it will have larger cost of simulation time.
  - Also, if the ***kinetic energy*** is *poorly-chosen* then the marginal energy distribution can become heavy-tailed itself in which case the stochastic exploration between level sets will become so slow that after any finite number of transitions the exploration of the Markov chain will be incomplete.
  - Another common obstruction to geometric ergodicity is neighborhoods in parameter space where the target distribution exhibits ***large curvature***. Most Markov transitions are not able to resolve these narrow neighborhoods, resulting in incomplete exploration and biased Markov chain Monte Carlo estimators.
  - Finally, HMC requires computation on the ***gradients*** in the symplectic integrator. It would be computational expensive if the target distribution has complex form.

5 Monte Carlo Optimization

- Similar to the problem of integration, differences between the numerical approach and the simulation approach to the problem

$$\max_{\boldsymbol{x}\in\mathcal{X}} h(\boldsymbol{x}) \tag{45}$$

  lie in the treatment of the function h.
  - In approaching an optimization problem using deterministic numerical methods, the ***analytical properties*** of the target function (*convexity*, *boundedness*, *smoothness*) are often paramount.
  - For the simulation approach, we are more concerned with h from a ***probabilistic*** (rather than analytical) point of view.
- We want to distinguish between two approaches to ***Monte Carlo optimization*** [Robert and Casella, 1999].
  - The first is an ***exploratory approach***, in which the goal is to optimize the function h by *describing its entire range*. The *actual properties* of the function play a *lesser role* here, with the Monte Carlo aspect more closely tied to the ***exploration of the entire space*** $\mathcal{X}$, even though, for instance, the slope of h can be used to speed up the exploration.
  - The second approach is based on a ***probabilistic approximation*** of the *objective function h* and is somewhat of a *preliminary step* to the actual optimization. Here, the Monte Carlo aspect ***exploits the probabilistic properties of the function*** h to come up with an acceptable approximation and is less concerned with exploring $\mathcal{X}$. For instance, *Missing data methods*, such as the *EM algorithm*, are closely related to tied to this idea.

5.1 Gradient Methods

- The gradient method is a *deterministic* numerical approach to the problem (45). It produces a sequence $(\boldsymbol{x}_t)$ that converges to the exact solution of (45), $\boldsymbol{x}^*$, when the domain $\mathcal{X} \subseteq \mathbb{R}^d$ and the function $(-h)$ are both *convex*. The sequence $(\boldsymbol{x}_t)$ is constructed in a ***recursive*** manner through

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \alpha_t \nabla h(\boldsymbol{x}_t), \quad \alpha_t > 0.$$

- In more general setups (that is, when the function or the space is less regular), equation above can be modified by stochastic perturbations to again achieve convergence. One of these stochastic modifications is to choose a second sequence (β_t) to define the chain $(\boldsymbol{x}_t)$ by

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \frac{\alpha_t}{2\beta_t} \Delta\, h(\boldsymbol{x}_t, \beta_t \boldsymbol{\xi}_t)\boldsymbol{\xi}_t, \quad \alpha_t > 0. \tag{46}$$

$$\text{where } \Delta\, h(\boldsymbol{x}_t, \beta_t \boldsymbol{\xi}_t) = (h(\boldsymbol{x}_t + \beta_t \boldsymbol{\xi}_t) - h(\boldsymbol{x}_t - \beta_t \boldsymbol{\xi}_t)) \approx 2\,\beta_t\, \|\boldsymbol{\xi}_t\|_2\, \nabla h(\boldsymbol{x}_t),$$

  and the variables $\boldsymbol{\xi}_t$ are uniformly distributed on the unit sphere $\|\boldsymbol{\xi}_t\|_2 = 1$.

  In contrast to the deterministic approach, this method does not necessarily proceed along the *steepest slope* in $(\boldsymbol{x}_t)$, but this property is sometimes a *plus* in the sense that it may *avoid being trapped in* ***local maxima*** or in ***saddlepoints*** of h.

- Consider the objective function is the sample average of log-likelihood function $h(\boldsymbol{\theta}) := \sum_{i=1}^{N} \ell(\boldsymbol{\theta}; \boldsymbol{x}_i)$. The ***stochastic gradient ascent*** is of form

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_t; \boldsymbol{\xi}_t), \quad \alpha_t > 0. \tag{47}$$

  where $\boldsymbol{\xi}_t \sim \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is a random sample from data.

5.2 Simulated Annealing

- The ***fundamental idea*** of ***simulated annealing methods*** is that a change of scale, called ***temperature***, allows for faster moves on the surface of the function h to maximize, whose negative is called ***energy***. Therefore, rescaling partially avoids the trapping attraction of local maxima.

- Given a temperature parameter $T > 0$, a sample $(\boldsymbol{x}_t)_t$. is generated from the distribution

$$\pi(\boldsymbol{x}) \propto \exp\left(\frac{h(\boldsymbol{x})}{T}\right) \tag{48}$$

  As T decreases toward 0, the values simulated from this distribution become concentrated in a narrower and narrower neighborhood of the *local maxima* of h

- In Simulated Annealing, we run a fixed number of N_k iterations of MCMC (*Metropolis-Hastings* or *Gibbs sampling*) so that $(\boldsymbol{X}_t^{(k)})$ follows the stationary distribution $\pi_k(\boldsymbol{x}) \propto \exp(h(\boldsymbol{x})/T_k)$. Then we decrease temperature T_k to T_{k+1} and restart the MCMC with last update $\boldsymbol{X}_k$. As $T_k \to 0$, the target distribution is approaching to an indicator on its mode, thus the simulated samples would be close to the optimal solution.

- The ***Simulated Annealing algorithm*** can be implemented as

1. Simulate $\boldsymbol{\xi}$ from an *instrumental distribution* with density $g(\|\boldsymbol{\xi} - \boldsymbol{X}_t\|)$;
2. Accept $\boldsymbol{X}_{t+1} = \boldsymbol{\xi}$ with probability

$$\alpha(\boldsymbol{X}_t, \boldsymbol{\xi}) = \min\left\{1, \exp\left(\frac{\Delta h_t}{T_t}\right)\right\}$$

   where $\Delta h_t := h(\boldsymbol{\xi}) - h(\boldsymbol{X}_t)$;
3. Otherwise, accept $\boldsymbol{X}_{t+1} = \boldsymbol{X}_t$.
4. Update T_t to T_{t+1}.

- Since there is non-zero probability of accepting a new proposal even if it is not local maximal, it allows the algorithm to *escape the attraction* of $\boldsymbol{x}_t$ if $\boldsymbol{x}_t$ is a local maximum of h. The temperature T_k controls the probability of acceptance of non-optimal proposal.

5.3 Simulated Tempering

- ***Simulated Tempering (ST)*** is proposed to let a MCMC scheme move more freely in the state space [Liu, 2001].
- Define a new target distribution, $\pi(\boldsymbol{x}, i) \propto c_i \exp\{-h(\boldsymbol{x})/T_i\}$ on the augmented space $(\boldsymbol{x}, i) \in \mathcal{X} \times I$. Here, the c_i are constants that can be controlled by the user and they should be tuned so that each tempered distribution in the system should have a roughly equal chance to be visited. Ideally, the c_i should be proportional to the reciprocal of the i-th partition function, $Z_i = \int \exp\{-h(\boldsymbol{x})/T_i\}$.
- The ***Simulated Tempering (ST)*** is described as
  1. With the current state $(\boldsymbol{X}_t, i_t) = (\boldsymbol{x}, i)$, we draw $u \sim \mathcal{U}[0,1]$.
  2. (***Update sample***) If $u \le \alpha_0$, we let $i_{t+l} = i$ and let $\boldsymbol{X}_{t+1}$ be drawn from a ***MCMC transition*** $K_i(\boldsymbol{x}, \boldsymbol{x}_{t+1})$ that leaves $\pi_i \propto \exp(h(\boldsymbol{x})/T_i)$ invariant.
  3. (***Update temperature***) If $u > \alpha_0$, we let $\boldsymbol{X}_{t+l} = \boldsymbol{x}$ and propose a ***level transition***, $i \to i'$, from a transition function $\alpha(i, i')$ (usually a nearest-neighbor simple random walk with reflecting boundary), and let $i_{t+1} = i'$ with probability

$$\min\left\{1, \frac{c_{i'}\,\pi_{i'}(\boldsymbol{x})\,\alpha(i', i)}{c_i\,\pi_i(\boldsymbol{x})\,\alpha(i, i')}\right\};$$

     otherwise accept $i_{t+1} = i$.

5.4 Monte Carlo EM

- Given observation y and latent variables $\boldsymbol{x}$, the objective function for missing data problem is the *incomplete likelihood*

$$\ell(\boldsymbol{\theta}|\boldsymbol{y}) = \mathbb{E}\left[\ell_c(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X})\right] = \int_{\mathcal{X}} \ell_c(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x}$$

  We refer to the function $\ell_c(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ as the *complete-data likelihood*, which corresponds to the observation of the complete data $(\boldsymbol{x}, \boldsymbol{y})$ and $g(\boldsymbol{y}|\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}|\boldsymbol{y})$ is marginal distribution.

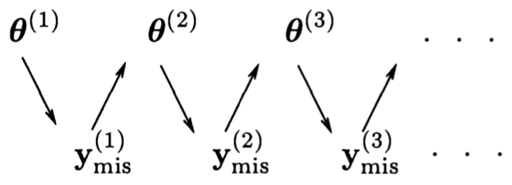
$\boldsymbol{\theta}^{(1)}$
$\boldsymbol{\theta}^{(2)}$
$\boldsymbol{\theta}^{(3)}$
$\mathbf{y}_{\text{mis}}^{(1)}$
$\mathbf{y}_{\text{mis}}^{(2)}$
$\mathbf{y}_{\text{mis}}^{(3)}$

FIGURE 6.2. A graphical illustration of the data augmentation scheme.

Figure 7: Scheme for iteratively fill in missing data and update parameter [Liu, 2001]

- We can obtain lower bound of incomplete log-likelihood by introducing the conditional distribution $q(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}) := \frac{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta})}{g(\boldsymbol{x}|\boldsymbol{\theta})}$:

$$\begin{aligned}\log \ell(\boldsymbol{\theta}|\boldsymbol{y}) = \log \int_{\mathcal{X}} p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}) d\boldsymbol{x} &= \log \int_{\mathcal{X}} q(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}) \frac{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta})}{q(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})} d\boldsymbol{x} \\ &\geq \int_{\mathcal{X}} q(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta})}{q(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})} d\boldsymbol{x} \\ &= \mathbb{E}_q \left[\ell_c(\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{y})\right] - \mathbb{E}_q \left[q(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})\right].\end{aligned}$$

  Denote the expected log-likelihood by

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}_0,\boldsymbol{y}) := \mathbb{E}_{\boldsymbol{\theta}_0} \left[\ell_c(\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{y})\right]$$

- The ***EM (Expectation-Maximization) algorithm*** consists of two steps iteratively:
  1. (**E-Step**): Compute $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t,\boldsymbol{y}) = \mathbb{E}_{\boldsymbol{\theta}_t}\left[\ell_c(\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{y})\right]$ with respect to $q(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}_t)$
  2. (**M-Step**): Maximize parameter $\boldsymbol{\theta}_{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t,\boldsymbol{y})$
- It can be shown that EM algorithm increase the incomplete log-likelihood at each iteration:

$$\log \ell(\boldsymbol{\theta}_t|\boldsymbol{y}) \geq \log \ell(\boldsymbol{\theta}_{t-1}|\boldsymbol{y})$$

- A difficulty with the implementation of the EM algorithm is that each "E-step" requires the computation of the expected log likelihood $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_0,\boldsymbol{y})$. An alternative solution is to approximate the expectation via Monte Carlo simulation, i.e.

$$\widehat{Q}_m(\boldsymbol{\theta}|\boldsymbol{\theta}_0,\boldsymbol{y}) = \frac{1}{m}\sum_{t=1}^{m} \ell_c(\boldsymbol{\theta}|\boldsymbol{X}_i,\boldsymbol{y}_i) \tag{49}$$

  where samples of missing data $\boldsymbol{X}_i \sim q(\boldsymbol{x}|\boldsymbol{y}_i,\boldsymbol{\theta}_0), i = 1,\ldots,m$.
- As $m \to \infty$, $\widehat{Q}_m(\boldsymbol{\theta}|\boldsymbol{\theta}_0,\boldsymbol{y}) \to Q(\boldsymbol{\theta}|\boldsymbol{\theta}_0,\boldsymbol{y})$.

References

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

Izrail Moiseevitch Gelfand, Richard A Silverman, et al. *Calculus of variations*. Courier Corporation, 2000.

Ernst Haier, Christian Lubich, and Gerhard Wanner. *Geometric Numerical integration: structure-preserving algorithms for ordinary differential equations*. Springer, 2006.

Benedict Leimkuhler and Sebastian Reich. *Simulating hamiltonian dynamics*. Number 14. Cambridge university press, 2004.

Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 10. Springer, 2001.

Christian P Robert and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Nisheeth K Vishnoi. An introduction to hamiltonian monte carlo method for sampling. *arXiv preprint arXiv:2108.12107*, 2021.