【20230405】-【20230425】

Basic sampling algorithms

对应 **11.1 Basic Sampling Algorithms** 小节，也可参考白板推导的相关笔记。

11.1 Basic Sampling Algorithms 小节基于概率密度转换公式（可参见Lecture1.tex (illinois.edu)）介绍了如何根据均匀分布得到任意目标分布的样本。

11.1.2 Rejection sampling 小节讨论了拒绝采样：对从提议分布 $q(z)$ 中采得的样本 $z^{(i)}$，以概率 $\frac{p(z^{(i)})}{kq(z^{(i)})}$ 接受这个样本（也就是 $1-\frac{p(z^{(i)})}{kq(z^{(i)})}$ 的概率拒绝或者说丢弃这个样本）。若 $z^{(i)}$ 被接受，则经过上述步骤，$z^{(i)} \sim p(z)$。

拒绝采样中 k 即要保证 $kq(z)$ 为 $p(z)$ 的上界，即 $kq(z) \geqslant p(z)$，$\text{for all } z$。同时，为了保证采样效率，不使拒绝率过高，k 要尽可能小，提议分布也要与目标分布尽可能相近。

Exercise 11.6 给出了拒绝采样的证明，我们也可以用一种很直观的方法去理解其正确性：和 slice sampling 一样上升一个维度，在 $p(z)$ 所围成的区域中均匀采样，最后丢弃新引入的维度 (不妨记为 u) 即可。 但是，$p(z)$ 的形状不规律不方便采样，为此，我们在一个包含 $p(z)$ 的更大的区域 $kq(z)$ 中采样，再判断 $(z^{(i)}, u^{(i)})$ 是否落入 $p(z)$ 的区域，即若 $u^{(i)} < p(z^{(i)})$ 则保留此样本，否则丢弃。经过上述操作得到的样本 $z^{(i)}$ 就服从目标分布 $p(z)$。对 $kq(z)$ 所围区域的采样又可以分为两步完成，首先是对 $q(z)$ 的采样 $z^{(i)}$，然后再对 $u|z^{(i)} \sim U[0, kq(z^{(i)})]$ 采样，最后再与 $p(z^{(i)})$ 比较大小，判断是否丢弃。而这就等价于拒绝分布中从 $U[0,1]$ 中采样后与接受率 $\frac{p(z^{(i)})}{kq(z^{(i)})}$ 比较大小，判断是否丢弃的操作。

11.1.3 Adaptive rejection sampling 小节则给出了目标分布 $p(x)$ log concave 时提议分布 $q(x)$ 的构造方法，且提议分布在采样过程中会被不断优化 (adaptive)。

11.1.4 Importance sampling 并不是一种采样方法，而是基于提议分布计算函数期望的方法。提议分布的概念不只存在于拒绝采样，当然也不一定存在拒绝其样本的步骤，它可以被理解为是人为选定作为基准的分布。**11.6 Estimating the Partition Function** 小节就是在讨论如何基于重要性采样计算配分函数，也就是归一化常数。

11.1.5 Sampling-importance-resampling 可以用于采样，也可以用于计算函数期望：

- 用于采样：它基于提议分布，构造了一个目标分布的近似分布，对目标分布的采样改为对近似分布的采样；
- 用于计算函数期望：直接就是 Importance sampling。

11.1.6 Sampling and the EM algorithm 小节则将 Monte Carlo EM，也就是基于采样的方法计算 EM 算法中的期望。

Markov Chain Monte Carlo

Markov chains

本小节先梳理下马尔可夫链的一些概念。

一阶马尔可夫链：A first-order Markov chain is defined to be a series of random variables $\mathrm{z}^{(1)}, \cdots, \mathrm{z}^{(M)}$ such that the following conditional independence property holds for $m \in \{1, \cdots, M-1\}$

$$p(z^{(m+1)}|\mathrm{z}^{(1)}, \cdots, \mathrm{z}^{(m)}) = p(z^{(m+1)}|\mathrm{z}^{(m)})$$

确定一条马尔可夫链：

1. initial distribution: $p(\mathrm{z}^{(0)})$
2. transition probabilities: $T_m(\mathrm{z}^{(m)}, \mathrm{z}^{(m+1)}) \equiv p(z^{(m+1)}|\mathrm{z}^{(m)})$

齐次 (homogeneous) 马尔可夫链：是指对每一步，转移分布的函数形式都相同 A Markov chain is called *homogeneous* if the transition probabilities are the same for all m.

马尔科夫链的静态分布 (invariant/stationary)：A distribution is said to be invariant, or stationary, with respect to a Markov chain if each step in the chain leaves that distribution invariant.

$$p^*(\mathrm{z}) = \sum_{\mathrm{z}'} T(\mathrm{z}', \mathrm{z}) p^*(\mathrm{z}')$$

马尔可夫链的静态分布不具有唯一性：Note that a given Markov chain may have more than one invariant distribution.

静态分布的一个充分条件是 detailed balance (细致平衡)：

$$p^*(\mathrm{z}) T(\mathrm{z}, \mathrm{z}') = p^*(\mathrm{z}') T(\mathrm{z}', \mathrm{z})$$

A Markov chain that respects detailed balance is said to be *reversible*. 下面介绍的各种 MCMC 算法都基于细致平衡构造转移概率。

> **Ergodicity (遍历性)** 的定义（摘自 Markov chain - Wikipedia）：
>
> A state *i* is said to be *ergodic* if it is aperiodic and positive recurrent. In other words, a state *i* is ergodic if it is recurrent, has a period of 1, and has finite mean recurrence time. If all states in an irreducible Markov chain are ergodic, then the chain is said to be ergodic. Some authors call any irreducible, positive recurrent Markov chains ergodic, even periodic ones.
>
> It can be shown that a finite state irreducible Markov chain is ergodic if it has an aperiodic state. More generally, a Markov chain is ergodic if there is a number *N* such that any state can be reached from any other state in any number of steps less or equal to a number *N*. In case of a fully connected transition matrix, where all transitions have a non-zero probability, this condition is fulfilled with *N* = 1.
>
> A Markov chain with more than one state and just one out-going transition per state is either not irreducible or not aperiodic, hence cannot be ergodic.

Ergodic Markov Chain 具有长时间尺度下的稳态行为，也就是说收敛，且收敛于唯一的静态分布，也称该静态分布称为平稳分布 (have only one *equilibrium* distribution).

It can be shown that a homogeneous Markov chain will be ergodic, subject only to weak restrictions on the invariant distribution and the transition probabilities. 也就是说 Markov Chain 的遍历性很容易满足。MCMC 就是构造平稳分布为目标分布的 Ergodic Markov Chain 以实现对目标分布的采样。

获得相互独立样本的方式： If we wish to obtain independent samples, then we can discard most of the sequence and just retain every M^{th} sample. For M sufficiently large, the retained samples will for all practical purposes be independent.

构造的 MCMC 要避免 random-walk behavior (也就是 MC 的各点相关性很强而不能很好地探索采样空间的现象。注意，这一现象不会随着 MC 达到平稳分布而消失：单对某一时刻的样本是服从平稳分布的，但相互之间仍存在相关性，我们通常会取一定间隔的两点作为相互独立的样本，但若存在较强的相关性，这一间隔就会很长，表现为 MC 大部分时间只在局部游走，采集独立样本的效率很低)，否则采样效率会很低。

Metropolis-Hastings algorithm & Gibbs sampling

> 关于 MH 和 Gibbs sampling 的具体推导强烈推荐靳志辉《LDA数学八卦》MCMC、Gibbs Sampling 部分，写得非常好，因此不再赘述。

11.2 小节开头介绍的 Metropolis algorithm 是 11.2.2 The Metropolis-Hastings algorithm 的特例，但我们又可基于前者推广得到后者（参见 《lda-LDA数学八卦》），这体现了特殊与一般的辩证关系。

注意，MH 算法中虽然存在"提议分布"和"拒绝"的概念，但并不是在进行拒绝采样。提议分布并非拒绝采样独有，而应理解为一个基准分布；此外，这里的"拒绝"也并不是像拒绝采样中那样丢弃掉样本，而是拒绝跳转到从提议分布中采到的 z'，但还是会跳转，只不过是原地跳转，即 $z^{\pi+1} = z^{\pi}$.

> This is in contrast to rejection sampling, where rejected samples are simply discarded. In the Metropolis algorithm when a candidate point is rejected, the previous sample is included instead in the final list of samples, leading to multiple copies of samples. Of course, in a practical implementation, only a single copy of each retained sample would be kept, along with an integer weighting factor recording how many times that state appears.

一方面，Gibbs sampling 可视为 MH 算法接受率为1的特殊情况；但另一方面，和 MH 算法一样，我们是直接基于细致平衡构造得到的 Gibbs sampling，而不是从 MH 推得的。此外，关于 Gibbs sampling，《LDA数学八卦》提到：

> 额外说明一下，我们看到教科书上的 Gibbs Sampling 算法大都是坐标轴轮换采样的，但是这其实是不强制要求的，可以在坐标轴轮换中引入随机性，这时候转移矩阵 Q 中任何两个点的转移概率中就会包含坐标轴选择的概率，而在通常的 Gibbs Sampling 算法中，坐标轴轮换是一个确定性的过程，也就是在给定时刻 t，在一根固定的坐标轴上转移的概率是1。

Slice sampling & Hamiltonian Monte Carlo

Slice sampling 和 Hamiltonian Monte Carlo 都属于 auxiliary variable methods：通过引入辅助变量升维，在高维空间中构造 MC 进行采样，因目标分布为高维分布的边缘分布，丢弃高维样本中新引入的维度（辅助变量的边缘分布是什么无所谓，因为会被丢弃），即得目标分布的样本。

在 MH 算法中，一方面我们希望状态转移时"步子"迈得尽量大，也就是前后两个状态的相关性尽量低，以通过少量的转移得到相互独立的样本，提高采样效率；但另一方面，步子迈得太大通常又会使拒绝率变高，转移总是在原地进行，有效转移变少，这使得总的转移次数又上去了，采样效率变低，由此陷入两难的境地。而 Slice sampling 和 Hamiltonian Monte Carlo 通过巧妙地构造马尔可夫链，在较好地探

索采样空间的同时，又保证了接受率 (being able to make large changes to the system state while keeping the rejection probability small)，避免了 random-walk behavior，具有较高的采样效率 。

> slice sampling 由 Neal 提出，原始论文：Slice sampling (projecteuclid.org)
>
> Slice sampling - Wikipedia

slice sampling：The method is based on the observation that to sample a random variable one can sample uniformly from the region under the graph of its density function.

Slice sampling, in its simplest form, samples uniformly from underneath the curve $f(x)$ without the need to reject any points, as follows:

1. Choose a starting value x_0 for which $f(x_0) > 0$.
2. Sample a y value uniformly between 0 and $f(x_0)$.
3. Draw a horizontal line across the curve at this y position.
4. Sample a point (x, y) from the line segments within the curve.
5. Repeat from step 2 using the new x value.

By using the x-position from the previous iteration of the algorithm, in the long run we select slices with probabilities proportional to the lengths of their segments within the curve. 上述做法实际上就是升维后再进行 Gibbs sampling，从而将问题转化为不断地对均匀分布进行采样。注意，不要将 Slice sampling 理解为如下的操作了：

- 在可行域内随机地选择 y，再在 y 对应的 slice 上采样，得到的 x 即为目标分布的样本；

上述做法虽然确实得到了目标分布的样本，也利用了 slice，但本质上是拒绝采样，效果等同于：

- 在可行域内随机地选择 x_0，再在 $0, \max_x(f(x))$ 之间采得 y，若 $y > f(x_0)$，则丢弃本次采得的 x_0，重复前面的步骤；否则，接受 x_0。

而 Slice sampling 是基于 MC 的。

另外，下面的做法就更离谱了，得到的 x 并不服从目标分布 (实际上只是做了 slice sampling 的一部分)：

- 在可行域内随机地选择 x_0，再在 $0, f(x_0)$ 间采样 y，接着在 y 对应的 slice 上采样 x 并结束.

上面讲了 slice sampling 最简单的情况，但目标分布很多时候并不是单峰的 (unimodal)，这使得第 4 步的采样很难进行。为此，Neal 的论文中介绍了第 4 步采用 stepping-out/doubling procedure + shrinkage procedure 的方案，并证明了修改后的第 4 步仍满足 detailed balance，因此平稳分布不变，采样依旧有效。

stepping-out/doubling procedure 以及对应 shrinkage procedure 的具体做法可参考原始论文，还是有很多细节的 (尤其是 doubling procedure 对应的方案，比 stepping-out 更复杂)。PRML 也只是简单提了下 stepping-out，光看那么小段内容是学不会的。网上文章介绍的做法基本上都不完整（说白了就是错的），完整做法还得看论文。

上面讨论的是单变量的情况，对多维分布，PRML: Slice sampling can be applied to multivariate distributions by repeatedly sampling each variable in turn, in the manner of Gibbs sampling. 但原始论文也给出了专门的方法: Rather than sample from a distribution for $x = (x_1, \cdots, x_n)$ by applying one of the single-variable slice sampling procedures described above to each x_i in turn,

we might try instead to apply the idea of slice sampling directly to the multivariate distribution. I will start by describing a straightforward generalization of the single-variable methods to multivariate distributions, and then describe a more sophisticated method, which can potentially allow for adaptation to the local dependencies between variables. 具体内容可参考原始论文。

Hamiltonian Monte Carlo 即 11.5 小节介绍的 Hybrid Monte Carlo，但前一说法更常用一点。

这里主要参考如下两篇论文

- MCMC using Hamiltonian dynamics (对，HMC slice sampling 的 Neal 大神的论文)
- [2108.12107] An Introduction to Hamiltonian Monte Carlo Method for Sampling (arxiv.org)

中的相关章节。网上也有推荐 [1701.02434] A Conceptual Introduction to Hamiltonian Monte Carlo (arxiv.org)，不过没看。

首先介绍物理背景。虽说是物理背景，但我们还是尽量少地引入物理概念（比如位移，速度等），因为过多的物理概念和类比反而还会对我们的理解造成干扰，所以我们这里尽量只介绍背后的数学：

1. The Hamiltonian of the particle, 也就是粒子（或者说系统）的总能量:

$$H(x,v)=f(x)+\frac{1}{2}\|v\|^2$$

2. The Hamiltonian dynamics for this particle are, 也就是 Hamiltonian 动力学要求如下的关系成立:

$$\begin{cases}\dfrac{\mathrm{d}x}{\mathrm{d}t}=\dfrac{\partial H}{\partial v}\\ \dfrac{\mathrm{d}v}{\mathrm{d}t}=-\dfrac{\partial H}{\partial x}\end{cases}$$

   其中，t 为时间。代入前面 H 的表达式，即

$$\begin{cases}\dfrac{\mathrm{d}x}{\mathrm{d}t} &= v\\ \dfrac{\mathrm{d}v}{\mathrm{d}t} &= -\nabla f(x)\end{cases} \tag{1}$$

基于上述内容，我们可推得如下性质：

1. 随着时间推移，x, v 会发生变化，但 H 保持不变；
2. Hamiltonian dynamics conserves the volume in the “phase space“ (即 x, v 组成的空间)。

下面开始梳理 Hamiltonian Monte Carlo 的思路：

> We will use x to represent the variables of interest, and introduce v just to allow Hamiltonian dynamics to operate.
>
> Using Hamiltonian dynamics to sample from a distribution requires translating the density function for this distribution to a potential energy function and introducing “momentum” variables to go with the original variables of interest (now seen as “position” variables). We can then simulate a Markov chain in which each iteration resamples the momentum and then does a Metropolis update with a proposal found using Hamiltonian dynamics.

假设我们要采集 $x \sim q(x)$ 的样本。首先，我们引入和 x 同维的变量 v，组成随机变量 (x, v)，其联合概率分布构造如下：

$$p(x, v) = \frac{1}{Z}\exp\left(-H(x, v)\right)$$

$$H(x, v) = -\log q(x) + \frac{1}{2}\|v\|^2$$

可以看到，x 的边缘分布为 $q(x)$，v 的边缘分布为高斯分布，且 x, v 相互独立。接下来，我们采用 MCMC 对联合概率分布采样，得到样本后丢弃掉 v 的部分即可。为此，我们需要知道转态转移是怎么做的。

HMC 的转态转移由 Hamiltonian dynamics，即 (1) 给出。不妨记 t 时间后状态由 (x_0, v_0) 变为 (x_t, v_t)，其中 t 是人为设定的。而为了保证遍历性，我们会对 v 的边缘分布（或条件分布，因为 x, v 相互独立，所以边缘分布和条件分布等价）采样，不妨记为 v_t'，然后替换掉 v_t，再基于 (x_t, v_t') 重复前面的转移。$(x_t, v_t) \to (x_t, v_t')$ 这一操作和 $(x_0, v_0) \to (x_t, v_t)$ 一样，均使联合分布 $p(x, v)$ 是其静态分布，因此，这两个操作合起来也会使 $p(x, v)$ 是静态分布。由此，转移可以 rearrange 为如下的过程：

1. 采样 $v_0' \sim p(v)$，替换 v_0，组成 (x_0, v_0')，即 $(x_0, v_0) \to (x_0, v_0')$
2. 基于 Hamiltonian dynamics，得到 t 时间后的状态，即 $(x_0, v_0') \to (x_t, v_t')$；再视 (x_t, v_t') 为 (x_0, v_0) 重复上述步骤。

> Each iteration of the HMC algorithm has two steps. The first changes only the momentum; the second may change both position and momentum. Both steps leave the canonical joint distribution of (x, v) invariant, and hence their combination also leaves this distribution invariant.
>
> In the first step, new values for the momentum variables are randomly drawn from their Gaussian distribution, independently of the current values of the position variables. Since x isn't changed, and v is drawn from it's correct conditional distribution given x (the same as its marginal distribution, due to independence), this step obviously leaves the canonical joint distribution invariant.
>
> In the second step, a Metropolis update is performed, using Hamiltonian dynamics to propose a new state.

关于遍历性，还需要说明的是，可能存在周期性的情况，使得对某个 t，$(x_t, v_t) \equiv (x_0, v_0)$。此时，遍历性是不满足的，解决方法是每次的 t 随机选择即可。

> This potential problem of non-ergodicity can be solved by randomly choosing t from some fairly small interval.

最后，$(x_0, v_0) \to (x_t, v_t)$ 是根据 Hamiltonian dynamics 得到的，理论上 $H(x_0, v_0) = H(x_t, v_t)$。但 (x_t, v_t) 的计算实际上是一个积分问题，会采用数值积分去计算（为了表述方便，记数值计算得到的值为 (x_t^*, v_t^*)），比如 leapfrog 法。而数值积分会引入误差，为了使联合概率分布仍然 invariant，需要增加接受概率的步骤，即从 $(x_0, v_0) \to (x_t, v_t)$ 的过程变为，以 $\min(1, \exp\{H(x_0, v_0) - H(x_t^*, v_t^*)\})$ 的概率接受转移到 (x_t^*, v_t^*)，否则，原地转移。

> If the proposed state is not accepted (ie, it is rejected), the next state is the same as the current state.

其他参考文献

写得不错的，比较宏观的，捋思路的知乎：

马尔可夫链蒙特卡洛算法 (一) MH - 知乎 (zhihu.com)

马尔可夫链蒙特卡洛算法 (二) HMC - 知乎 (zhihu.com)

马尔可夫链蒙特卡洛算法 (三) slice sampling - 知乎 (zhihu.com)

马尔可夫链蒙特卡罗算法（MCMC） - 知乎 (zhihu.com)：非常好，里面也参考了刘建平的博客。

HMC:哈密顿蒙特卡洛方法 - 丶工藤 - 博客园 (cnblogs.com)
