Machine Learning Notes **Date: April 20, 2019**

Markov chain Monte-Carlo Methods

Author: 申恒恒 *Scribe: ShenHengheng*

1 蒙特卡罗方法基础

由于蒙特卡罗方法自身带有随机的色彩，因此在介绍蒙特卡罗方法之前有必要介绍其依赖的概率理论基础，而蒙特卡罗方法又是一种工程广泛应用的采样方法，所以本章会重点介绍采样技术，采样技术又依赖于概率分布以及变分推断等知识.

1.1 概率基础：点估计方法

点估计方法是机器学习从业者都用到的方法，点估计是指用样本统计量估计总体参数的一种统计方法，由于样本统计量可以表现为数轴的一个点值，估计的结果也常为一个数值，因此称为点估计，比如在机器学习中，$\arg\max_\theta(f_\theta(X))$ 就可以使用点估计的方法来计算. 点估计的常用方法有矩估计法、顺序统计量法、最大似然法、最小二乘法等.

为了估计一个分布的参数，我们可以使用最大似然估计 (Maximum Likelihood estimation)、最大后验估计 (Maximum A Posterior).

在某些情况下我们并不能求解 $\arg\max_\theta(f_\theta(X))$，这时我们可以通过一些迭代的方法，比如最大期望估计方法（EM）算法进行求解.

1.1.1 最大似然估计

下面我通过一个例子介绍最大似然估计：假设我们的数据是服从正态分布 D 的, 且正态分布的参数已知为 θ, 且概率密度函数为 $\mathcal{N}_D$，这时正态分布的参数 $\theta=(\mu,\sigma)$ 应该如何估计呢?

我们独立同步分布地在总体中进行抽样，即 $x_i \overset{i.i.d}{\sim} X$，抽样的数据为 $(x_1,x_2,...,x_N)$, 已知似然函数为

$$\begin{aligned} L(\theta|x_1,x_2,...,x_N) &= \mathcal{N}_\theta(x_1,x_2,...,x_N) \\ &= \prod_{i=1}^{N}\mathcal{N}(x_i|\theta) \end{aligned}$$

这时我们可以通过最大似然估计方法可以计算出：

$$\begin{aligned}\theta^{\mathrm{MLE}} &= \arg\max_{\theta}(\log[p(X|\theta)]) \\ &= \arg\max_{\theta}(\sum_{i=1}^{N}\log[\mathcal{N}(x_i;\mu,\sigma)])\end{aligned}$$

接下来我们就可以通过对上面的式子求导求得 θ.

注意：

最大化一个似然函数同最大化它的自然对数是等价的. 因为自然对数 log 是一个连续且在似然函数的值域内严格递增的上凸函数. 求对数通常能够一定程度上简化运算，比如在这个例子中可以看到：

$$\begin{aligned}0 &= \frac{\partial}{\partial\mu}\log\left(\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} e^{-\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2+n(\bar{x}-\mu)^2}{2\sigma^2}}\right) \\ &= \frac{\partial}{\partial\mu}\left(\log\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} - \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2+n(\bar{x}-\mu)^2}{2\sigma^2}\right) \\ &= 0 - \frac{-2n(\bar{x}-\mu)}{2\sigma^2}\end{aligned}$$

这个方程的解是 $\widehat{\mu}=\bar{x}=\sum_{i=1}^{n}x_i/n$ 这的确是这个函数的最大值，因为它是 μ 里头惟一的一阶导数等于零的点并且二阶导数严格小于零.

同理，我们对 σ 求导，并使其为零.

$$\begin{aligned}0 &= \frac{\partial}{\partial\sigma}\log\left(\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} e^{-\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2+n(\bar{x}-\mu)^2}{2\sigma^2}}\right) \\ &= \frac{\partial}{\partial\sigma}\left(\frac{n}{2}\log\left(\frac{1}{2\pi\sigma^2}\right) - \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2+n(\bar{x}-\mu)^2}{2\sigma^2}\right) \\ &= -\frac{n}{\sigma} + \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2+n(\bar{x}-\mu)^2}{\sigma^3}\end{aligned}$$

这个方程的解是 $\widehat{\sigma}^2=\sum_{i=1}^{n}(x_i-\widehat{\mu})^2/n$

因此，其关于 $\theta=(\mu,\sigma^2)$ 的最大似然估计为：$\widehat{\theta}=(\widehat{\mu},\widehat{\sigma}^2)=(\bar{x},\sum_{i=1}^{n}(x_i-\bar{x})^2/n)$.

1.1.2 最大后验估计

MAP 与 MLE 有密切的联系，最大后验概率进一步考虑了被估计参数的先验分布，所以最大后验概率估计可以看作是规则化的最大似然估计. 比如在正态分布的例子中 $\mu\sim\mathcal{N}(\mu_0,\sigma_0)$, 这种估计被称为最大后验估计. 此时的优化目标为

$$\theta^{\mathrm{MLE}} = \arg\max_{\theta}(\log[p(X|\theta)p(\theta)])$$

此时，需要估计出均值:

$$\mu^{\mathrm{MAP}} = \arg\max_{\theta}(\sum_{i=1}^{N}\log[\mathcal{N}(x_i|\mu,\sigma)\mathcal{N}(\mu;\mu_0,\sigma_0)])$$

那么如何计算 μ 呢? 对 μ 求导让其等 0, 即可求得 μ.

特殊地，针对正态分布的结果为:

$$\mu^{\mathrm{MAP}} = \frac{n\sigma^2}{n\sigma^2+\sigma^2}\left(\frac{1}{n}\sum_{j=1}^{n}x_i\right) + \frac{\sigma^2}{n\sigma^2+\sigma^2}\mu_0 \tag{1}$$

通过公式 1我们可以发现，当 $n\to\infty$ 时，$\mu^{\mathrm{MAP}} \to \frac{1}{n}\sum_{j=1}^{n}x_i$

1.1.3 EM 算法

前面也提到了如果我们不能找到 $\arg\max_{\theta}(\log[p(X|\theta)p(\theta)])$, 这时我们可以利用数值方法 EM 算法（如果将 EM 算法应用到 GMM 上，其实它也是一个点估计方法）进行求解.

给定一个初始的参数 θ_0，我们可以得到以下参数 $\{\theta_0,\theta_1,\theta_2,...,\theta_g,\theta_{g+1},...\}$, 其中他们满足以下条件:

$$\log[p(X|\theta^{(g+1)})p(\theta^{(g+1)})] \geq \log[p(X|\theta^{(g)})p(\theta^{(g)})]$$

1.1.4 哲学

在机器学习问题中，我们往往寻求在贝叶斯的框架下，我们的先验与后验有如下关系:

$$p(\theta|\mathrm{Data}) \propto p(\mathrm{data}|\theta)p(\theta)$$

其中 $p(\mathrm{data}|\theta)$ 为参数的似然函数，$p(\theta)$ 为参数的先验分布，$p(\theta|\mathrm{Data})$ 表示后验概率。另外公式表明了 $p(\theta)$ 和 $p(\theta|\mathrm{Data})$ 是共轭的，即先验和后验服从同一种分布。但是实际情况下往往是不满足的，即我们的先验与似然函数并不能产生与先验同一种分布，此时后验分布往往比较复杂，这时我们需要其他的方法进行求解我们的后验分布，最重要的方法就是使用蒙特卡罗的方法进行求解。

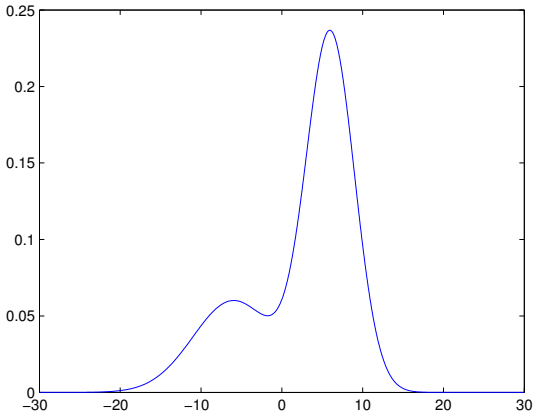


图 1: 这是我们的后验分布 $p(\theta|X)$, 我们该如何求解该分布的参数，其中最主要的方法就是蒙特卡罗方法，就是我们在该分布下进行取样。

1.2 采样方法

蒙特卡罗方法是以数据采样为基础的，在计算机模拟中有很重要的作用，这一节主要以采样方法为目标，重点阐述不同采样方法的特点、之间的联系和区别，主要涉及以下采样方法：

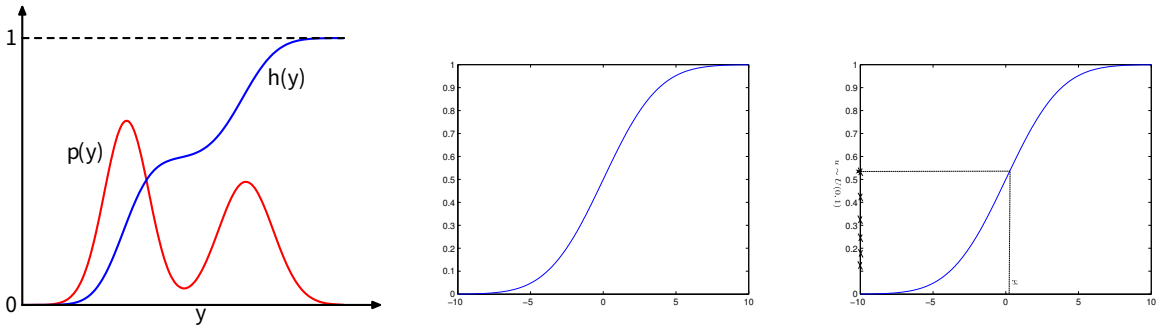
- 累计分布函数的反函数方法
- 拒绝采样
- 自设应拒绝采样
- 蒙特卡罗方法
- 重要采样

1.2.1 累计分布函数的反函数方法

累计分布函数的反函数方法又被称直接从分布中取样, 主要过程如下 (参考图 2(a)):

- 计算出累计分布函数: $h(y) = \int_{-\infty}^{y} p(y')dy'$
- 在 $[0,1]$ 进行均匀采样: $u \sim \text{Uniform}[0,1]$
- 通过 cdf 反函数计算出样本值: $y(u) = h^{-1}(u)$

但是我们往往求不出来累计分布函数的反函数, 那么应该怎么做呢? ↓↓



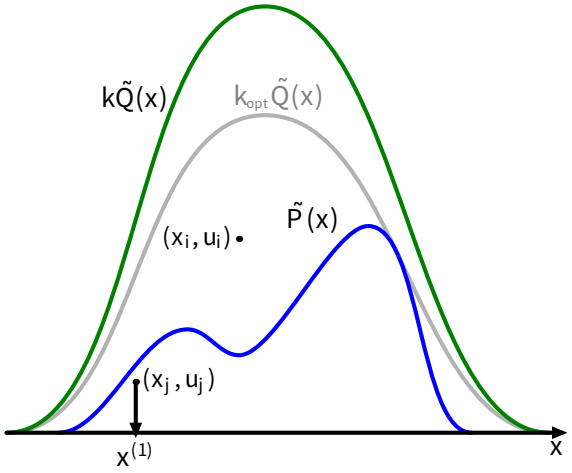
(a) 从分布中直接采样　(b) CDF　(c) Sampling

图 2: CDF　$u \sim U(0,1)$,　$x = CDF^{-1}(u)$

1.2.2 拒绝采样

拒绝采样是用于从分布生成观测值的基本技术。它通常也被称为接受拒绝方法，并且是一种蒙特卡罗方法。该方法适用于具有密度 $\mathbb{R}^m$ 中的任何分布。拒绝抽样是基于以下观察：对随机变量进行抽样，可以从其密度函数图下的区域均匀地采样.

拒绝采样方法是一种针对复杂、高维度数据的随机采样方法，拒绝采样方法主要借助于一个简单的参考分布 (proposal distribution), 记为 $\tilde{Q}(x)$，该分布的采样易于实现，如均匀分布、高斯分布。然后引入常数 k，使得对所有的的 x，满足 $k\tilde{Q}(x) \geq \tilde{P}(x)$，如图 3(a)所示，蓝色的曲线为 $\tilde{P}(x)$，绿色的曲线为 $k\tilde{Q}(x)$。



(a) 拒绝采样

1: $i \leftarrow 0$
2: **while** $i \neq N$ **do**
3: $\quad x^{(i)} \sim \tilde{Q}(x)$
4: $\quad u \sim U(0,1)$
5: $\quad$ **if** $u < \frac{\tilde{P}(x^{(i)})}{k\tilde{Q}(x^{(i)})}$ **then**
6: $\quad\quad$ accept $x^{(i)}$
7: $\quad\quad i \leftarrow i+1$
8: $\quad$ **else**
9: $\quad\quad$ reject $x^{(i)}$
10: $\quad$ **end if**
11: **end while**

(b) Algorithm

图 3: 拒绝采样, 其中 $\tilde{Q}(x)$ 是一个已知简单的概率分布

在每次采样中，首先从 $\tilde{Q}(x)$ 采样一个数值 x_0，然后在区间 $[0, \tilde{Q}(x_0)]$ 进行均匀采样，得到 u_0。如

果 $u_0 < \tilde{P}(x0)$，则保留该采样值，否则舍弃该采样值。最后得到的数据就是对该分布的一个近似采样。

每次采样的接受概率计算如下：

$$p(accept) = \int \frac{\tilde{P}(x)}{kQ(z)} Q(z)dz = \frac{1}{k} \int \tilde{P}(x)dz$$

所以，为了提高接受概率，防止舍弃过多的采样值而导致采样效率低下，k 的选取应该在满足 $k\tilde{Q}(z) \geq \tilde{P}(x)$ 的基础上尽可能小。

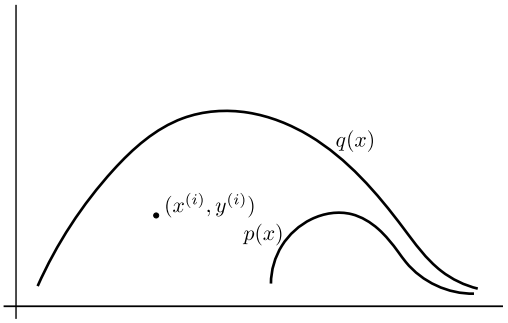


图 4: 效率低的参考分布

如果被采样的函数高度集中在某个区域，例如在某个位置具有尖峰的函数，则拒绝采样可能导致大量不需要的样本被采集。对于许多分布，可以使用自适应拒绝采样方法来解决该问题（参见 1.2.3）。此外，随着问题的尺寸变大，嵌入体积与嵌入体积“角”的比率趋向于零，因此在生成有用样本之前可能发生大量拒绝，可能导致算法效率低下且不切实际。在高维度上，有必要使用不同的方法，通常是马尔可夫链蒙特卡罗（MCMC）方法，例如 Metropolis 采样或 Gibbs 采样。（然而，Gibbs 采样将多维采样问题分解为一系列低维采样，可以使用拒绝采样作为其中一个步骤。）

1.2.3 自适应拒绝采样方法

拒绝采样方法需要找到合适的参考分布，而且对于很多分布函数来说，这样的参考分布是很难找到的，往往找到的参考分布的效率低下，这也是拒绝采样方法无法在工程中广泛使用的原因。而在算法和工程中会大量使用到自适应拒绝采样。自适应扩展使得采样算法效率更高。

该方法仅适用于我们概率密度函数的 log 是凹函数 (log concave densities), 算法的基本思想是自适应的形成一个上限（$p(x)$ 的上限），并在拒绝采样中使用它代替 $k\tilde{Q}(x)$。

如图 5所示，考虑了对数密度 $\log p(x)$。然后从上边界（upper envelope）采样 $x^{(i)}$，如拒绝采样接受或拒绝（即 $y^{(i)} \leq \log(p(x))$ 或 $y^{(i)} > \log(p(x))$）。如果它被拒绝，则绘制切线，通过 $x = x^{(i)}$ 和 $y = \log(p)$ 从而可以减少被拒绝的样本数。这些切平面的交叉使得能够自适应地形成边界。要从上边界进行采样，我们需要通过取指数和使用指数分布的属性从 log 空间转换到我们 $p(x)$ 空间。

该方法的问题在于只适合我们的概率密度函数为 log-concave 的，并且我们要找的是超平面之间的

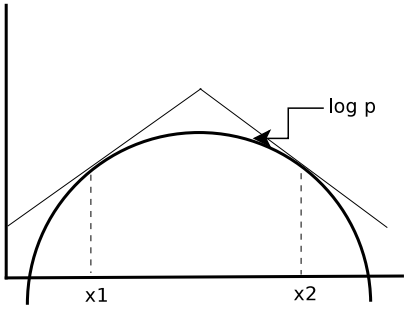


图 5: Adapter Reject sampling

交叉，因此该方法并不适用于高维数据的采样。

1.2.4 重要采样

给出函数的期望 $E[f(x)]$ 为

$$E_{p(x)}[f(x)] = \int_x f(x)p(x)dx$$

那么我们可以独立同分布在总体样本中进行采样，即 $x^{(i)} \sim p(x)$, $i = 1,2,...,N$，这样我们可以估计出函数的期望为

$$E_{p(x)}[f(x)] = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})dx$$

但是问题在于不能采样。加入我们有一个容易采样的分布，（比如均匀分布，正态分布等等），这时我们采样 $x^{(i)} \sim q(x)$, 这时可以定义重要性权重（importance weight）为

$$w(x^{(i)}) = \frac{p(x^{(i)})}{q(x^{(i)})}$$

考虑加权后的蒙特卡罗和：

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})w(x^{(i)}) &= \frac{1}{N}\sum_{i=1}^{N} f\left(x^{(i)}\frac{p(x^{(i)})}{q(x^{(i)})}\right) \\
&\xrightarrow{a.s} \int \left(f(x)\frac{p(x)}{q(x)}\right)q(x)dx \quad \text{(Law of Large Numbers)} \\
&= \int f(x)p(x)dx
\end{aligned}$$

原理上讲，我们可以从任意分布 $q(x)$ 中进行采样, 但是在实际中，我们倾向于选择 $q(x)$ 尽可能与 $|f(x)|w(x)$ 相近进而可以减少我们的估计误差。

Remark 1. 我们不需要关心 $p(x)$ 和 $q(x)$ 的正规化系数, 因为 w 是 $\frac{p}{q}$, 我们可以计算下面的公式

$$\int f(x)p(x)dx \approx \frac{\sum_{i=1}^{N} f(x^{(i)})w(x^{(i)})}{\sum_{i=1}^{N} w(x^{(i)}))} \tag{2}$$

重要采样与拒绝采样的区别在于没有样本会拒绝掉。

1.2.5 Metropolis-Hastings

在 Metropolis-Hastings 的采样中，样本大多朝着较高密度区域移动，但有时也会向低密度区域移动。与拒绝抽样，我们总是丢弃被拒绝的样本相比，这里我们有时也保留这些样本。它的伪代码如下：

1: Init $x^{(0)}$
2: **for** $i = 0$ to $N-1$ **do**
3: $u \sim U(0,1)$
4: $x^* \sim q(x^*|x^{(i)})$
5: **if** $u < \min\left\{1, \frac{p(x^*)q(x^{(i)}|x^*)}{p(x^{(i)})q(x^*|x^{(i)})}\right\}$ **then**
6: $x^{(i+1)} \leftarrow x^*$
7: **else**
8: $x^{(i+1)} \leftarrow x^{(i)}$
9: **end if**
10: **end for**

Remark 2. In line 5 of the algorithm, if q is symmetric then to the original Metropolis algorithm by Hastings. $\frac{p(x^*)q(x^{(i)}|x^*)}{p(x^{(i)})q(x^*|x^{(i)})} = 1$. This term was later introduced

1.2.6 Gibbs Sampling

Let $x = (x_1, x_2, ..., x_p)$. In order to obtain samples $x^{(i)}$ from the joint distribution $P(x)$ do the following:

- Initialize $x^{(0)}$ and let $i = 0$.

- Repeatedly:

$$
\begin{aligned}
&\text{Sample} \quad x_1^{(i+1)} \sim P(x_1|x_2^{(i)}, x_3^{(i)}, x_4^{(i)}, ..., x_p^{(i)}) \\
&\text{Sample} \quad x_2^{(i+1)} \sim P(x_2|x_1^{(i+1)}, x_3^{(i)}, x_4^{(i)}, ..., x_p^{(i)}) \\
&\quad \vdots \\
&\text{Sample} \quad x_p^{(i+1)} \sim P(x_p|x_1^{(i+1)}, x_2^{(i+1)}, x_3^{(i+1)}, ..., x_{p-1}^{(i+1)}) \\
&\text{Set} \quad i = i+1
\end{aligned}
$$

It is possible to do this block-wise, i.e. sample blocks of the x_i together. Various approaches exist (and can be justified) to ordering the variables in the sampling loop. One approach is random sweeps: variables are chosen at random to resample.

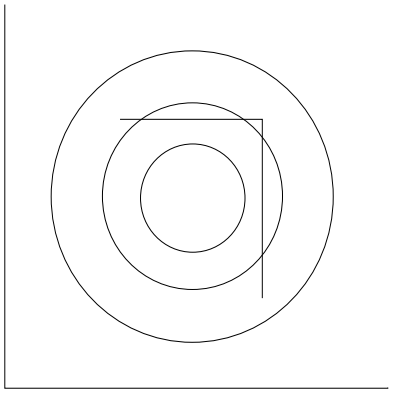


图 6: x_1, x_2 actually independent. Gibbs sampler makes big jumps. This is desirable.

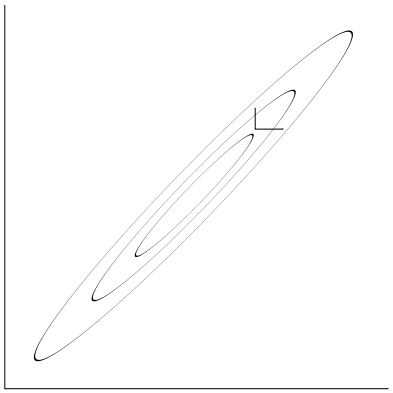


图 7: x_1, x_2 highly correlated. Gibbs sampler makes only small moves. This is called chattering and is undesirable.

Example 3 (Gibbs Sampling)**.**

$$\begin{aligned} y_{ij} &\sim \mathcal{N}(\theta_j, \sigma^2) \\ \theta_j &\sim \mathcal{N}(\mu, \tau^2) \\ (\mu, \sigma, \tau) &\frac{1}{\sigma}) \end{aligned}$$

We want to sample all of $(\theta_1, ..., \theta_J, \mu, \sigma, \tau | y)$. Here's the Gibbs sampler:

$$\begin{aligned} \theta_j | \mu, \sigma^2, \tau^2, y &\sim \mathcal{N}\left(\frac{\frac{1}{\tau^2 \mu + \frac{n_j}{\sigma^2 \bar{y}_{ij}}}}{\frac{1}{\tau^2} + \frac{n}{\sigma^2}, \frac{1}{\frac{1}{\tau^2} + \frac{n_j}{\sigma^2}}} \right) \\ \mu | \theta_1, ..., \theta_J, \sigma^2, \tau^2, y &\sim \mathcal{N}\left(\frac{1}{J} \sum_j \theta_j, \frac{\tau^2}{J} \right) \\ \tau^2 | \theta, \mu &\sim \text{IG}\left(\frac{J-1}{2}, \frac{\sum_j (\theta_j - u)^2}{2} \right) \\ \sigma^2 | \theta, y &\sim \text{IG}\left(\frac{n}{2}, \frac{\sum_i \sum_j (y_{ij} - \theta_j)^2}{2} \right) \end{aligned}$$