

# 采样方法

## 采样方法

无意识统计学家法则(LOTUS)

蒙特卡洛数值积分

Monte Carlo principle

生成一个概率分布的样本

逆变换采样

接受拒绝采样

重要性采样

采样在DL中的应用

    重要性采样

    噪声对比估计

    负采样

马尔可夫过程

    马尔可夫链

    状态转移概率

    状态转移矩阵的性质

    马尔科夫链收敛定理

    基于马尔科夫链采样

MCMC采样

    细致平稳条件

M-H采样

Gibbs Sampling

    重新寻找合适的细致平稳条件

    二维Gibbs采样

    多维Gibbs采样

    二维Gibbs采样实例

    Gibbs采样小结

采样方法总结

    逆变换采样

    接受拒绝采样

    重要性采样

    基于马尔科夫链采样

    MCMC采样

    M-H采样

    Gibbs采样

## 无意识统计学家法则(LOTUS)

已知随机变量 $X$ 的概率密度为 $p(x)$ ,  $g(X)$ 为 $X$ 的函数, 则:

$$E(g(X)) = \int_a^b g(x)p_X(x)dx \quad (1)$$

## 蒙特卡洛数值积分

对于一个连续函数 $f$ , 要计算的积分有如下形式:

$$F = \int_a^b f(x)dx \quad (2)$$

转换:

$$F = \int_a^b \frac{f(x)}{q(x)} q(x)dx \quad (3)$$

根据 $q(x)$ 采样 $X_i$ ,  $f$ 的蒙特卡洛积分公式为:

$$F^N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{q(X_i)} \quad (4)$$

这公式的作用相当于在对 $f(x)$ 做积分, 只不过不那么“精确”, 即蒙特卡罗积分是对理想积分的近似。

那么这个近似是如何完成的? 很简单, 核心就是两个字: **采样(Sampling)**。对一个连续函数的采样方法是在该函数的定义域中随机挑 $N$ 个值, 并求出对应的 $N$ 个 $f(X_i)$ , 就得到了样本集合。再对这些样本集合做一些换算, 就可以得到一个近似的积分了。对于蒙特卡罗积分, 采样样本越多, 就越逼近真实的积分结果, 这是蒙特卡罗积分的最核心特性。

证明:

$$\begin{aligned} \mathbb{E}[F^N] &= \frac{1}{N} \sum_{i=1}^N \mathbb{E}\left[\frac{f(X_i)}{q(X_i)}\right] \\ &= \frac{1}{N} \sum_{i=0}^{N-1} \int_{\omega} \frac{f(x)}{q(x)} q(x)dx \\ &= F \end{aligned} \quad (5)$$

实际应用中,  $x$ 可以选择均匀分布,  $q(x) = \frac{1}{m}$ , 随机采样得到 $\frac{f(x_i)}{q(x_i)}$ , 采样很多次后, 求期望。

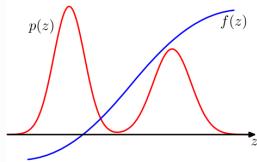
这样把 $q(x)$ 看做是 $x$ 在区间内的概率分布, 而把前面的分数部份看做一个函数, 然后在 $q(x)$ 下抽取 $n$ 个样本, 当 $n$ 足够大时, 可以用采用均值来近似:

因此只要 $q(x)$ 比较容易采到数据样本就行了。随机模拟方法的核心就是如何对一个概率分布得到样本, 即抽样(sampling)。

## Monte Carlo principle

Monte Carlo 抽样计算随即变量的期望值是接下来内容的重点:  $X$  表示随即变量, 服从概率分布 $p(x)$ , 那么要计算 $f(x)$ 的期望, 只需要我们不停从 $p(x)$ 中抽样 $x_i$ , 然后对这些 $f(x_i)$ 取平均即可近似 $f(x)$ 的期望。

$$E_N(f) = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)} \quad (6)$$



## 生成一个概率分布的样本

而我们常见的概率分布，无论是连续的还是离散的分布，都可以基于 $Uniform(0, 1)$ 的样本生成。例如正态分布可以通过著名的Box-Muller变换得到。如果随机变量 $U_1, U_2$ 独立且 $U_1, U_2 \sim Uniform[0, 1]$ ，则

$$\begin{aligned} Z_0 &= \sqrt{-2 \ln U_1} \cos(2\pi U_2) \\ Z_1 &= \sqrt{-2 \ln U_1} \sin(2\pi U_2) \end{aligned} \quad (7)$$

则 $Z_0, Z_1$ 独立且服从标准正态分布。

## 逆变换采样

比较简单的一种情况是，我们可以通过PDF与CDF之间的关系，求出相应的CDF。或者我们根本就不知道PDF，但是知道CDF。此时就可以使用Inverse CDF的方法来进行采样。这种方法又称为逆变换采样（Inverse transform sampling）。

所以，通常通过对PDF进行积分来得到概率分布的CDF。然后我们再得到CDF的反函数，如果你想得到 $n$ 个观察值，则重复下面的步骤 $n$ 次：

- 从 $Uniform(0,1)$ 中随机生成一个值（前面已经说过，计算机可以实现从均匀分布中采样），用 $U$ 表示。
- 计算 $x = CDF^{-1}(U)$ 的值，则 $x$ 就是从PDF中得出的一个采样点。

举个具体例子吧，例如我想按照标准正态分布 $N(0, 1)$ 取10个随机数，那么我首先在 $(0, 1)$ 上按照均匀分布取10个点

0.4505 0.0838 0.2290 0.9133 0.1524 0.8258 0.5383 0.9961 0.0782 0.4427

然后，我去找这些值在CDF上对应的 $x$ ，如下

-0.1243 -1.3798 -0.7422 1.3616 -1.0263 0.9378 0.0963 2.6636 -1.4175 -0.1442

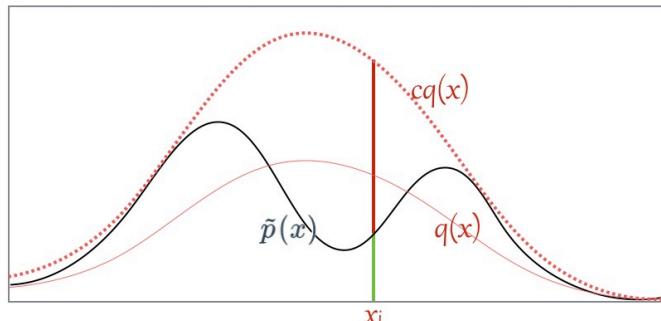
那么上述这些点，就是我按照正态分布取得的10个随机数。

缺点：

- 有时CDF不好求
- CDF的反函数不好求

## 接受拒绝采样

很多实际问题中， $p(x)$ 是很难直接采样的，因此，我们需要求助其他的手段来采样。既然 $p(x)$ 太复杂在程序中没法直接采样，那么我设定一个程序可抽样的分布 $q(x)$ 比如高斯分布，然后按照一定的方法拒绝某些样本，达到接近 $p(x)$ 分布的目的。其中 $q(x)$ 叫做proposal distribution。



具体操作如下，设定一个方便抽样的函数  $q(x)$ ，以及一个常量  $c$ ，使得  $p(x)$  总在  $cq(x)$  的下方。

- $x$  轴方向：从  $q(x)$  分布抽样得到  $x(i)$ ；
- $y$  轴方向：对  $x(i)$  计算接受概率： $\alpha = \frac{p(x_i)}{cq(x_i)}$ ；
- 从均匀分布  $(0, 1)$  中抽样得到  $u$ ；
- 如果： $\alpha \geq u$ ，则接受  $x(i)$  作为  $p(x)$  的抽样；否则，拒绝，重复以上过程

最终得到  $n$  个接受的样本  $x_0, x_1, \dots, x_n$ ，则最后的蒙特卡洛求解结果为： $\frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{q(x_i)}$ 。它的原理从直观上来解释也是相当容易理解的。在上图的例子中，从哪些位置抽出的点会比较容易被接受？显然，红色曲线和绿色曲线所示之函数更加接近的地方接受概率较高，也即是更容易被接受，所以在这样的地方采到的点就会比较多，而在接受概率较低（即两个函数差距较大）的地方采到的点就会比较少，这也就保证了这个方法的有效性。

在高维的情况下，**Rejection Sampling** 会出现两个问题：

- 合适的  $q$  分布比较难以找到，
- 很难确定一个合理的  $c$  值。

这两个问题会导致拒绝率很高，无用计算增加。

## 重要性采样

$f(x)$  为  $x$  的函数， $p(x)$  为  $x$  的 PDF，问题为求下式的积分：

$$E[f(x)] = \int_X f(x)p(x)dx \quad (8)$$

按照蒙特卡洛求定积分的方法，我们将从满足  $p(x)$  的概率分布中独立地采样出一系列随机变量  $x_i$ ，然后便有

$$E[f(x)] \approx \frac{1}{N} \sum_{i=1}^N f(x_i) \quad (9)$$

但是现在的困难是对满足  $p(x)$  的概率分布进行采样非常困难，毕竟实际中很多  $p(x)$  的形式都相当复杂。这时我们该怎么做呢？于是想到做等量变换，将其转化为：

$$\int_X f(x)p(x)dx = \int_X f(x) \frac{p(x)}{q(x)} q(x)dx = \int_X f(x)w(x)q(x)dx \quad (10)$$

其中  $w(x) = \frac{p(x)}{q(x)}$ ，称其为重要性权重。那么，根据  $q(x)$  采样  $x_i$ （此过程可以使用逆变换采样），那么蒙特卡洛估计就是：

$$I_N(f) = \frac{1}{N} \sum_{i=1}^N f(x_i)w(x_i) \quad (11)$$

我们来考察一下上面的式子， $p(x)$  和  $f(x)$  是确定的，我们要确定的是  $q(x)$ 。要确定一个什么样的分布才会让采样的效果比较好呢？直观的感觉是，样本的方差越小期望收敛速率越快。举个简单的例子比如一次采样是 0，一次采样是 1000，平均值是 500，这样采样效果很差，如果一次采样是 499，一次采样是 501，你说期望是 500，可信度还比较高。因此，我们很有必要研究相应的蒙特卡洛的方差：

$$var_{q(x)}(f(x)w(x)) = \mathbb{E}_{q(x)}(f^2(x)w^2(x)) - I^2(f) \quad (12)$$

上式中第二项不依赖于  $q(x)$ ，因此我们只需要最小化第一项就可以。那么根据 Jensen 不等式可知具有下界即：

$$\mathbb{E}_{q(x)}(f^2(x)w^2(x)) \geq (\mathbb{E}_{q(x)}(|f(x)|w(x)))^2 = (\int |f(x)|p(x)dx)^2 \quad (13)$$

那么下界达到即等号成立当且仅当

$$q^*(x) = \frac{|f(x)p(x)|}{\int |f(x)p(x)|dx} \quad (14)$$

尽管在实际中，上式很难拿取到，但是他告我们一个真理就是，当我们取样 $p(x)$ 时候，应该是取 $|f(x)|p(x)$ 相当大值，这样才有高效率的采样。这表明重要性采样有可能比用原来的 $p(x)$ 分布抽样更加有效。

## 采样在DL中的应用

### 重要性采样

语言模型：在训练阶段，我们的目标是使得训练集中每个词语 $w$ 的交叉熵最小，也就是使得softmax层输出值的负对数取值最小。模型的损失函数可以写成：

$$J_\theta = -\log \frac{\exp(h^T v'_w)}{\sum_{w_i \in V} \exp(h^T v'_{w_i})} \quad (15)$$

为了便于推导，我们将 $J_\theta$ 改写为：

$$J_\theta = -h^T v'_w + \log \sum_{w_i \in V} \exp(h^T v'_{w_i}) \quad (16)$$

令 $-\mathcal{E}(w)$ 代替 $h^T v'_w$ ，于是得到等式：

$$J_\theta = \mathcal{E}(w) + \log \sum_{w_i \in V} \exp(-\mathcal{E}(w_i)) \quad (17)$$

在反向传播阶段，我们可以将损失函数对于 $\theta$ 的偏导写为：

$$\nabla_\theta J_\theta = \nabla_\theta \mathcal{E}(w) + \nabla_\theta \log \sum_{w_i \in V} \exp(-\mathcal{E}(w_i)) \quad (18)$$

因为 $\log x$ 的导数是 $\frac{1}{x}$ ，则上式又可以改写为：

$$\begin{aligned} \nabla_\theta J_\theta &= \nabla_\theta \mathcal{E}(w) + \frac{1}{\sum_{w_i \in V} \exp(-\mathcal{E}(w_i))} \nabla_\theta \sum_{w_i \in V} \exp(-\mathcal{E}(w_i)) \\ &= \nabla_\theta \mathcal{E}(w) + \frac{1}{\sum_{w_i \in V} \exp(-\mathcal{E}(w_i))} \sum_{w_i \in V} \nabla_\theta \exp(-\mathcal{E}(w_i)) \\ &= \nabla_\theta \mathcal{E}(w) + \frac{1}{\sum_{w_i \in V} \exp(-\mathcal{E}(w_i))} \sum_{w_i \in V} \exp(-\mathcal{E}(w_i)) \nabla_\theta (-\mathcal{E}(w_i)) \\ &= \nabla_\theta \mathcal{E}(w) + \sum_{w_i \in V} \frac{\exp(-\mathcal{E}(w_i))}{\sum_{w_i \in V} \exp(-\mathcal{E}(w_i))} \nabla_\theta (-\mathcal{E}(w_i)) \end{aligned} \quad (19)$$

注意： $\frac{\exp(-\mathcal{E}(w_i))}{\sum_{w_i \in V} \exp(-\mathcal{E}(w_i))}$  就是词语 $w_i$ 的softmax概率值 $P(w_i)$ 。将其代入上面的等式中得到：

$$\begin{aligned} \nabla_\theta J_\theta &= \nabla_\theta \mathcal{E}(w) + \sum_{w_i \in V} P(w_i) \nabla_\theta (-\mathcal{E}(w_i)) \\ &= \nabla_\theta \mathcal{E}(w) - \sum_{w_i \in V} P(w_i) \nabla_\theta (\mathcal{E}(w_i)) \end{aligned} \quad (20)$$

梯度值可以分解为两个部分：一部分与目标词语 $w$ 正相关（等式右边的第一项），另一部分与其余所有词语负相关，按照各个词语的出现概率分配权重（等式右边的第二项）。我们可以发现，等式右边的第二项其实就是词表 $V$ 中所有词语 $w_i$ 的期望值：

$$\sum_{w_i \in V} P(w_i) \nabla_\theta (\mathcal{E}(w_i)) = E_{w_i \sim P} [\nabla_\theta (\mathcal{E}(w_i))] \quad (21)$$

现在大多数基于采样方法的核心都是用简单的过程来近似计算后一项的值。

如果已知网络模型的分布  $P(w)$ , 于是我们就可以从中随机采样  $m$  个词语  $w_1, \dots, w_m$ , 并用下面的公式计算期望值:

$$E_{w_i \sim P} [\nabla_\theta (\mathcal{E}(w_i))] \approx \frac{1}{m} \sum_{i=1}^m \nabla_\theta (\mathcal{E}(w_i)) \quad (22)$$

但是为了实现从概率值分布  $P$  中采样, 我们必须先计算得到  $P$ , 而这个过程正是我们想绕开的。于是, 我们用另一种类似于  $P$  但是采样更方便的分布  $Q$  来代替。在语言建模的任务中, 直接把训练集的 *unigram* 分布作为  $Q$  不失为良策。这就是经典的重要性采样的做法: 它使用蒙特卡洛方法得到分布  $Q$  来模拟真实的分布  $P$ 。可是, 被采样到的词语  $w$  仍然需要计算其概率值  $P(w)$ 。

上面把  $P_{w_i}$  赋值为  $\nabla_\theta (\mathcal{E}(w_i))$  的权重, 这里我们把权重值改为与  $Q$  相关的一个因子。这个因子是  $\frac{r(w_i)}{R}$ 。其中:

$$r(w) = \frac{\exp(-\mathcal{E}(w))}{Q(w)}, R = \sum_{j=1}^m r(w_j) \quad (23)$$

于是期望的估计值公式可以写为:

$$E_{w_i \sim P} [\nabla_\theta (\mathcal{E}(w_i))] \approx \sum_{i=1}^m \frac{r(w_i)}{R} \nabla_\theta (\mathcal{E}(w_i)) \quad (24)$$

若是采样的数量越少, 估计的分布与真实分布差别越大。如果样本数量非常少, 在训练过程中网络模型的分布  $P$  可能与 *unigram* 的分布  $Q$  差异很大, 会导致模型发散, 因此我们需要调整到合适的样本数量。Bengio 和 Senécal 的论文中介绍了一种快速选择样本数量的方法。最终的运算速度比传统的 softmax 提升了 19 倍。

注:

$$\frac{1}{R} \sum_{i=1}^m r(w_i) \nabla_\theta (\mathcal{E}(w_i)) = \sum_{i=1}^m \frac{\exp(-\mathcal{E}(w_i))}{\sum_{i=1}^m \exp(-\mathcal{E}(w_i))} \nabla_\theta (-\mathcal{E}(w_i)) \quad (25)$$

## 噪声对比估计

噪声对比估计是 Mnih 和 Teh 发明的一种比重要性采样更稳定的采样方法, 因为某些情况下重要性采样存在导致分布  $Q$  与  $P$  分道扬镳的风险。*NCE* 不是直接估计某个词语的概率值。相反, 它借助一个辅助的损失值, 从而实现了正确词语概率值最大化这一目标。

*NCE* 的想法: 训练一个模型来区分目标词语与噪声。于是待解决的问题就由预测正确的词语简化为一个二值分类器任务, 分类器试图将正确的词语与其它噪声样本中区分开来。对于每个词语  $w_i$ , 它的前  $n$  个词语  $w_{t-1}, \dots, w_{t-n+1}$  表示为  $w_i$  的语境  $c_i$ 。然后从含有噪声的分布  $Q$  中生成  $k$  个噪声样本  $\tilde{w}_{ik}$ 。参照重要性采样的方法, 这里也可以从训练数据的 *unigram* 分布中采样。由于分类器需要用到标签数据, 我们把语境  $c_i$  对应的所有正确的词语  $w_i$  标记为正样本 ( $y = 1$ ), 其余的噪声词语  $\tilde{w}_{ik}$  作为负样本 ( $y = 0$ )。

接着, 用逻辑回归模型来训练样本数据:

$$J_\theta = - \sum_{w_i \in V} [\log P(y = 1 | w_i, c_i) + k E_{\tilde{w}_{ik} \sim Q} [\log P(y = 0 | \tilde{w}_{ik}, c_i)]] \quad (26)$$

由于计算所有噪声样本的期望  $E_{\tilde{w}_{ik} \sim Q}$  仍需要对词表  $V$  中的词语求和, 得到标准化的概率值。于是可以采用蒙特卡洛方法来估算:

$$\begin{aligned} J_\theta &= - \sum_{w_i \in V} [\log P(y = 1 | w_i, c_i) + k \sum_{j=1}^k \frac{1}{k} \log P(y = 0 | \tilde{w}_{ij}, c_i)] \\ &= - \sum_{w_i \in V} [\log P(y = 1 | w_i, c_i) + \sum_{j=1}^k \log P(y = 0 | \tilde{w}_{ij}, c_i)] \end{aligned} \quad (27)$$

实际上，我们是从两个不同的分布中采样数据：正样本是根据语境 $c$ 从训练数据集 $P_{train}$ 中采样，而负样本从噪声分布 $Q$ 中采样获得。因此，无论是正样本还是负样本，其概率值都可以表示成上述两种分布带权重的组合，权重值对应于来自该分布的样本值：

$$P(y, w|c) = \frac{1}{k+1} P_{train}(w|c) + \frac{k}{k+1} Q(w) \quad (28)$$

于是，样本来自于 $P_{train}$ 的概率值可以表示为条件概率的形式：

$$\begin{aligned} P(y=1|w, c) &= \frac{\frac{1}{k+1} P_{train}(w|c)}{\frac{1}{k+1} P_{train}(w|c) + \frac{k}{k+1} Q(w)} \\ &= \frac{P_{train}(w|c)}{P_{train}(w|c) + kQ(w)} \end{aligned} \quad (29)$$

由于不知道 $P_{train}$ （待计算项），我们就用 $P$ 来代替：

$$P(y=1|w, c) = \frac{P(w|c)}{P(w|c) + kQ(w)} \quad (30)$$

当然，样本为负样本的概率值就是 $P(y=0|w, c) = 1 - P(y=1|w, c)$ 。值得注意的是，已知 $c$ 求词语 $w$ 出现的概率值 $P(w|c)$ 的计算方法实际上就是 $softmax$ 的定义：

$$P(w|c) = \frac{\exp(h^T v'_w)}{\sum_{w_i \in V} \exp(h^T v'_{w_i})} \quad (31)$$

因为分母只与 $h$ 相关， $h$ 的值与 $c$ 相关（假设 $V$ 不变），那么分母可以简化为 $Z(c)$ 来表示。 $softmax$ 就变为下面的形式：

$$P(w|c) = \frac{\exp(h^T v'_w)}{Z(c)} \quad (32)$$

为了求解 $Z(c)$ ，还是需要对 $V$ 中所有词语出现的概率值求和。 $NCE$ 则用了一个小技巧巧妙地绕开：即把标准化后的分母项 $Z(c)$ 当作模型的待学习参数。Mnih和Teh、Vaswani等在论文中都把 $Z(c)$ 的值固定设为1，他们认为这样不会对模型的效果造成影响。Zoph则认为，即使训练模型，最终得到 $Z(c)$ 的值也是趋近于1，并且方差很小。若是我们把上面 $softmax$ 等式中的 $Z(c)$ 项改为常数1，等式就变为：

$$P(w|c) = \exp(h^T v'_w) \quad (33)$$

再把上面的式子代入求解 $P(y=1|w, c)$ ，得到：

$$P(y=1|w, c) = \frac{\exp(h^T v'_w)}{\exp(h^T v'_w) + kQ(w)} \quad (34)$$

继续把上式代入逻辑回归的目标函数中，得到：

$$J_\theta = - \sum_{w_i \in V} [\log \frac{\exp(h^T v'_w)}{\exp(h^T v'_w) + kQ(w)} + \sum_{j=1}^k \log(1 - \frac{\exp(h^T v'_{\tilde{w}_{ij}})}{\exp(h^T v'_{\tilde{w}_{ij}}) + kQ(\tilde{w}_{ij})})] \quad (35)$$

$NCE$ 方法有非常完美的理论证明：随着噪声样本 $k$ 的数量增加， $NCE$ 导数趋近于 $softmax$ 函数的梯度。

Jozefowicz认为**NCE**与**IS**的相似点不仅在于它们都是基于采样的方法，而且相互之间联系非常紧密。**NCE**等价于解决二分类任务，他认为**IS**问题也可以用一个代理损失函数来描述：**IS**相当于用 $softmax$ 和交叉熵损失函数来优化解决多分类问题。他觉得**IS**是多分类问题，可能更适用于自然语言的建模，因为它迭代更新受到数据和噪声样本的共同作用，而**NCE**的迭代更新则是分别作用。事实上，Jozefowicz等人选用**IS**作为语言模型并且取得了最佳的效果。

## 负采样

负采样(Negative Sampling)可以被认为是NCE的一种近似版本。我们之前也提到过，随着样本数量 $k$ 的增加，NCE近似于softmax的损失。由于NEG的目标是学习高质量的词向量表示，而不是降低测试集的perplexity指标，于是NEG对NCE做了简化。

NEG也采用逻辑回归模型，使得训练集中词语的负对数似然最小。再回顾一下NCE的计算公式：

$$P(y = 1|w, c) = \frac{\exp(h^T v'_w)}{\exp(h^T v'_w) + kQ(w)} \quad (36)$$

NEG与NCE的关键区别在于NEG以尽可能简单的方式来估计这个概率值。为此，上式中计算量最大的 $kQ(w)$ 项被置为1，于是得到：

$$P(y = 1|w, c) = \frac{\exp(h^T v'_w)}{\exp(h^T v'_w) + 1} \quad (37)$$

当 $k = |V|$ 并且 $Q$ 是均匀分布时， $kQ(w) = 1$ 成立。此时，NEG等价于NCE。我们将 $kQ(w)$ 设置为1，而不是其它常数值的原因在于， $P(y = 1|w, c)$ 可以改写为sigmoid函数的形式：

$$P(y = 1|w, c) = \frac{1}{1 + \exp(-h^T v'_w)} \quad (38)$$

如果我们再把这个等式代入之前的逻辑回归损失函数中，可以得到：

$$\begin{aligned} J_\theta &= - \sum_{w_i \in V} [\log \frac{1}{1 + \exp(-h^T v'_w)} + \sum_{j=1}^k \log(1 - \frac{1}{1 + \exp(-h^T v'_{\tilde{w}_{ij}})})] \\ &= - \sum_{w_i \in V} [\log \frac{1}{1 + \exp(-h^T v'_w)} + \sum_{j=1}^k \log \frac{1}{1 + \exp(h^T v'_{\tilde{w}_{ij}})}] \\ &= - \sum_{w_i \in V} [\log \sigma(h^T v'_w) + \sum_{j=1}^k \log \sigma(-h^T v'_{\tilde{w}_{ij}})] \end{aligned} \quad (39)$$

而且仅当 $k = |V|$ 并且 $Q$ 是均匀分布时，NEG才等价于NCE。在其它情况下，NEG只是近似于NCE，也就是说前者不会直接优化正确词语的对数似然，所以不适合用于自然语言建模。NCE更适用于训练词向量表示。

## 马尔可夫过程

MCMC(Markov Chain Monte Carlo)的基础理论为马尔可夫过程，在MCMC算法中，为了在一个指定的分布上采样，根据马尔可夫过程，首先从任一状态出发，模拟马尔可夫过程，不断进行状态转移，最终收敛到平稳分布。

用蒙特卡罗方法随机模拟来求解一些复杂的连续积分或者离散求和的方法，但是这个方法需要得到对应的概率分布的样本集，而想得到这样的样本集很困难。因此我们需要马尔科夫链来帮忙。[参考链接](#)

## 马尔可夫链

设 $X_t$ 表示随机变量 $X$ 在离散时间 $t$ 时刻的取值。若该变量随时间变化的转移概率仅仅依赖于它的当前取值，即

$$P(X_{t+1} = s_j | X_0 = s_0, X_1 = s_1, \dots, X_t = s_t) = P(X_{t+1} = s_j | X_t = s_t) \quad (40)$$

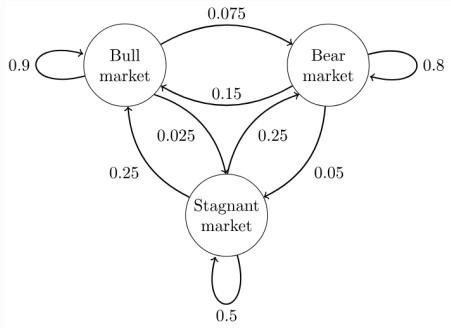
也就是说状态转移的概率只依赖于前一个状态，称这个变量为马尔可夫变量。这个性质称为马尔可夫性质，具有马尔可夫性质的随机过程称为马尔可夫过程。

马尔可夫链指的是在一段时间内随机变量 $X$ 的取值序列 $(X_0, X_1, \dots, X_m)$ ，它们满足如上的马尔可夫性质。

既然某一时刻状态转移的概率只依赖于它的前一个状态，那么我们只要能求出系统中任意两个状态之间的转换概率，这个马尔科夫链的模型就定了。

## 状态转移概率

这个马尔科夫链是表示股市模型的，共有三种状态：牛市(Bull market), 熊市(Bear market)和横盘(Stagnant market)。



每一个状态都以一定的概率转化到下一个状态。比如，牛市以0.025的概率转化到横盘的状态。这个状态概率转化图可以以矩阵的形式表示。如果我们定义矩阵  $P$  某一位置  $P(i, j)$  的值为  $P(j|i)$ ，即从状态  $i$  转化到状态  $j$  的概率，并定义牛市为状态0，熊市为状态1，横盘为状态2。这样我们得到了马尔科夫链模型的状态转移矩阵为：

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix} \quad (41)$$

## 状态转移矩阵的性质

假设我们当前股市的概率分布为：[0.3, 0.4, 0.3]，即30%概率的牛市，40%概率的熊市与30%的横盘。然后这个状态作为序列概率分布的初始状态  $t_0$ ，将其带入这个状态转移矩阵计算  $t_1, t_2, \dots$  的状态。代码如下：

```
1 import numpy as np
2 matrix = np.matrix([[0.9, 0.075, 0.025], [0.15, 0.8, 0.05], [0.25, 0.25, 0.5]], 
3 dtype=float)
4 vector = np.matrix([0.3, 0.3, 0.4], dtype=float)
5 for i in range(100):
6     vector = vector*matrix
7     print('current round: ', vector)
```

可以发现，从第60轮开始，我们的状态概率分布就不变了，一直保持在[0.625, 0.3125, 0.0625]，即62.5%的牛市，31.25%的熊市与6.25%的横盘。初始化概率分布[0.7, 0.1, 0.2]，结果保持不变。

同时，对于一个确定的状态转移矩阵  $P$ ，它的  $n$  次幂  $P^n$  在当  $n$  大于一定的值的时候也可以发现是确定的，

也就是说我们的马尔科夫链模型的状态转移矩阵收敛到的稳定概率分布与我们的初始状态概率分布无关。如果我们得到了这个稳定概率分布对应的马尔科夫链模型的状态转移矩阵，则我们可以用任意的概率分布样本开始，带入马尔科夫链模型的状态转移矩阵，这样经过一些序列的转换，最终就可以得到符合对应稳定概率分布的样本。这个性质不光对我们上面的状态转移矩阵有效，对于绝大多数其他的马尔科夫链模型的状态转移矩阵也有效。同时不光是离散状态，连续状态时也成立。

## 马尔科夫链收敛定理

马氏链定理：如果一个非周期马氏链具有转移概率矩阵  $P$ ，且它的任何两个状态是连通的，那么  $\lim_{n \rightarrow \infty} p_{ij}^n$  存在且与  $i$  无关，记  $\lim_{n \rightarrow \infty} p_{ij}^n = \pi$ ，我们有

1)

$$\lim_{n \rightarrow \infty} P_{ij}^n = \pi(j) \quad (42)$$

2)

$$\lim_{n \rightarrow \infty} P^n = \begin{bmatrix} \pi(1) & \cdots & \pi(2) & \cdots & \pi(j) & \cdots \\ \pi(1) & \cdots & \pi(2) & \cdots & \pi(j) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \pi(1) & \cdots & \pi(2) & \cdots & \pi(j) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (43)$$

3)

$$\pi_{(j)}^{t+1} = \sum_{i=0}^{\infty} \pi_{(i)}^t P_{ij} \quad (44)$$

4)  $\pi$ 是方程 $\pi P = \pi$ 的唯一非负解, 其中:

$$\pi = [\pi(1), \pi(2), \dots, \pi(j), \dots], \quad \sum_{i=0}^{\infty} \pi_i = 1 \quad (45)$$

上面的性质中需要解释的有:

- 非周期的马尔科夫链: 这个主要是指马尔科夫链的状态转化不是循环的, 如果是循环的则永远不会收敛。幸运的是我们遇到的马尔科夫链一般都是非周期性的。用数学方式表述则是: 对于任意某一状态*i*, *d*为集合 $\{n|n \geq 1, P_{ii}^n > 0\}$ 的最大公约数, 如果*d* = 1, 则该状态为非周期的。
- 任何两个状态是连通的: 是指存在一个*n*, 使得矩阵*P<sup>n</sup>*中的任何一个元素的数值都大于零。
- 马尔科夫链的状态数可以是有限的, 也可以是无限的。因此可以用于连续概率分布和离散概率分布。
- $\pi$ 通常称为马尔科夫链的平稳分布。
- 我们用*X<sub>i</sub>*表示在马氏链上跳转第*i*步后所处的状态, 如果 $\lim_{n \rightarrow \infty} P_{ij}^n = \pi(j)$ 存在, 很容易证明以上定理的第3个结论。由于

$$\begin{aligned} P(X_{n+1} = j) &= \sum_{i=0}^{\infty} P(X_n = i)P(X_{n+1} = j|X_n = i) \\ &= \sum_{i=0}^{\infty} P(X_n = i)P_{ij} \end{aligned} \quad (46)$$

上式两边取极限就得到 $\pi^{t+1}(j) = \sum_{i=0}^{\infty} \pi^t(i)P_{ij}$ 。假设状态的数目为*n*, 则有 $\pi^{t+1} = \pi^t \cdot P$ , 其中 $\pi^t = (\pi_1^{(t)}, \pi_2^{(t)}, \dots, \pi_n^{(t)})$ 。

- 稳定分布与特征向量的关系:

稳定分布 $\pi$ 是一个(行)向量, 它的元素都非负且和为1, 不随施加*P*操作而改变, 定义为 $\pi P = \pi$ 。

那么:  $P^T \pi^T = \pi^T$ ,

对比定义可以看出,这两个概念是相关的,并且 $\pi = \frac{e}{\sum_i e_i}$ 是由 $(\sum_i \pi_i = 1)$ 归一化的转移矩阵*P*的左特征向量 $e$ 的倍数, 其特征值为1.

操作上:

1. 对*P*的转置进行特征值分解得到特征向量和特征值。
2. 最大的特征值应为1,其对应的特征向量为矩阵的第*i*列
3. 对特征向量进行归一化,可以得到该状态转移矩阵的稳定分布

演示代码如下:

```

1 w,v=np.linalg.eig(P.transpose())
2 idx = np.where(np.abs(w-1.0)<1e-9)[0][0]
3 print w[idx]
4 print v[:,idx]/sum(v[:,idx])

```

实际上 $n * n$ 矩阵特征向量的一种求法就是用一个随机向量不断迭代与该矩阵相乘。

## 基于马尔科夫链采样

对于给定的概率分布 $p(x)$ ，我们希望能有便捷的方式生成它对应的样本。由于马氏链能收敛到平稳分布，于是一个很的漂亮想法是：如果我们能构造一个转移矩阵为 $P$ 的马氏链，使得该马氏链的平稳分布恰好是 $p(x)$ ，那么我们从任何一个初始状态 $x_0$ 出发沿着马氏链转移，得到一个转移序列 $x_0, x_1, \dots, x_n, x_{n+1}, \dots$ ，如果马氏链在第 $n$ 步已经收敛了，于是我们就得到了 $p(x)$ 的样本 $x_n, x_{n+1}, \dots$

从初始概率分布 $\pi^0$ 出发，我们在马氏链上做状态转移，记 $X_i$ 的概率分布为 $\pi^i$ ，则有

$$\begin{aligned} X_0 &\sim \pi^0(x) \\ X_i &\sim \pi^i(x) \\ \pi^i(x) &= \pi^{i-1}(x)P = \pi^0(x)P^n \end{aligned} \tag{47}$$

由马氏链收敛的定理，概率分布 $\pi^i(x)$ 将收敛到平稳分布 $\pi(x)$ 。假设到第 $n$ 步的时候马氏链收敛，则有

$$\begin{aligned} X_0 &\sim \pi^0(x) \\ X_1 &\sim \pi^1(x) \\ &\dots \\ X_n &\sim \pi^n(x) = \pi(x) \\ X_{n+1} &\sim \pi(x) \\ X_{n+2} &\sim \pi(x) \\ &\dots \end{aligned} \tag{48}$$

所以 $X_n, X_{n+1}, X_{n+2}, \dots \sim \pi(x)$ 都是同分布的随机变量，当然他们并不独立。如果我们从一个具体的初始状态 $x_0$ 开始，沿着马氏链按照概率转移矩阵做跳转，那么我们得到一个转移序列 $x_0, x_1, x_2, \dots, x_n, x_{n+1}, \dots$ 由于马氏链的收敛行为， $x_n, x_{n+1}, \dots$ 都将是平稳分布 $\pi(x)$ 的样本。

总结下基于马尔科夫链的采样过程：

- 输入Markov Chain状态转移概率矩阵 $P$ ，设定状态转移次数阈值 $n_1$ ，需要的样本个数 $n_2$ ；
- 从任意简单概率分布采样得到初始状态值 $x_0$ ；
- $for 0 \rightarrow n_1 + n_2 - 1 : from P(x|x_t) 中采样得到样本 x_{t+1}$ ；

样本集 $(x_{n_1}, \dots, x_{n_1+n_2-1})$ 即为 $P$ 平稳分布 $\pi$ 对应的样本集；

如果假定我们可以得到我们需要采样样本的平稳分布所对应的马尔科夫链状态转移矩阵，那么我们就可以用马尔科夫链采样得到我们需要的样本集，进而进行蒙特卡罗模拟。但是一个重要问题是，随意给定一个平稳分布 $\pi$ ，即目标分布 $p(x)$ ，如何得到它所对应的马尔科夫链状态转移矩阵 $P$ 呢？

## MCMC采样

在马尔科夫链中我们讲到给定一个概率平稳分布 $\pi$ ，很难直接找到对应的马尔科夫链状态转移矩阵 $P$ 。而只要解决这个问题，我们就可以找到一种通用的概率分布采样方法，进而用于蒙特卡罗模拟。

马氏链的收敛性质主要由转移矩阵 $P$ 决定，所以基于马氏链做采样的关键问题是如何构造转移矩阵 $P$ ，使得平稳分布恰好是我们要的分布 $p(x)$ 。如何能做到这一点呢？我们主要使用如下的定理。

## 细致平稳条件

如果非周期马氏链的转移矩阵  $P$  和分布  $\pi(x)$  满足:

$$\pi(i)P_{ij} = \pi(j)P_{ji} \quad \text{for all } i, j \quad (49)$$

则  $\pi(x)$  是马氏链的平稳分布, 上式被称为细致平稳条件。

其实这个定理是显而易见的, 因为细致平稳条件的物理含义就是对于任何两个状态  $i, j$ , 从  $i$  转移出去到  $j$  而丢失的概率质量, 恰好会被从  $j$  转移回  $i$  的概率质量补充回来, 所以状态  $i$  上的概率质量  $\pi(i)$  是稳定的, 从而  $\pi(x)$  是马氏链的平稳分布。数学上的证明也很简单, 由细致平稳条件可得:

$$\begin{aligned} \sum_{i=1}^{\infty} \pi(i)P_{ij} &= \sum_{i=1}^{\infty} \pi(j)P_{ji} = \pi(j) \sum_{i=1}^{\infty} P_{ji} = \pi(j) \\ \Rightarrow \pi P &= \pi \end{aligned} \quad (50)$$

由于  $\pi$  是方程  $\pi P = \pi$  的解, 所以  $\pi$  是  $P$  的平稳分布。

**注:** 细致平稳条件为马尔可夫链有平稳分布的充分条件

假设我们已经有一个转移矩阵为  $Q$  马氏链 ( $q(i, j)$  表示从状态  $i$  转移到状态  $j$  的概率, 也可以写为  $q(j|i)$  或者  $q(i \rightarrow j)$ ), 显然, 通常情况下

$$p(i)q(i, j) \neq p(j)q(j, i) \quad (51)$$

也就是细致平稳条件不成立, 所以  $p(x)$  不太可能是这个马氏链的平稳分布。我们可否对马氏链做一个改造, 使得细致平稳条件成立呢? 譬如, 我们引入一个  $\alpha(i, j)$ , 我们希望:

$$p(i)q(i, j)\alpha(i, j) = p(j)q(j, i)\alpha(j, i) \quad (52)$$

取什么样的  $\alpha(i, j)$  以上等式能成立呢? 最简单的, 按照对称性, 我们可以取:

$$\begin{aligned} \alpha(i, j) &= p(j)q(j, i) \\ \alpha(j, i) &= p(i)q(i, j) \end{aligned} \quad (53)$$

所以有:

$$\underbrace{p(i)q(i, j)\alpha(i, j)}_{Q'(i, j)} = \underbrace{p(j)q(j, i)\alpha(j, i)}_{Q'(j, i)} \quad (***) \quad (54)$$

于是我们把原来具有转移矩阵  $Q$  的一个很普通的马氏链, 改造为了具有转移矩阵  $Q'$  的马氏链, 而  $Q'$  恰好满足细致平稳条件, 由此马氏链  $Q'$  的平稳分布就是  $p(x)$ !

在改造  $Q$  的过程中引入的  $\alpha(i, j)$  称为接受率, 物理意义可以理解为在原来的马氏链上, 从状态  $i$  以  $q(i, j)$  的概率转跳转到状态  $j$  的时候, 我们以  $\alpha(i, j)$  的概率接受这个转移, 于是得到新的马氏链  $Q'$  的转移概率为  $q(i, j)\alpha(i, j)$ 。

为了使  $q(i, j)\alpha(i, j)$  满足细致平稳条件. 一般来说  $q(i, j)\alpha(i, j)$  也是不方便直接采样的. 实际的做法是采用拒绝采样方法, 把  $\alpha(i, j)$  看作一个状态转移的接受概率. 从  $(0, 1)$  均匀分布中做一个采样得到  $u$ , 如果  $u < \alpha(i, j)$  则接受  $q(i, j)$  采样出样本的状态转移, 否则拒绝, 保持原状态。

**马氏链转移和接受概率:**

假设我们已经有一个转移矩阵  $Q$  (对应元素为  $q(i, j)$ ), 把以上的过程整理一下, 我们就得到了如下的用于采样概率分布  $p(x)$  的算法。

1: 输入任意选定的马尔可夫链状态转移矩阵  $Q$ , 平稳分布  $p(x)$ , 状态转移次数阈值  $n_1$ , 需要的样本个数  $n_2$ ;

2: 从任意简单概率分布采样得到初始状态值  $x_0$ ;

3: for  $0 \rightarrow n_1 + n_2 - 1$ :

- 从条件概率分布  $q(x|x_t)$  中采样  $y \sim q(x|x_t)$
- 从均匀分布采样  $u \sim Uniform[0, 1]$
- 如果  $u < \alpha(x_t, y) = p(y)q(y|x_t)$ , 则接受转移  $x_t \rightarrow y$ , 即  $x_{t+1} = y$
- 否则拒绝转移, 即  $t = max\{t - 1, 0\}$

上述过程中  $p(x), q(x|y)$  说的都是离散的情形, 事实上即便这两个分布是连续的, 以上算法仍然是有效, 于是就得到更一般的连续概率分布  $p(x)$  的采样算法, 而  $q(x|y)$  就是任意一个连续二元概率分布对应的条件分布。

由于  $\alpha(x_t, y)$  可能非常的小, 比如 0.1, 导致我们大部分的采样值都被拒绝转移, 采样效率很低。有可能我们采样了上百万次马尔可夫链还没有收敛, 也就是上面这个  $n_1$  要非常非常的大, 这让人难以接受, 怎么办呢? 这时就轮到我们的 M-H 采样出场了。

## M-H 采样

M-H 算法主要是解决接受率过低的问题, 回顾 MCMC 采样的细致平稳条件:

$$p(i)q(i, j)\alpha(i, j) = p(j)q(j, i)\alpha(j, i) \quad (55)$$

我们采样效率低的原因是  $\alpha(i, j)$  太小了, 比如  $\alpha(i, j)$  为 0.1, 而  $\alpha(j, i)$  为 0.2. 即:

$$p(i)Q(i, j) \times 0.1 = p(j)Q(j, i) \times 0.2 \quad (56)$$

如果两边同时扩大五倍, 接受率提高到了 0.5, 但是细致平稳条件却仍然是满足的, 即:

$$p(i)Q(i, j) \times 0.5 = p(j)Q(j, i) \times 1 \quad (57)$$

这样我们的接受率可以做如下改进, 即:

$$\alpha(i, j) = min\left(\frac{p(j)Q(j, i)}{p(i)Q(i, j)}, 1\right) \quad (58)$$

于是, 经过对上述 MCMC 采样算法中接受率的微小改造, 我们就得到了如下教科书中最常见的 Metropolis-Hastings 算法。

1: 输入任意选定的马尔可夫链状态转移矩阵  $Q$ , 平稳分布  $p(x)$ , 状态转移次数阈值  $n_1$ , 需要的样本个数  $n_2$ ;

2: 从任意简单概率分布采样得到初始状态值  $x_0$ ;

3: for  $0 \rightarrow n_1 + n_2 - 1$ :

- 从条件概率分布  $q(x|x_t)$  中采样  $y \sim q(x|x_t)$
- 从均匀分布采样  $u \sim Uniform[0, 1]$
- 如果  $u < \alpha(x_t, y) = min\{1, \frac{p(y)q(y|x_t)}{p(x_t)q(x_t|y)}\}$ , 则接受转移  $x_t \rightarrow y$ , 即  $x_{t+1} = y$
- 否则拒绝转移, 即  $t = max\{t - 1, 0\}$

举例: 我们的目标平稳分布是一个均值 3, 标准差 2 的正态分布, 而选择的马尔可夫链状态转移矩阵  $Q(i, j)$  的条件转移概率是以  $i$  为均值, 方差 1 的正态分布在位置  $j$  的值。

M-H 采样完整解决了使用蒙特卡罗方法需要的任意概率分布样本集的问题, 因此在实际生产环境得到了广泛的应用。但是在大数据时代, M-H 采样面临着两大难题:

- 数据特征非常多, 需要计算接受率, 在高维时计算量大。并且由于接受率的原因导致算法收敛时间变长;
- 由于特征维度大, 特征的条件概率分布好求, 但是特征的联合分布不好求。

这时候我们能不能只有各维度之间条件概率分布的情况下方便的采样呢?

## Gibbs Sampling

## 重新寻找合适的细致平稳条件

从二位数据分布开始，假设 $\pi(x_1, x_2)$ 是一个二维联合概率分布，观察第一个特征维度相同的两个点 $A(x_1^{(1)}, x_2^{(1)})$ 和 $A(x_1^{(1)}, x_2^{(2)})$ ，容易发现下面两式成立：

$$\begin{aligned}\pi(x_1^{(1)}, x_2^{(1)})\pi(x_2^{(2)}|x_1^{(1)}) &= \pi(x_1^{(1)})\pi(x_2^{(1)}|x_1^{(1)})\pi(x_2^{(2)}|x_1^{(1)}) \\ \pi(x_1^{(1)}, x_2^{(2)})\pi(x_2^{(1)}|x_1^{(1)}) &= \pi(x_1^{(1)})\pi(x_2^{(2)}|x_1^{(1)})\pi(x_2^{(1)}|x_1^{(1)})\end{aligned}\quad (59)$$

所以有：

$$\pi(x_1^{(1)}, x_2^{(1)})\pi(x_2^{(2)}|x_1^{(1)}) = \pi(x_1^{(1)}, x_2^{(2)})\pi(x_2^{(1)}|x_1^{(1)}) \quad (60)$$

即：

$$\pi(A)\pi(x_2^{(2)}|x_1^{(1)}) = \pi(B)\pi(x_2^{(1)}|x_1^{(1)}) \quad (61)$$

观察上式再观察细致平稳条件的公式，我们发现在 $x_1 = x_1^{(1)}$ 这条直线上如果用条件概率分布 $\pi(x_2|x_1^{(1)})$ 作为马尔可夫链的状态转移概率，则任意两个点之间的转移满足细致平稳条件！同样的道理，在 $x_2 = x_2^{(1)}$ 这条直线上，如果用条件概率分布 $\pi(x_1|x_2^{(1)})$ 作为马尔可夫链的状态转移概率，则任意两个点之间的转移也满足细致平稳条件。那是因为假如有一点 $C(x_1^{(2)}, x_2^{(1)})$ ，我们可以得到：

$$\pi(A)\pi(x_1^{(2)}|x_2^{(1)}) = \pi(C)\pi(x_1^{(1)}|x_2^{(1)}) \quad (62)$$

基于上面的发现，我们可以这样构造分布 $\pi(x_1, x_2)$ 的马尔可夫链对应的状态转移矩阵 $P$ ：

$$\begin{aligned}P(A \rightarrow B) &= \pi(x_2^{(B)}|x_1^{(1)}) \quad if \ x_1^{(A)} = x_1^{(B)} = x_1^{(1)} \\ P(A \rightarrow C) &= \pi(x_1^{(C)}|x_2^{(1)}) \quad if \ x_2^{(A)} = x_2^{(C)} = x_2^{(1)} \\ P(A \rightarrow D) &= 0 \quad else\end{aligned}\quad (63)$$

有了这个状态转移矩阵，我们很容易验证平面上的任意两点 $E, F$ ，满足细致平稳条件：

$$\pi(E)P(E \rightarrow F) = \pi(F)P(F \rightarrow E) \quad (64)$$

## 二维Gibbs采样

利用上一节找到的状态转移矩阵，我们就得到了二维Gibbs采样，这个采样需要两个维度之间的条件概率。具体过程如下：

1: 输入平稳分布 $\pi(x_1, x_2)$ ，设定状态转移次数阈值 $n_1$ ，需要的样本个数 $n_2$

2: 随机初始化状态值 $x_1^{(1)}$  和 $x_2^{(1)}$

3: *for*  $t = 0 \rightarrow n_1 + n_2 - 1$ :

- 从条件概率分布 $P(x_2|x_1^{(t)})$ 中采样得到样本 $x_2^{(t+1)}$
- 从条件概率分布 $P(x_1|x_2^{(t+1)})$ 中采样得到样本 $x_1^{(t+1)}$

样本集 $\{(x_1^{(n_1)}, x_2^{(n_1)}), (x_1^{(n_1+1)}, x_2^{(n_1+1)}), \dots, (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)})\}$ 就是我们需要的平稳分布对应的样本集。

采样是在两个坐标轴上不停的轮换的。当然，坐标轴轮换不是必须的，我们也可以每次随机选择一个坐标轴进行采样。不过常用的Gibbs采样的实现都是基于坐标轴轮换的。

## 多维Gibbs采样

上面的这个算法推广到多维的时候也是成立的。比如一个 $n$ 维的概率分布 $\pi(x_1, x_2, \dots, x_n)$ ，可以通过在 $n$ 个坐标轴上轮换采样，来得到新的样本。对于轮换到的任意一个坐标轴 $x_i$ 上的转移，马尔科夫链的状态转移概率为 $P(x_i|x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ ，即固定 $n-1$ 个坐标轴，在某一个坐标轴上移动。具体的算法过程如下：

1: 输入平稳分布 $\pi(x_1, x_2, \dots, x_n)$ , 设定状态转移次数阈值 $n_1$ , 需要的样本个数 $n_2$

2: 随机初始化状态值 $(x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)})$

3: *for*  $t = 0 \rightarrow n_1 + n_2 - 1$ :

- 从条件概率分布 $P(x_1|x_2^{(t)}, x_3^{(t)}, \dots, x_n^{(t)})$ 中采样得到样本 $x_1^{(t+1)}$
- 从条件概率分布 $P(x_2|x_1^{(t+1)}, x_3^{(t+1)}, \dots, x_n^{(t+1)})$ 中采样得到样本 $x_2^{(t+1)}$
- ...
- 从条件概率分布 $P(x_j|x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{j-1}^{(t+1)}, x_{j+1}^{(t+1)}, \dots, x_n^{(t+1)})$ 中采样得到样本 $x_j^{(t+1)}$
- ...
- 从条件概率分布 $P(x_n|x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{n-1}^{(t+1)})$ 中采样得到样本 $x_n^{(t+1)}$

样本集 $\{(x_1^{(n_1)}, x_2^{(n_1)}, \dots, x_n^{(n_1)}), \dots, (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)}, \dots, x_n^{(n_1+n_2-1)})\}$ 就是我们需要的平稳分布对应的样本集。

整个采样过程和*Lasso*回归的坐标轴下降法算法非常类似, 只不过*Lasso*回归是固定 $n - 1$ 个特征, 对某一个特征求极值。而*Gibbs*采样是固定 $n - 1$ 个特征在某一个特征采样。

## 二维Gibbs采样实例

假设我们要采样的是一个二维正态分布  $Norm(\mu, \Sigma)$ , 其中:

$$\begin{aligned}\mu &= (\mu_1, \mu_2) = (5, -1) \\ \Sigma &= \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}\end{aligned}\tag{65}$$

而采样过程中的需要的状态转移条件分布为:

$$\begin{aligned}P(x_1|x_2) &= Norm(\mu_1 + \rho\sigma_1/\sigma_2(x_2 - \mu_2), 1 - \rho^2\sigma_1^2) \\ P(x_2|x_1) &= Norm(\mu_2 + \rho\sigma_2/\sigma_1(x_1 - \mu_1), 1 - \rho^2\sigma_2^2)\end{aligned}\tag{66}$$

## Gibbs采样小结

由于Gibbs采样在高维特征时的优势, 目前我们通常意义上的MCMC采样都是用的Gibbs采样。

当然Gibbs采样是从M-H采样的基础上的进化而来的, 同时Gibbs采样要求数据至少有两个维度, 一维概率分布的采样是没法用Gibbs采样的, 这时M-H采样仍然成立。

有了Gibbs采样来获取概率分布的样本集, 有了蒙特卡罗方法来用样本集模拟求和, 他们一起就奠定了MCMC算法在大数据时代高维数据模拟求和时的作用。

## 采样方法总结

问题: 已知随机变量 $X$ 的概率密度为 $p(x)$ ,  $f(x)$ 为 $X$ 的函数,  $E(f(X)) = \int_a^b f(x)p_X(x)dx$ 。

根据MCMC原理, 如果能从 $p(x)$ 中抽样 $x_i$ , 然后对这些 $f(x_i)$ 取平均即可近似 $f(x)$ 的期望  $E_N(f) = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}$ 。

## 逆变换采样

通过CDF与PDF的关系, 得到样本 $x_i$ 。

缺点:

- 有时CDF不好求
- CDF的反函数不好求

## 接受拒绝采样

从容易抽样的 $q(x)$ 中抽样，以一定的方法拒绝某些样本，达到接近 $p(x)$ 分布的目的。

最终得到 $n$ 个接受的的样本 $x_0, x_1, \dots, x_n$ ，则最后的蒙特卡洛求解结果为： $\frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{q(x_i)}$ 。

缺点：

- 合适的 $q$ 分布比较难以找到
- 很难确定一个合理的 $c$ 值。

## 重要性采样

从容易抽样的概率分布 $q(x)$ 中抽样，不需要拒绝样本。

最终得到 $n$ 个接受的的样本 $x_0, x_1, \dots, x_n$ ，则最后的蒙特卡洛求解结果为： $\frac{1}{n} \sum_{i=1}^n \frac{f(x_i) \cdot p(x)}{q(x_i)}$ 。

## 基于马尔科夫链采样

对于给定的概率分布 $p(x)$ 。我们希望能有便捷的方式生成它对应的样本。获得样本之后即可对期望进行蒙特卡洛近似。如果我们将构造一个转移矩阵为 $P$ 的马氏链，使得该马氏链的平稳分布恰好是 $p(x)$ ，那么我们从任何一个初始状态 $x_0$ 出发沿着马氏链转移，得到一个转移序列 $x_0, x_1, \dots, x_n, x_{n+1}, \dots$ 。如果马氏链在第 $n$ 步已经收敛了，于是我们就得到了 $p(x)$ 的样本 $x_n, x_{n+1}, \dots$ 。

重要的问题：随意给定一个平稳分布 $\pi$ ，即目标分布 $p(x)$ ，如何得到它所对应的马尔科夫链状态转移矩阵 $P$ 呢？

## MCMC采样

给定的概率分布 $p(x)$ ，一个转移矩阵为 $Q$ 马氏链，构造新的转移矩阵 $Q'$ ，使得 $p(x)$ 是 $Q'$ 的平稳分布。新的马氏链 $Q'$ 的转移概率为 $q(i, j)\alpha(i, j)$ ，其中 $\alpha(i, j) = p(j)q(j, i)$ 。

这时，从 $Q'$ 的边缘转移概率采样，得到的样本服从 $p(x)$ 的概率分布。但是 $Q'$ 的边缘转移概率含有 $p(x)$ ，不易采样。

采用拒绝采样的做法，目的：从 $q(x_t, x)\alpha(x_t, x)$ 中采样，选择的 proposal distribution 为 $q(x_t, x)$ ，因为 $\alpha(x_t, x) < 1$ ，故选择常量 $c = 1$ ，这时，接受概率为：

$$\alpha = \frac{q(x_t, x)\alpha(x_t, x)}{c \cdot q(x_t, x)} = \alpha(x_t, x) \quad (67)$$

故从均匀分布 $(0, 1)$ 中抽样得到 $u$ ，如果 $u < \alpha(x_t, y) = p(y)q(y|x_t)$ ，则接受转移 $x_t \rightarrow y$ 。

缺点：由于 $\alpha(x_t, y)$ 可能非常的小，比如 $0.1$ ，导致我们大部分的采样值都被拒绝转移，采样效率很低。

## M-H采样

经过对上述 MCMC 采样算法中接受率的微小改造，令 $\alpha(i, j) = \min\left(\frac{p(j)Q(j, i)}{p(i)Q(i, j)}, 1\right)$ 。

缺点：

- 数据特征非常多，需要计算接受率，在高维时计算量大。并且由于接受率的原因导致算法收敛时间变长；
- 由于特征维度大，特征的条件概率分布好求，但是特征的联合分布不好求。

## Gibbs采样

采用了轮换坐标轴的方法进行采样，能够处理高维特征。要求数据至少有两个维度，一维概率分布的采样是没法用Gibbs采样的，这时M-H采样仍然成立。

有了Gibbs采样来获取概率分布的样本集，有了蒙特卡罗方法来用样本集模拟求和，他们一起就奠定了MCMC算法在大数据时代高维数据模拟求和时的作用。