漫漫成长
算法工程师

以下内容来自刘建平Pinard-博客园的学习笔记，总结如下：

1 MCMC蒙特卡罗方法

作为一种随机采样方法，马尔科夫链蒙特卡罗（Markov Chain Monte Carlo，以下简称MCMC）在机器学习,深度学习以及自然语言处理等领域都有广泛的应用，是很多复杂算法求解的基础。下面我们就对MCMC的原理做一个总结。

1.1 MCMC概述

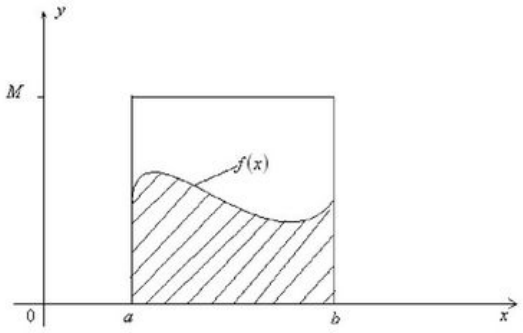
从名字我们可以看出，MCMC由两个MC组成，即蒙特卡罗方法（Monte Carlo Simulation，简称MC）和马尔科夫链（Markov Chain ，也简称MC）。要弄懂MCMC的原理我们首先得搞清楚蒙特卡罗方法和马尔科夫链的原理。本节关注于蒙特卡罗方法。

1.2 蒙特卡罗方法引入

蒙特卡罗原来是一个赌场的名称，用它作为名字大概是因为蒙特卡罗方法是一种随机模拟的方法，这很像赌博场里面的扔骰子的过程。最早的蒙特卡罗方法都是为了求解一些不太好求解的求和或者积分问题。比如积分：

$$\theta = \int_a^b f(x)dx$$

如果我们很难求解出f(x)的原函数，那么这个积分比较难求解。当然我们可以通过蒙特卡罗方法来模拟求解近似值。如何模拟呢？假设我们函数图像如下图:



则一个简单的近似求解方法是在[a,b]之间随机的采样一个点。比如 x_0 ，然后用 $f(x_0)$ 代表在[a,b]区间上所有的 $f(x)$ 的值。那么上面的定积分的近似求解为:

$$(b-a)f(x_0)$$

当然，用一个值代表[a,b]区间上所有的 $f(x)$ 的值，这个假设太粗糙。那么我们可以采样[a,b]区间的n个值： x_0， $x_1, \ldots, x_{n-1}$ ，用它们的均值来代表[a,b]区间上所有的 $f(x)$ 的值。这样我们上面的定积分的近似求解为:

$$\frac{b-a}{n}\sum_{i=0}^{n-1} f(x_i)$$

结果很可能和真实值相差甚远。

怎么解决这个问题呢?

如果我们可以得到x在[a,b]的概率分布函数p(x)，那么我们的定积分求和可以这样进行:

36

$$\theta = \int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)}p(x)dx \approx \frac{1}{n}\sum_{i=0}^{n-1}\frac{f(x_i)}{p(x_i)}$$

上式最右边的这个形式就是蒙特卡罗方法的一般形式。当然这里是连续函数形式的蒙特卡罗方法，但是在离散时一样成立。

可以看出，最上面我们假设x在[a,b]之间是均匀分布的时候， $p(x_i) = 1/(b-a)$ ，带入我们有概率分布的蒙特卡罗积分的上式，可以得到:

$$\frac{1}{n}\sum_{i=0}^{n-1}\frac{f(x_i)}{1/(b-a)} = \frac{b-a}{n}\sum_{i=0}^{n-1}f(x_i)$$

也就是说，我们最上面的均匀分布也可以作为一般概率分布函数p(x)在均匀分布时候的特例。那么我们现在的问题转到了如何求出x的分布p(x)的若干和样本上来。

1.3 概率分布采样

上一节我们讲到蒙特卡罗方法的关键是得到x的概率分布。如果求出了x的概率分布，我们可以基于概率分布去采样基于这个概率分布的n个x的样本集，带入蒙特卡罗求和的式子即可求解。但是还有一个关键的问题需要解决，即如何基于概率分布去采样基于这个概率分布的n个x的样本集。

对于常见的均匀分布 $uniform(0,1)$ 是非常容易采样样本的，一般通过线性同余发生器可以很方便的生成(0,1)之间的伪随机数样本。而其他常见的概率分布，无论是离散的分布还是连续的分布，它们的样本都可以通过 $uniform(0,1)$ 的样本转换而得。比如二维正态分布的样本 (Z_1, Z_2) 可以通过通过独立采样得到的 $uniform(0,1)$ 样本对 (X_1, X_2) 通过如下的式子转换而得:

$$Z_1 = \sqrt{-2lnX_1}\,cos(2\pi X_2)$$

$$Z_2 = \sqrt{-2lnX_1}\,sin(2\pi X_2)$$

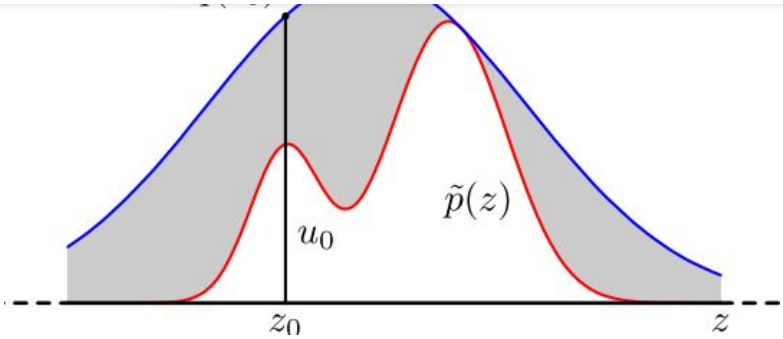
其他一些常见的连续分布，比如t分布，F分布，Beta分布，Gamma分布等，都可以通过类似的方式从uniform(0,1)得到的采样样本转化得到。在python的numpy，scikit-learn等类库中，都有生成这些常用分布样本的函数可以使用。

不过很多时候，我们的x的概率分布不是常见的分布，这意味着我们没法方便的得到这些非常见的概率分布的样本集。那这个问题怎么解决呢?

1.4 接受-拒绝采样

对于概率分布不是常见的分布，一个可行的办法是采用接受-拒绝采样来得到该分布的样本。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采样的分布 q(x) 比如高斯分布，然后按照一定的方法拒绝某些样本，以达到接近 p(x) 分布的目的，其中q(x)叫做 proposal distribution。

36



具体采用过程如下，设定一个方便采样的常用概率分布函数 q(x)，以及一个常量 k，使得 p(x) 总在 kq(x) 的下方。如上图。

首先，采样得到$q(x)$的一个样本z_0，采样方法如第三节。然后，从均匀分布$(0, kq(z_0))$中采样得到一个值u。如果u落在了上图中的灰色区域，则拒绝这次抽样，否则接受这个样本z_0。重复以上过程得到n个接受的样本$z_0, z_1, \ldots z_{n-1}$，则最后的蒙特卡罗方法求解结果为：

$$\frac{1}{n}\sum_{i=0}^{n-1}\frac{f(z_i)}{p(z_i)}$$

整个过程中，我们通过一系列的接受拒绝决策来达到用q(x)模拟p(x)概率分布的目的。

1.5 蒙特卡罗方法小结

使用接受-拒绝采样，我们可以解决一些概率分布不是常见的分布的时候，得到其采样集并用蒙特卡罗方法求和的目的。但是接受-拒绝采样也只能部分满足我们的需求，在很多时候我们还是很难得到我们的概率分布的样本集。比如：

1）对于一些二维分布$p(x, y)$，有时候我们只能得到条件分布$p(x|y)$和$p(y|x)$和却很难得到二维分布$p(x, y)$一般形式，这时我们无法用接受-拒绝采样得到其样本集。

2）对于一些高维的复杂非常见分布$p(x_1, x_2, \ldots, x_n)$，我们要找到一个合适的$q(x)$和k非常困难。

从上面可以看出，要想将蒙特卡罗方法作为一个通用的采样模拟求和的方法，必须解决如何方便得到各种复杂概率分布的对应的采样样本集的问题。马尔科夫链就是帮助找到这些复杂概率分布的对应的采样样本集的白衣骑士。

2 MCMC马尔科夫链

用蒙特卡罗方法来随机模拟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。

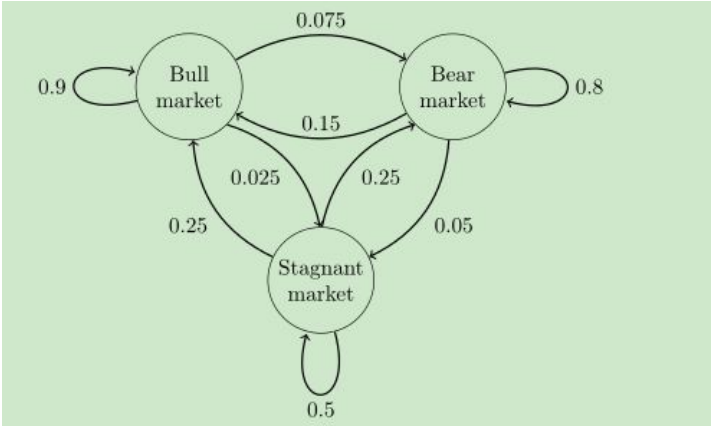
2.1 马尔科夫链概述

马尔科夫链定义本身比较简单，它假设某一时刻状态转移的概率只依赖于它的前一个状态。举个形象的比喻，假如每天的天气是一个状态的话，那个今天是不是晴天只依赖于昨天的天气，而和前天的天气没有任何关系。当然这么说可能有些武断，但是这样做可以大大简化模型的复杂度，因此马尔科夫链在很多时间序列模型中得到广泛的应用，比如循环神经网络RNN，隐式马尔科夫模型HMM等，当然MCMC也需要它。

如果用精确的数学定义来描述，则假设我们的序列状态是

$$P(X_{t+1}|\dots X_{t-2}, X_{t-1}, X_t) = P(X_{t+1}|X_t)$$

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。我们来看看下图这个马尔科夫链模型的具体的例子(来源于维基百科)。



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：

$$P = \begin{pmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}$$

马尔科夫链模型的状态转移矩阵和我们蒙特卡罗方法需要的概率分布样本集有什么关系呢？

这需要从马尔科夫链模型的状态转移矩阵的性质讲起。

2.2 马尔科夫链模型状态转移矩阵的性质

得到了马尔科夫链模型的状态转移矩阵，我们来看看马尔科夫链模型的状态转移矩阵的性质。

仍然以上面的这个状态转移矩阵为例。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%概率的熊盘与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：

```
import numpy as np
matrix = np.matrix([[0.9,0.075,0.025],[0.15,0.8,0.05],[0.25,0.25,0.5]], dtype=float)
vector1 = np.matrix([[0.3,0.4,0.3]], dtype=float)
for i in range(100):
    vector1 = vector1*matrix
    print "Current round:" , i+1
    print vector1
```

部分输出结果如下：

```
Current round: 1
[[ 0.405   0.4175  0.1775]]
```

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。那么这个是巧合吗？

我们现在换一个初始概率分布试一试，现在我们用[0.7,0.1,0.2]作为初始概率分布，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 $t_1, t_2, \ldots$ 的状态。

代码如下：

```
matrix = np.matrix([[0.9,0.075,0.025],[0.15,0.8,0.05],[0.25,0.25,0.5]], dtype=float)
vector1 = np.matrix([[0.7,0.1,0.2]], dtype=float)
for i in range(100):
    vector1 = vector1*matrix
    print "Current round:" , i+1
    print vector1
```

部分输出结果如下：

```
Current round: 1
[[ 0.695    0.1825   0.1225]]
Current round: 2
[[ 0.6835    0.22875   0.08775]]
```

可以看出，尽管这次我们采用了不同初始概率分布，最终状态的概率分布趋于同一个稳定的概率分布[0.625　0.3125　0.0625]，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。这是一个非常好的性质，也就是说，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

布的样本。

这个性质不光对我们上面的状态转移矩阵有效，对于绝大多数的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。

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，我们还是以上面的例子为例，计算代码如下：

```
matrix = np.matrix([[0.9,0.075,0.025],[0.15,0.8,0.05],[0.25,0.25,0.5]], dtype=float)
for i in range(10):
    matrix = matrix*matrix
    print "Current round:" , i+1
    print matrix
```

输出结果如下：

```
Current round: 1
[[ 0.8275    0.13375   0.03875]
 [ 0.2675    0.66375   0.06875]
 [ 0.3875    0.34375   0.26875]]
Current round: 2
[[ 0.73555    0.212775   0.051675]
 [ 0.42555    0.499975   0.074475]
 [ 0.51675    0.372375   0.110875]]
```

我们可以发现，在n≥6以后， P^n 的值稳定不再变化，而且每一行都为[0.625　0.3125　0.0625]，这和我们前面的稳定分布是一致的。这个性质同样不光是离散状态，连续状态时也成立。

用数学语言总结下马尔科夫链的收敛性质：

$\lim_{n\to\infty} P_{ij}$ 形式，我们有：

1）

$$\lim_{n\to\infty} P_{ij}^n = \pi(j)$$

36

2）

$$\lim_{n\to\infty} P^n = \begin{pmatrix} \pi(1) & \pi(2) & \dots & \pi(j) & \dots \\ \pi(1) & \pi(2) & \dots & \pi(j) & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \pi(1) & \pi(2) & \dots & \pi(j) & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

3）

$$\pi(j) = \sum_{i=0}^{\infty} \pi(i) P_{ij}$$

4) π是方程$\pi P = \pi$的唯一非负解，其中：

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

上面的性质中需要解释的有：

1）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。

2）任何两个状态是连通的：这个指的是从任意一个状态可以通过有限步到达其他的任意一个状态，不会出现条件概率一直为0导致不可达的情况。

3）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。

4） π 通常称为马尔科夫链的平稳分布。

2.3 基于马尔科夫链采样

如果我们得到了某个平稳分布所对应的马尔科夫链状态转移矩阵，我们就很容易采用出这个平稳分布的样本集。

假设我们任意初始的概率分布是$\pi_0(x)$，经过第一轮马尔科夫链状态转移后的概率分布是$\pi_1(x)$，。。。。第n轮的概率分布是$\pi_i(x)$。假设经过n轮后马尔科夫链收敛到我们的平稳分布$\pi(x)$，即：

$$\pi_n(x) = \pi_{n+1}(x) = \pi_{n+2}(x) = \ldots = \pi(x)$$

对于每个分布$\pi_i(x)$，我们有：

$$\pi_i(x) = \pi_{i-1}(x)P = \pi_{i-2}(x)P^2 = \pi_0(x)P^i$$

现在我们可以开始采样了，首先，基于初始任意简单概率分布比如高斯分布 $\pi_0(x)$ 采样得到状态值 x_0 ，基于条件概率分布 $P(x|x_0)$ 采样状态值 x_1 ，一直进行下去，当状态转移进行到一定的次数时，比如到n次时，我们认为此时的采样集 $(x_n, x_{n+1}, \ldots)$ 即是符合我们的平稳分布的对应样本集，可以用来做蒙特卡罗模拟求和了。

总结下基于马尔科夫链的采样过程：

2）从任意简单概率分布采样得到初始状态值x_0

3）for $t = 0$ to $n_1 + n_2 - 1$: 从条件概率分布$P(x|x_t)$中采样得到样本x_{t+1}

样本集$(x_{n_1}, x_{n_1+1}, \ldots, x_{n_1+n_2-1})$即为我们需要的平稳分布对应的样本集。

36

2.4 马尔科夫链采样小结

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。

但是一个重要的问题是，随意给定一个平稳分布π,如何得到它所对应的马尔科夫链状态转移矩阵P呢?

这是个大问题。

我们绕了一圈似乎还是没有解决任意概率分布采样样本集的问题。

幸运的是，MCMC采样通过迂回的方式解决了上面这个大问题，下面讨论MCMC的采样，以及它的使用改进版采样: M-H采样和Gibbs采样.

3 MCMC采样和M-H采样

在MCMC马尔科夫链中我们讲到给定一个概率平稳分布π, 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。本节我们就讨论解决这个问题的办法：MCMC采样和它的易用版M-H采样。

3.1 马尔科夫链的细致平稳条件

在解决从平稳分布π, 找到对应的马尔科夫链状态转移矩阵P之前，我们还需要先看看马尔科夫链的细致平稳条件。定义如下：

如果非周期马尔科夫链的状态转移矩阵P和概率分布$\pi(x)$对于所有的i, j满足：

$$\pi(i)P(i,j) = \pi(j)P(j,i)$$

则称概率分布$\pi(x)$是状态转移矩阵P的平稳分布。

证明很简单,由细致平稳条件有：

$$\sum_{i=1}^{\infty}\pi(i)P(i,j) = \sum_{i=1}^{\infty}\pi(j)P(j,i) = \pi(j)\sum_{i=1}^{\infty}P(j,i) = \pi(j)$$

将上式用矩阵表示即为：

$$\pi P = \pi$$

即满足马尔可夫链的收敛性质。也就是说，只要我们找到了可以使概率分布π(x)满足细致平稳分布的矩阵P即可。这给了我们寻找从平稳分布π, 找到对应的马尔科夫链状态转移矩阵P的新思路。

不过不幸的是，仅仅从细致平稳条件还是很难找到合适的矩阵P。比如我们的目标平稳分布是π(x), 随机找一个马尔科夫链状态转移矩阵Q,它是很难满足细致平稳条件的，即：

$$\pi(i)Q(i,j) \neq \pi(j)Q(j,i)$$

那么如何使这个等式满足呢？下面我们来看MCMC采样如何解决这个问题。

由于一般情况下，目标平稳分布π(x)和某一个马尔科夫链状态转移矩阵Q不满足细致平稳条件，即

$$\pi(i)Q(i,j) \neq \pi(j)Q(j,i)$$

36

我们可以对上式做一个改造，使细致平稳条件成立。方法是引入一个α(i,j),使上式可以取等号，即：

$$\pi(i)Q(i,j)\alpha(i,j) = \pi(j)Q(j,i)\alpha(j,i)$$

问题是什么样的α(i,j)可以使等式成立呢？其实很简单，只要满足下两式即可：

$$\alpha(i,j) = \pi(j)Q(j,i)$$
$$\alpha(j,i) = \pi(i)Q(i,j)$$

这样，就得到了分布π(x)对应的马尔科夫链状态转移矩阵P，满足：

$$P(i,j) = Q(i,j)\alpha(i,j)$$

也就是说，我们的目标矩阵P可以通过任意一个马尔科夫链状态转移矩阵Q乘以α(i,j)得到。

α(i,j)我们有一般称之为接受率。取值在[0,1]之间，可以理解为一个概率值。即目标矩阵P可以通过任意一个马尔科夫链状态转移矩阵Q以一定的接受率获得。这个很像接受-拒绝采样，那里是以一个常用分布通过一定的接受-拒绝概率得到一个非常见分布，这里是以一个常见的马尔科夫链状态转移矩阵Q通过一定的接受-拒绝概率得到目标转移矩阵P,两者的解决问题思路是类似的。

总结下MCMC的采样过程。

1）输入我们任意选定的马尔科夫链状态转移矩阵Q，平稳分布$\pi(x)$，设定状态转移次数阈值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 $t = 0$ to $n_1 + n_2 - 1$:

a) 从条件概率分布$Q(x|x_t)$中采样得到样本x_*

b) 从均匀分布采样$u \sim uniform[0,1]$

c) 如果$u < \alpha(x_t, x_*) = \pi(x_*)Q(x_*, x_t)$，则接受转移$x_t \rightarrow x_*$，即$x_{t+1} = x_*$

d) 否则不接受转移，$t = max(t-1, 0)$

样本集$(x_{n_1}, x_{n_1+1}, \ldots, x_{n_1+n_2-1})$即为我们需要的平稳分布对应的样本集。

上面这个过程基本上就是MCMC采样的完整采样理论了，但是这个采样算法还是比较难在实际中应用，为什么呢？问题在上面第三步的c步骤，接受率这儿。由于 $\alpha(x_t, x_*)$ 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。

3.3 M-H采样

M-H采样是Metropolis-Hastings采样的简称，这个算法首先由Metropolis提出，被Hastings改进，因此被称之为Metropolis-Hastings采样或M-H采样

M-H采样解决了我们上一节MCMC采样接受率过低的问题。

$$\pi(i)Q(i,j)\alpha(i,j) = \pi(j)Q(j,i)\alpha(j,i)$$

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

$$\pi(i)Q(i,j) \times 0.1 = \pi(j)Q(j,i) \times 0.2$$

36

这时我们可以看到，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：

$$\pi(i)Q(i,j) \times 0.5 = \pi(j)Q(j,i) \times 1$$

这样接受率可以做如下改进，即：

$$\alpha(i,j) = min\{\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)}, 1\}$$

通过这个微小的改造，我们就得到了可以在实际应用中使用的M-H采样算法过程如下：

1）输入我们任意选定的马尔科夫链状态转移矩阵Q，平稳分布$\pi(x)$，设定状态转移次数阈值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 $t = 0$ to $n_1 + n_2 - 1$:

a) 从条件概率分布$Q(x|x_t)$中采样得到样本x_*

b) 从均匀分布采样$u \sim uniform[0,1]$

c) 如果$u < \alpha(x_t, x_*) = min\{\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)}, 1\}$，则接受转移$x_t \rightarrow x_*$，即$x_{t+1} = x_*$

d) 否则不接受转移，$t = max(t-1, 0)$

样本集$(x_{n_1}, x_{n_1+1}, \ldots, x_{n_1+n_2-1})$即为我们需要的平稳分布对应的样本集。

很多时候，我们选择的马尔科夫链状态转移矩阵Q如果是对称的，即满足$Q(i,j) = Q(j,i)$，这时我们的接受率可以进一步简化为：

$$\alpha(i,j) = min\{\frac{\pi(j)}{\pi(i)}, 1\}$$

3.4 M-H采样实例

为了更容易理解，这里给出一个M-H采样的实例。

在例子里，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。这个例子仅仅用来让大家加深对M-H采样过程的理解。

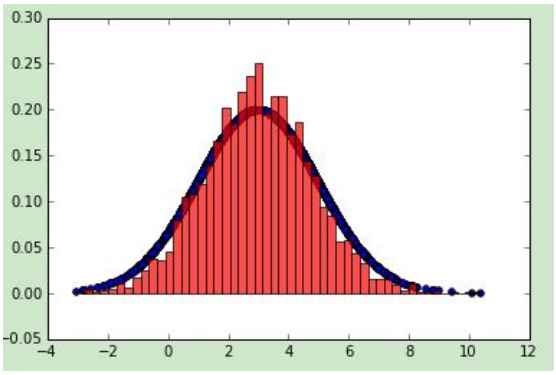
代码如下：

```
from scipy.stats import norm
import matplotlib.pyplot as plt
%matplotlib inline

def norm_dist_prob(theta):
    y = norm.pdf(theta, loc=3, scale=2)
    return y

T = 5000
pi = [0 for i in range(T)]
sigma = 1
t = 0
while t < T-1:
    t = t + 1
    pi_star = norm.rvs(loc=pi[t - 1], scale=sigma, size=1, random_state=None)
    alpha = min(1, (norm_dist_prob(pi_star[0]) / norm_dist_prob(pi[t - 1])))

    u = random.uniform(0, 1)
    if u < alpha:
        pi[t] = pi_star[0]
    else:
        pi[t] = pi[t - 1]
```

36

```
plt.scatter(pi, norm.pdf(pi, loc=3, scale=2))
num_bins = 50
plt.hist(pi, num_bins, normed=1, facecolor='red', alpha=0.7)
plt.show()
```

输出的图中可以看到采样值的分布与真实的分布之间的关系如下，采样集还是比较拟合对应分布的。



3.5 M-H采样总结

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：

1） 我们的数据特征非常的多，M-H采样由于接受率计算式$\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)}$的存在，在高维时需要的计算时间非常的可观，算法效率很低。同时$\alpha(i,j)$一般小于1，有时候辛苦计算出来却被拒绝了。能不能做到不拒绝转移呢？

2） 由于特征维度大，很多时候我们甚至很难求出目标的各特征维度联合分布，但是可以方便求出各个特征之间的条件概率分布。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采样解决了上面两个问题，因此在大数据时代，MCMC采样基本是Gibbs采样的天下。

4 MCMC:Gibbs采样

但是M-H采样有两个缺点：

一是需要计算接受率，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。

二是有些高维数据，特征的条件概率分布好求，但是特征的联合分布不好求。

36 因此需要一个好的方法来改进M-H采样，这就是我们下面讲到的Gibbs采样。

4.1 重新寻找合适的细致平稳条件

细致平稳条件：如果非周期马尔科夫链的状态转移矩阵P和概率分布π(x)对于所有的i,j满足：

$$\pi(i)P(i,j)=\pi(j)P(j,i)$$

则称概率分布π(x)是状态转移矩阵P的平稳分布。

在M-H采样中我们通过引入接受率使细致平稳条件满足。现在我们换一个思路。

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

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

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

由于两式的右边相等，因此我们有：

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

也就是：

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

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$\pi(x_2|x_1^{(1)})$作为马尔科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！这真是一个开心的发现，同样的道理，在在$x_2=x_2^{(1)}$这条直线上，如果用条件概率分布$\pi(x_1|x_2^{(1)})$作为马尔科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：

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

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：

$$P(A\rightarrow B)=\pi(x_2^{(B)}|x_1^{(1)})\ \ if\ x_1^{(A)}=x_1^{(B)}=x_1^{(1)}$$

$$P(A\rightarrow C)=\pi(x_1^{(C)}|x_2^{(1)})\ \ if\ x_2^{(A)}=x_2^{(C)}=x_2^{(1)}$$

$$P(A\rightarrow D)=0\ \ else$$

有了上面这个状态转移矩阵，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：

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

4.2 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：

2）随机初始化初始状态值x_1和x_2

3）for $t = 0$ to $n_1 + n_2 - 1$:

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

b) 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}

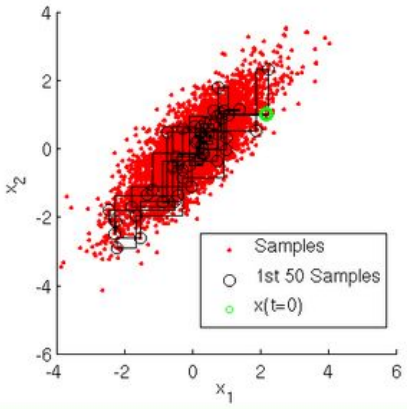
样本集$\{(x_1^{(n_1)}, x_2^{(n_1)}), (x_1^{(n_1+1)}, x_2^{(n_1+1)}), \ldots, (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)})\}$即为我们需要的平稳分布对应的样本集。

整个采样过程中，我们通过轮换坐标轴，采样的过程为：

$$(x_1^{(1)}, x_2^{(1)}) \to (x_1^{(1)}, x_2^{(2)}) \to (x_1^{(2)}, x_2^{(2)}) \to \ldots \to (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)})$$

36

用下图可以很直观的看出，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。



4.3 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 $\pi(x_1, x_2, \ldots, x_n)$ ，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 $P(x_i|x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ ，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：

2）随机初始化初始状态值$(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)})$

3）for $t = 0$ to $n_1 + n_2 - 1$:

a) 从条件概率分布$P(x_1|x_2^{(t)}, x_3^{(t)}, \ldots, x_n^{(t)})$中采样得到样本$x_1^{t+1}$

b) 从条件概率分布$P(x_2|x_1^{(t+1)}, x_3^{(t)}, x_4^{(t)}, \ldots, x_n^{(t)})$中采样得到样本$x_2^{t+1}$

c)...

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

e)...

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

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

36

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。

同样的，轮换坐标轴不是必须的，我们可以随机选择某一个坐标轴进行状态转移，只不过常用的Gibbs采样的实现都是基于坐标轴轮换的。

4.4 二维Gibbs采样实例

这里给出一个Gibbs采样的例子。假设我们要采样的是一个二维正态分布 $Norm(\mu, \Sigma)$ ，其中：

$$\mu = (\mu_1, \mu_2) = (5, -1)$$

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}$$

而采样过程中的需要的状态转移条件分布为：

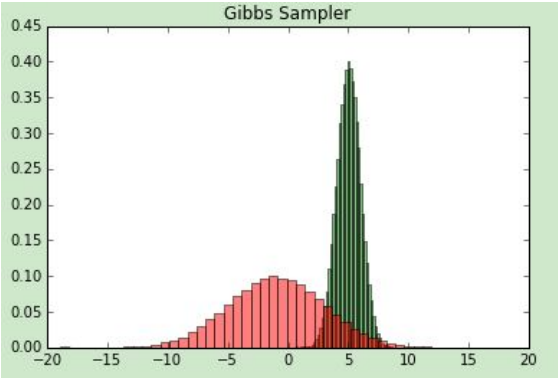
$$P(x_1|x_2) = Norm\left(\mu_1 + \rho\sigma_1/\sigma_2\left(x_2 - \mu_2\right), (1-\rho^2)\sigma_1^2\right)$$
$$P(x_2|x_1) = Norm\left(\mu_2 + \rho\sigma_2/\sigma_1\left(x_1 - \mu_1\right), (1-\rho^2)\sigma_2^2\right)$$

具体的代码如下：

```
samplesource = multivariate_normal(mean=[5,-1], cov=[[1,0.5],[0.5,2]])

def p_ygivenx(x, m1, m2, s1, s2):
    return (random.normalvariate(m2 + rho * s2 / s1 * (x - m1), math.sqrt(1 - rho **
2) * s2))

def p_xgiveny(y, m1, m2, s1, s2):
    return (random.normalvariate(m1 + rho * s1 / s2 * (y - m2), math.sqrt(1 - rho **
2) * s1))

N = 5000
K = 20
x_res = []
y_res = []
z_res = []
m1 = 5
m2 = -1
s1 = 1
s2 = 2

rho = 0.5
y = m2
```

```
for i in xrange(N):
    for j in xrange(K):
        x = p_xgiveny(y, m1, m2, s1, s2)
        y = p_ygivenx(x, m1, m2, s1, s2)
        z = samplesource.pdf([x,y])
        x_res.append(x)
        y_res.append(y)
        z_res.append(z)

num_bins = 50
plt.hist(x_res, num_bins, normed=1, facecolor='green', alpha=0.5)
plt.hist(y_res, num_bins, normed=1, facecolor='red', alpha=0.5)
plt.title('Histogram')
plt.show()
```

输出的两个特征各自的分布如下：

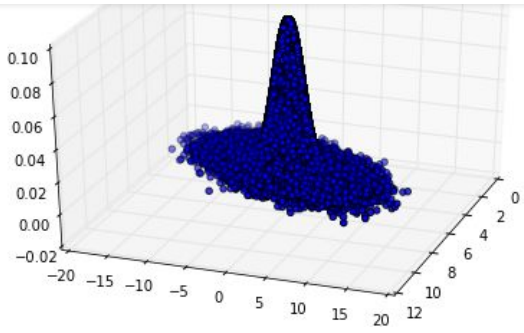


然后我们看看样本集生成的二维正态分布，代码如下：

```
fig = plt.figure()
ax = Axes3D(fig, rect=[0, 0, 1, 1], elev=30, azim=20)
ax.scatter(x_res, y_res, z_res,marker='o')
plt.show()
```

输出的正态分布图如下：

36



4.5 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。

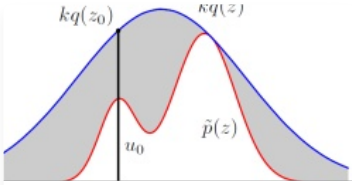
当然Gibbs采样是从M-H采样的基础上的进化而来的，同时Gibbs采样要求数据至少有两个维度，一维概率分布的采样是没法用Gibbs采样的,这时M-H采样仍然成立。

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。

编辑于 2017-10-12

MCMC采样

推荐阅读



Topic Model Series [3]: 采样有什么难的?

斤木 发表于自然语言居...

浅谈「Gibbs采样」

[原理] Gibbs采样Gibbs采样是一种特殊的马尔可夫链算法，常被用于解决包括矩阵分解、张量分解等在内的一系列问题，也被称为交替条件采样（alternating conditional sampling），其中，“交...

Xinyu Chen

MCMC-Gibbs Sampling

在Metropolis-Hastings里面提到，遇到多元分布的时候，选取初始值非常困难，在每一个维度上都必须恰到好处。而Gibbs Sampling则是专门解决此类问题的算法，它对每一维进行同时的单独算法迭...

青蛙君

MCM

叮咚小

5 条评论 ⇌ 切换为时间排序

写下你的评论...

不开心 4 个月前

这么好的答案居然没早点看到

赞

沈陰 3 个月前

赞……讲的很细

赞