**MCMC算法整理**

MCMC详细介绍：<http://www.cnblogs.com/ywl925/archive/2013/06/05/3118875.html>

<http://www.52nlp.cn/lda-math-mcmc-%E5%92%8C-gibbs-sampling2>

1. 为什么要用随机模拟？

X 表示随机变量，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，当抽样次数足够的时候，就非常接近真实值了。估计值的精度与 x 的维度无关（虽然维度越高，但是每次抽样获得的信息也越多），而是与抽样次数有关。

2.背景知识

统计模拟中有一个重要的问题就是给定一个概率分布p(x)，我们如何在计算机中生成它的样本。一般而言均匀分布 Uniform(0,1)的样本是相对容易生成的。 通过线性同余发生器可以生成伪随机数，我们用确定性算法生成[0,1]之间的伪随机数序列后，这些序列的各种统计指标和均匀分布 Uniform(0,1) 的理论计算结果非常接近。这样的伪随机序列就有比较好的统计性质，可以被当成真实的随机数使用。

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x0出发沿着马氏链转移, 得到一个转移序列 x0,x1,x2,⋯xn,xn+1⋯,， 如果马氏链在第n步已经收敛了，于是我们就得到了 π(x) 的样本xn,xn+1⋯。

如果非周期马氏链的转移矩阵P和分布π(x) 满足：

π(i)\*Pij = π(j)\*Pji

则 π(x) 是马氏链的平稳分布，上式被称为细致平稳条件(detailed balance condition)。

3. 算法详解

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→j)), 显然，通常情况下

p(i)q(i,j)≠p(j)q(j,i)

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 α(i, j), 我们希望

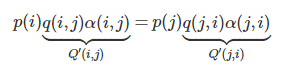
p(i) q(i,j) α(i,j) = p(j) q(j,i) α(j,i) (∗)

就是说可以通过一个系数使得方程向着我们期望的方向演化。

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

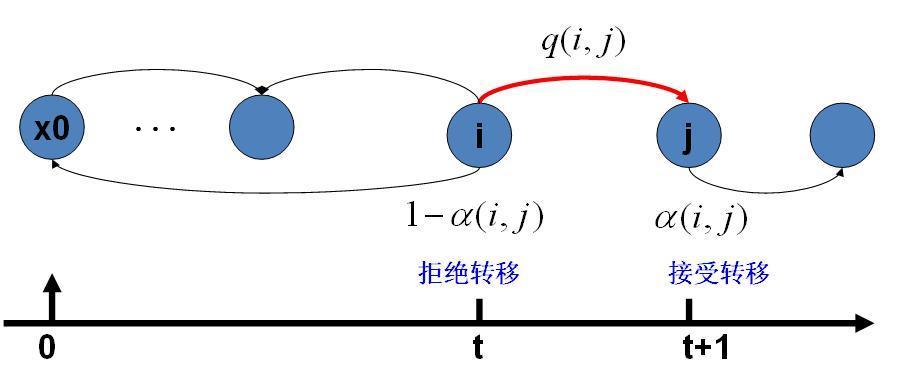
α(i,j)=p(j)q(j,i)，α(j,i)=p(i)q(i,j)

于是(\*)式就成立了，所以有：

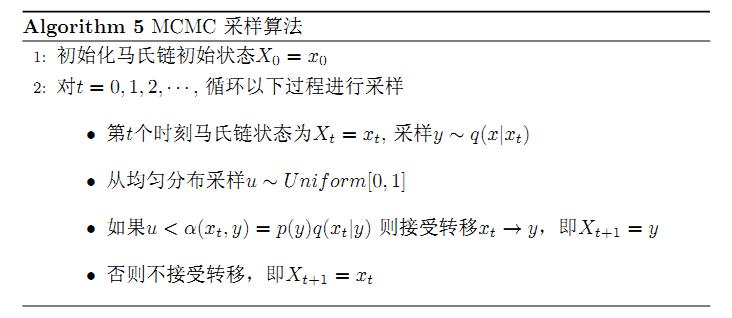


于是我们把原来具有转移矩阵Q的一个很普通的马氏链，改造为了具有转移矩阵Q′的马氏链，而 Q′恰好满足细致平稳条件，由此马氏链Q′的平稳分布就是p(x)！

在改造 Q 的过程中引入的 α(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α(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)\*α(i,j)，如果不接受的话第t+1个状态还是继承第t个状态。



把上面的过程整理成算法:



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。

以上的 MCMC 采样算法已经能工作了，不过有一个小的问题：马氏链Q在转移的过程中的接受率 α(i,j) 可能偏小，这样采样过程中马氏链容易原地踏步，拒绝大量的跳转，这使得马氏链遍历所有的状态空间要花费太长的时间，收敛到平稳分布p(x)的速度太慢。有没有办法提升一些接受率呢?

假设 α(i,j) = 0.1, α(j,i) = 0.2, 此时满足细致平稳条件，于是

p(i)q(i,j)×0.1 = p(j)q(j,i)×0.2

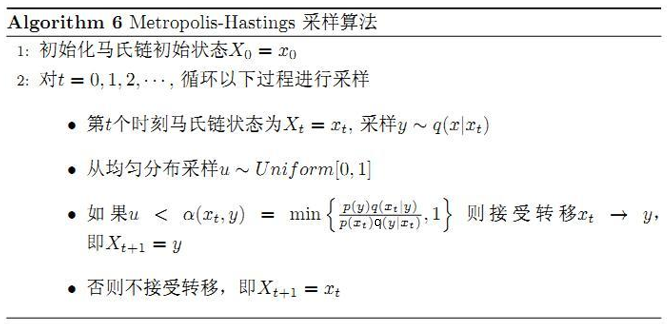
将上式两边扩大5倍，则改写为

p(i)q(i,j)×0.5 = p(j)q(j,i)×1

我们提高了接受率，而细致平稳条件并没有打破。这启发我们可以把细致平稳条件式中的α(i,j)和α(j,i) 同比例放大，使得两数中最大的一个放大到1，这样我们就提高了采样中的跳转接受率。所以我们可以取

α(i,j) = min{1，p(j)×q(j,i)/p(i)×q(i,j)}

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：



对于分布 p(x),我们构造转移矩阵 Q′ 使其满足细致平稳条件

p(x)Q′(x→y) = p(y)Q′(y→x)

此处 x 和y并不要求是一维的，可以使向量。对于高维空间的 p(x)，如果满足上述细致平稳条件，那么以上的 Metropolis-Hastings 算法一样有效。