Bin的专栏

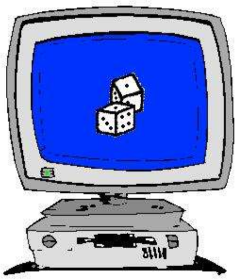
随机采样方法整理与讲解（MCMC、Gibbs Sampling等）

本文是对参考资料中多篇关于sampling的内容进行总结+搬运，方便以后自己翻阅。其实参考资料中的资料写的比我好，大家可以看一下！好东西多分享！PRML的第11章也是sampling，有时间后面写到PRML的笔记中去：）

背景

随机模拟也可以叫做蒙特卡罗模拟(Monte Carlo Simulation)。这个方法的发展始于20世纪40年代，和原子弹制造的曼哈顿计划密切相关，当时的几个大牛，包括乌拉姆、冯.诺依曼、费米、费曼、Nicholas Metropolis， 在美国洛斯阿拉莫斯国家实验室研究裂变物质的中子连锁反应的时候，开始使用统计模拟的方法，并在最早的计算机上进行编程实现。[3]

随机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。



下面总结这么几点：

1、蒙特卡洛数值积分

2、均匀分布，Box-Muller 变换

3、Monte Carlo principle

4、接受-拒绝抽样（Acceptance-Rejection sampling)

5、重要性抽样(Importance sampling)

6、马尔科夫链，马尔科夫稳态

7、MCMC——Metropolis-Hasting算法

8、MCMC——Gibbs Sampling算法

1、蒙特卡洛数值积分

如果我们要求f(x)的积分，如

而f(x)的形式比较复杂积分不好求，则可以通过数值解法来求近似的结果。常用的方法是蒙特卡洛积分：

$$\int_a^b \frac{f(x)}{q(x)} q(x)dx$$

公告

昵称：Bin的专栏
园龄：6年5个月
粉丝：214
关注：2
+加关注

导航

博客园
首页
新随笔
联系
订阅 XML
管理

< 2018年5月 >

日	一	二	三	四	五	六
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

统计

随笔 - 29
文章 - 0
评论 - 61
引用 - 0

搜索

找找看
谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Machine Learning(14)
PRML(6)
前沿动态-技术收录(3)
论文阅读笔记(2)
matlab(2)
计算机科学(1)
算法(1)
信息检索技术(1)

随笔分类

All Articles(29)
Deep Learning
Matlab(2)
Research Notes(1)

这样把q(x)看做是x在区间内的概率分布，而把前面的分数部门看做一个函数，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）。下面我们将介绍常用的抽样方法。

2、均匀分布，Box-Muller 变换

在计算机中生成[0,1]之间的伪随机数序列，就可以看成是一种均匀分布。而随机数生成方法有很多，最简单的如：

$$x_{n+1} = (ax_n + c) \mod m,$$

当然计算机产生的随机数都是伪随机数，不过一般也就够用了。

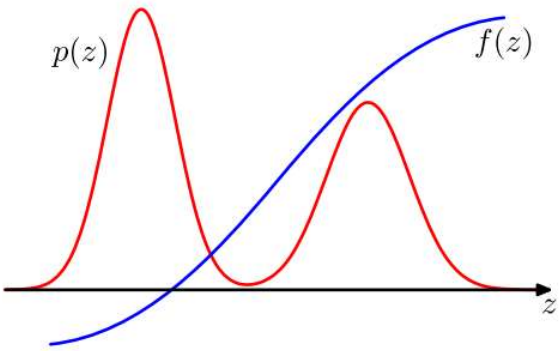
[Box-Muller 变换] 如果随机变量 U1,U2 独立且U1,U2~Uniform[0,1]，

$$Z_0 = \sqrt{-2\ln U_1}\,cos(2\pi U_2)$$
$$Z_1 = \sqrt{-2\ln U_1}\,sin(2\pi U_2)$$

则 Z0,Z1 独立且服从标准正态分布。

3、Monte Carlo principle

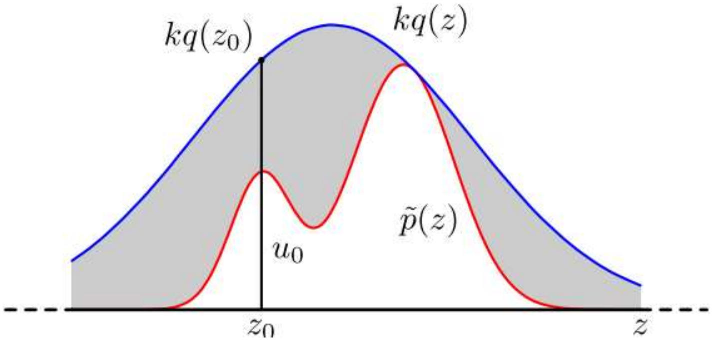
Monte Carlo 抽样计算随即变量的期望值是接下来内容的重点：X 表示随即变量，服从概率分布 p(x), 那么要计算 f(x) 的期望，只需要我们不停从 p(x) 中抽样xi，然后对这些f（xi）取平均即可近似f(x)的期望。



$$\mathbf{E_N}(f) = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}),$$

4、接受-拒绝抽样（Acceptance-Rejection sampling)[2]

很多实际问题中，p(x)是很难直接采样的的，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 k，使得 p(x) 总在 kq(x) 的下方。（参考上图）

- x 轴方向：从 q(x) 分布抽样得到 a。(如果是高斯，就用之前说过的 tricky and faster 的算法更快）
- y 轴方向：从均匀分布（0, kq(a)) 中抽样得到 u。
- 如果刚好落到灰色区域： u > p(a), 拒绝， 否则接受这次抽样
- 重复以上过程

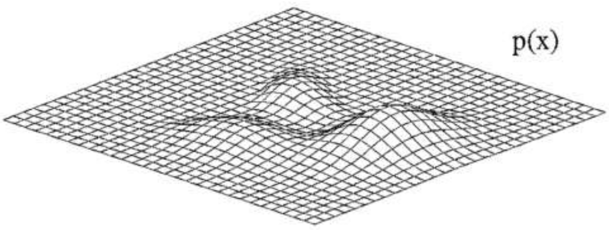
在高维的情况下，Rejection Sampling 会出现两个问题，第一是合适的 q 分布比较难以找到，第二是很难确定一个合理的 k 值。这两个问题会导致拒绝率很高，无用计算增加。

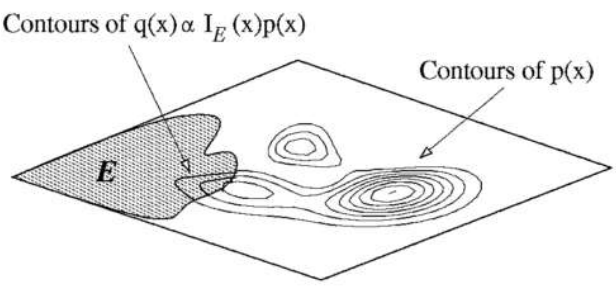
5、重要性抽样(Importance sampling)[2]

Importance Sampling 也是借助了容易抽样的分布 q (proposal distribution)来解决这个问题，直接从公式出发：

$$\begin{aligned}\mathbb{E}[f] &= \int f(\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &= \int f(\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}f(\mathbf{z}^{(l)}).\end{aligned}$$

其中，p(z) / q(z) 可以看做 importance weight。我们来考察一下上面的式子，p 和 f 是确定的，我们要确定的是 q。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。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。在上式中，我们目标是 p×f/q 方差越小越好，所以 |p×f| 大的地方，proposal distribution q(z) 也应该大。举个稍微极端的例子：





第一个图表示 p 分布， 第二个图的阴影区域 f = 1，非阴影区域 f = 0, 那么一个良好的 q 分布应该在左边箭头所指的区域有很高的分布概率，因为在其他区域的采样计算实际上都是无效的。这表明 Importance Sampling 有可能比用原来的 p 分布抽样更加有效。

4. 机器学习降维算法三：LLE (Locally Linear Embedding) 局部线性嵌入(6)
5. 距离计算方法总结(5)

推荐排行榜

1. 随机采样方法整理与讲解（MCMC、Gibbs Sampling等）(11)
2. 稀疏矩阵存储格式总结+存储效率对比:COO,CSR,DIA,ELL,HYB(6)
3. 机器学习降维算法一：PCA (Principal Component Analysis)(4)
4. 今天开始学Pattern Recognition and Machine Learning (PRML)，章节1.1，介绍与多项式曲线拟合(Polynomial Curve Fitting)(4)
5. 机器学习-特征选择 Feature Selection 研究报告(4)

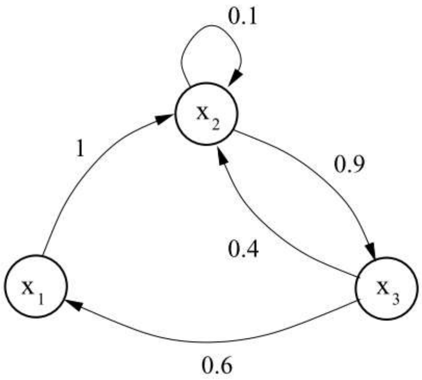
但是可惜的是，在高维空间里找到一个这样合适的 q 非常难。即使有 Adaptive importance sampling 和 Sampling-Importance-Resampling(SIR) 的出现，要找到一个同时满足 easy to sample 并且 good approximations 的 proposal distribution, it is often impossible!

6、马尔科夫链，马尔科夫稳态

在讲蒙特卡洛方法之前，必须要先讲一下马尔科夫链；马氏链的数学定义：

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

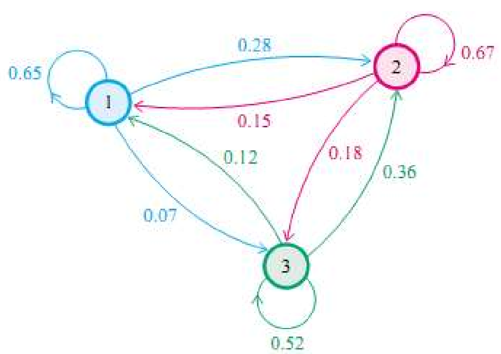
也就是说前一个状态只与当前状态有关，而与其他状态无关，Markov Chain 体现的是状态空间的转换关系，下一个状态只决定与当前的状态(可以联想网页爬虫原理，根据当前页面的超链接访问下一个网页)。如下图：



举一个例子，如果当前状态为 u(x) = (0.5, 0.2, 0.3), 那么下一个矩阵的状态就是 u(x)T = (0.18, 0.64, 0.18), 依照这个转换矩阵一直转换下去，最后的系统就趋近于一个稳定状态 (0.22, 0.41, 0.37) (此处只保留了两位有效数字)。而事实证明无论你从那个点出发，经过很长的 Markov Chain 之后都会汇集到这一点。[2]

再举一个例子，社会学家经常把人按其经济状况分成3类：下层(lower-class)、中层(middle-class)、上层(upper-class)，我们用1,2,3 分别代表这三个阶层。社会学家们发现决定一个人的收入阶层的最重要的因素就是其父母的收入阶层。如果一个人的收入属于下层类别，那么他的孩子属于下层收入的概率是 0.65, 属于中层收入的概率是 0.28, 属于上层收入的概率是 0.07。事实上，从父代到子代，收入阶层的变化的转移概率如下

		子代		
	State	1	2	3
父代	1	0.65	0.28	0.07
	2	0.15	0.67	0.18
	3	0.12	0.36	0.52



使用矩阵的表示方式，转移概率矩阵记为

$$P=\begin{bmatrix}0.65 & 0.28 & 0.07\\ 0.15 & 0.67 & 0.18\\ 0.12 & 0.36 & 0.52\end{bmatrix}$$

假设当前这一代人处在下层、中层、上层的人的比例是概率分布向量 $\pi_0 = [\pi_0(1), \pi_0(2), \pi_0(3)]$，那么他们的子女的分布比例将是 $\pi_1 = \pi_0 P$，他们的孙子代的分布比例将是 $\pi_2 = \pi_1 P = \pi_0 P^2$，……，第$n$代子孙的收入分布比例将是 $\pi_n = \pi_{n-1} P = \pi_0 P^n$。

假设初始概率分布为$\pi_0 = [0.21, 0.68, 0.11]$，则我们可以计算前$n$代人的分布状况如下

第n代人	下层	中层	上层
0	0.210	0.680	0.110
1	0.252	0.554	0.194
2	0.270	0.512	0.218
3	0.278	0.497	0.225
4	0.282	0.490	0.226
5	0.285	0.489	0.225
6	0.286	0.489	0.225
7	0.286	0.489	0.225
8	0.289	0.488	0.225
9	0.286	0.489	0.225
10	0.286	0.489	0.225
...	...	...	...

我们发现从第7代人开始，这个分布就稳定不变了，事实上，在这个问题中，从任意初始概率分布开始都会收敛到这个上面这个稳定的结果。

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 $\lim_{n\to\infty} P_{ij}^n$ 存在且与i无关，记 $\lim_{n\to\infty} P_{ij}^n = \pi(j)$, 我们有

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

1. $\pi(j) = \sum_{i=0}^{\infty} \pi(i) P_{ij}$
2. π 是方程 $\pi P = \pi$ 的唯一非负解

其中,

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

π称为马氏链的平稳分布。

注：要求图是联通的（没有孤立点），同时不存在一个联通的子图是没有对外的出边的（就像黑洞一样）。

这个马氏链的收敛定理非常重要，所有的 MCMC(Markov Chain Monte Carlo) 方法都是以这个定理作为理论基础的。

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⋯。

这个绝妙的想法在1953年被 Metropolis想到了，为了研究粒子系统的平稳性质， Metropolis 考虑了物理学中常见的波尔兹曼分布的采样问题，首次提出了基于马氏链的蒙特卡罗方法，即Metropolis算法，并在最早的计算机上编程实现。Metropolis 算法是首个普适的采样方法，并启发了一系列 MCMC方法，所以人们把它视为随

机模拟技术腾飞的起点。 Metropolis的这篇论文被收录在《统计学中的重大突破》中， Metropolis算法也被遴选为二十世纪的十个最重要的算法之一。

我们接下来介绍的MCMC 算法是 Metropolis 算法的一个改进变种，即常用的 Metropolis-Hastings 算法。由上一节的例子和定理我们看到了，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。

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

$$\pi(i)P_{ij} = \pi(j)P_{ji} \qquad \text{for all} \quad i,j \tag{1}$$

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

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

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

由于π 是方程 $\pi P = \pi$的解，所以π是平稳分布。

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$q(i \to j)$), 显然，通常情况下

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

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 $\alpha(i,j)$, 我们希望

$$p(i)q(i,j)\alpha(i,j) = p(j)q(j,i)\alpha(j,i) \quad (*) \tag{2}$$

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

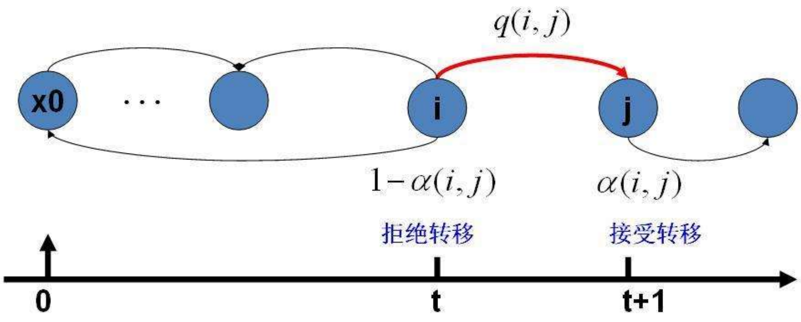
$$\alpha(i,j) = p(j)q(j,i)\,, \quad \alpha(j,i) = p(i)q(i,j)$$

于是(*)式就成立了。所以有

$$p(i)\underbrace{q(i,j)\alpha(i,j)}_{Q'(i,j)} = p(j)\underbrace{q(j,i)\alpha(j,i)}_{Q'(j,i)} \quad (**) \tag{3}$$

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$q(i,j)\alpha(i,j)$。



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。

Algorithm 5 MCMC 采样算法

1: 初始化马氏链初始状态$X_0 = x_0$

2: 对$t = 0, 1, 2, \cdots$, 循环以下过程进行采样

- 第t个时刻马氏链状态为$X_t = x_t$, 采样$y \sim q(x|x_t)$
- 从均匀分布采样$u \sim Uniform[0, 1]$
- 如果$u < \alpha(x_t, y) = p(y)q(x_t|y)$ 则接受转移$x_t \rightarrow y$，即$X_{t+1} = y$
- 否则不接受转移，即$X_{t+1} = x_t$

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 采样算法已经能很漂亮的工作了，不过它有一个小的问题：马氏链Q在转移的过程中的接受率 $\alpha(i,j)$ 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 $\alpha(i,j) = 0.1, \alpha(j,i) = 0.2$, 此时满足细致平稳条件，于是

$$p(i)q(i,j) \times 0.1 = p(j)q(j,i) \times 0.2$$

上式两边扩大5倍，我们改写为

$$p(i)q(i,j) \times 0.5 = p(j)q(j,i) \times 1$$

看，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$\alpha(i,j), \alpha(j,i)$ 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

$$\boxed{\alpha(i,j) = \min\left\{\frac{p(j)q(j,i)}{p(i)q(i,j)}, 1\right\}}$$

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。

Algorithm 6 Metropolis-Hastings 采样算法

1: 初始化马氏链初始状态$X_0 = x_0$
2: 对$t = 0, 1, 2, \cdots$, 循环以下过程进行采样

- 第t个时刻马氏链状态为$X_t = x_t$, 采样$y \sim q(x|x_t)$
- 从均匀分布采样$u \sim Uniform[0, 1]$
- 如果$u < \alpha(x_t, y) = \min\left\{\frac{p(y)q(x_t|y)}{p(x_t)p(y|x_t)}, 1\right\}$ 则接受转移$x_t \to y$, 即$X_{t+1} = y$
- 否则不接受转移，即$X_{t+1} = x_t$

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

$$p(x)Q'(x \to y) = p(y)Q'(y \to x)$$

此处 x 并不要求是一维的，对于高维空间的 $p(\mathbf{x})$，如果满足细致平稳条件

$$p(\mathbf{x})Q'(\mathbf{x} \to \mathbf{y}) = p(\mathbf{y})Q'(\mathbf{y} \to \mathbf{x})$$

那么以上的 Metropolis-Hastings 算法一样有效。

8、MCMC——Gibbs Sampling算法

对于高维的情形，由于接受率 α的存在(通常 $\alpha < 1$), 以上 Metropolis-Hastings 算法的效率不够高。能否找到一个转移矩阵Q使得接受率 $\alpha = 1$ 呢？我们先看看二维的情形，假设有一个概率分布 $p(x, y)$, 考察x坐标相同的两个点$A(x_1, y_1), B(x_1, y_2)$，我们发现

$$p(x_1, y_1)p(y_2|x_1) = p(x_1)p(y_1|x_1)p(y_2|x_1)$$
$$p(x_1, y_2)p(y_1|x_1) = p(x_1)p(y_2|x_1)p(y_1|x_1)$$

所以得到

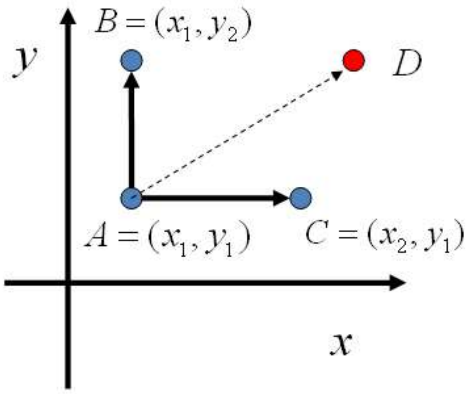
$$p(x_1, y_1)p(y_2|x_1) = p(x_1, y_2)p(y_1|x_1) \quad (***) \tag{4}$$

即

$$p(A)p(y_2|x_1) = p(B)p(y_1|x_1)$$

基于以上等式，我们发现，在 $x = x_1$ 这条平行于 y轴的直线上，如果使用条件分布 $p(y|x_1)$做为任何两个点之间的转移概率，那么任何两个点之间的转移满足细致平稳条件。同样的，如果我们在 $y = y_1$ 这条直线上任意取两个点 $A(x_1, y_1), C(x_2, y_1)$,也有如下等式

$$p(A)p(x_2|y_1) = p(C)p(x_1|y_1).$$



平面上马氏链转移矩阵的构造

于是我们可以如下构造平面上任意两点之间的转移概率矩阵Q

$$
\begin{aligned}
Q(A\to B) &= p(y_B|x_1) \quad \text{如果} \quad x_A = x_B = x_1 \\
Q(A\to C) &= p(x_C|y_1) \quad \text{如果} \quad y_A = y_C = y_1 \\
Q(A\to D) &= 0 \qquad\qquad\quad \text{其它}
\end{aligned}
$$

有了如上的转移矩阵 Q, 我们很容易验证对平面上任意两点 X, Y, 满足细致平稳条件

$$p(X)Q(X\to Y) = p(Y)Q(Y\to X)$$

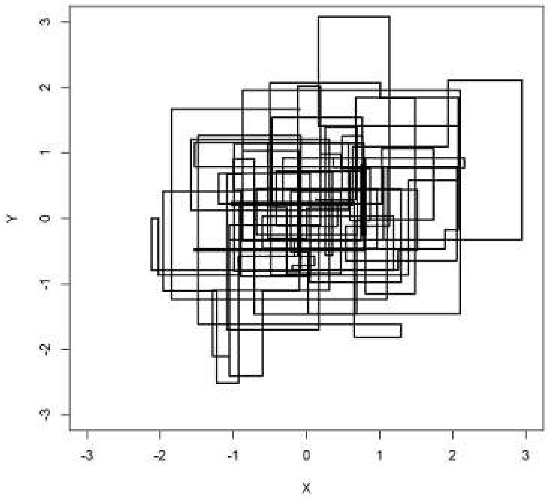
于是这个二维空间上的马氏链将收敛到平稳分布 $p(x,y)$。而这个算法就称为 Gibbs Sampling 算法,是 Stuart Geman 和Donald Geman 这两兄弟于1984年提出来的，之所以叫做Gibbs Sampling 是因为他们研究了Gibbs random field, 这个算法在现代贝叶斯分析中占据重要位置。

Algorithm 7 二维Gibbs Sampling 算法

1: 随机初始化$X_0 = x_0 Y_0 = y_0$

2: 对$t = 0, 1, 2, \cdots$ 循环采样

1. $y_{t+1} \sim p(y|x_t)$
2. $x_{t+1} \sim p(x|y_{t+1})$



Gibbs Sampling 算法中的马氏链转移

以上采样过程中，如图所示，马氏链的转移只是轮换的沿着坐标轴 x轴和y轴做转移，于是得到样本 $(x_0,y_0),(x_0,y_1),(x_1,y_1),(x_1,y_2),(x_2,y_2),\cdots$ 马氏链收敛后，最终得到的样本就是 $p(x,y)$ 的样本，而收敛之前的阶段称为 burn-in period。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。最一般的情形可以是，在t时刻，可以在x轴和y轴之间随机的选一个坐标轴，然后按条件概率做转移，马氏链也是一样收敛的。轮换两个坐标轴只是一种方便的形式。

以上的过程我们很容易推广到高维的情形，对于(***) 式，如果x_1 变为多维情形$\mathbf{x_1}$，可以看出推导过程不变，所以细致平稳条件同样是成立的

$$p(\mathbf{x_1},y_1)p(y_2|\mathbf{x_1}) = p(\mathbf{x_1},y_2)p(y_1|\mathbf{x_1}) \qquad (5)$$

此时转移矩阵 Q 由条件分布 $p(y|\mathbf{x_1})$ 定义。上式只是说明了一根坐标轴的情形，和二维情形类似，很容易验证对所有坐标轴都有类似的结论。所以n维空间中对于概率分布 $p(x_1,x_2,\cdots,x_n)$ 可以如下定义转移矩阵

1. 如果当前状态为$(x_1,x_2,\cdots,x_n)$，马氏链转移的过程中，只能沿着坐标轴做转移。沿着 x_i 这根坐标轴做转移的时候，转移概率由条件概率 $p(x_i|x_1,\cdots,x_{i-1},x_{i+1},\cdots,x_n)$ 定义；
2. 其它无法沿着单根坐标轴进行的跳转，转移概率都设置为 0。

于是我们可以把Gibbs Smapling 算法从采样二维的 $p(x,y)$ 推广到采样n 维的 $p(x_1,x_2,\cdots,x_n)$

Algorithm 8 n维Gibbs Sampling 算法

1: 随机初始化$\{x_i : i = 1,\cdots,n\}$

2: 对$t = 0,1,2,\cdots$ 循环采样

1. $x_1^{(t+1)} \sim p(x_1|x_2^{(t)},x_3^{(t)},\cdots,x_n^{(t)})$
2. $x_2^{(t+1)} \sim p(x_2|x_1^{(t+1)},x_3^{(t)},\cdots,x_n^{(t)})$
3. $\cdots$
4. $x_j^{(t+1)} \sim p(x_j|x_1^{(t+1)},\cdots,x_{j-1}^{(t+1)},x_{j+1}^{(t)},\cdots,x_n^{(t)})$
5. $\cdots$
6. $x_n^{(t+1)} \sim p(x_n|x_1^{(t+1)},x_2^{t},\cdots,x_{n-1}^{(t+1)})$

以上算法收敛后，得到的就是概率分布p(x1,x2,⋯,xn)的样本，当然这些样本并不独立，但是我们此处要求的是采样得到的样本符合给定的概率分布，并不要求独立。同样的，在以上算法中，坐标轴轮换采样不是必须的，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。

参考资料

[1] http://blog.csdn.net/xianlingmao/article/details/7768833

[2] http://www.cnblogs.com/daniel-D/p/3388724.html

[3] http://cos.name/2013/01/lda-math-mcmc-and-gibbs-sampling/

[4] An Introduction to MCMC for Machine Learning，2003

[5] Introduction to Monte Carlo Methods

分类: All Articles,机器学习 Machine Learning

标签: Machine Learning

« 上一篇：今天开始学Pattern Recognition and Machine Learning (PRML)，章节5.2-5.3，Neural Networks神经网络训练（BP算法）
» 下一篇：机器学习距离公式总结

posted on 2015-02-01 21:46 Bin的专栏 阅读(42026) 评论(11) 编辑 收藏

评论

#1楼 2015-02-02 11:15 AlanWang

灰常的高大上，虽然看不进去，但也要顶楼主一下

支持(0) 反对(0)

#2楼 2015-09-25 21:13 beggar_gangs

很厉害，不过自己有的地方理解起来有一些模糊

支持(0) 反对(0)

#3楼 2016-03-08 16:45 eulerwang

找到的最完整最清晰的mcmc解释文章了。

支持(0) 反对(0)

#4楼 2016-04-02 17:25 想学马尔科夫

大神，假如我要生成正态分布的抽样样本，那么算法6里面采样y~q（x|xt）怎么求啊，分布q（x|xt）就是正态分布本身嘛?

支持(0) 反对(0)

#5楼 2016-04-14 10:07 burkun

@ 想学马尔科夫
xt是上一轮留下来的采样，可以用 X~N(xt, delta)来生成q(x|xt)的样本。xt可以取为均值，是否可以取成标准差，有待考证

支持(0) 反对(0)

#6楼 2016-05-05 16:55 yunhe

作者讲的清晰易懂，条理清楚。非常感谢！我去，感觉这些大牛们太厉害了！！

支持(0) 反对(0)

#7楼 2016-09-01 13:03 dangerman

问下，您后面讲MCMC算法的各种文字截图是出自哪本书？多谢

支持(0) 反对(0)

#8楼 2016-09-14 13:17 leolotus

写得真好，一般教科书都不会有Gibbs抽样和MH抽样的详细讲解

支持(0) 反对(0)

#9楼 2016-09-14 13:19 leolotus

@ burkun
取成均值就是jump distribution为random walk的MH算法

支持(0) 反对(0)

#10楼 2016-11-08 12:03 姜楠

部分公式有问题，希望博主能补上

支持(0) 反对(0)

#11楼 2017-07-05 10:07 幻灭的星星

清晰易懂，厉害

支持(0) 反对(0)

最新IT新闻:
· 小游戏正在毁灭微信群聊
· 摩拜终于免押金了，但可能还是没有你的份
· Windows 10版微软翻译加入新功能：支持小娜和手写笔
· 保洁阿姨拿600万融资公司？抱歉，靠童话融不到资
· 滴滴：司机已经和被打乘客张先生和解 双方互相道歉
» 更多新闻...

最新知识库文章:
· 菜鸟工程师的超神之路 -- 从校园到职场
· 如何识别人的技术能力和水平？
· 写给自学者的入门指南
· 和程序员谈恋爱
· 学会学习
» 更多知识库文章...