

saltriver的专栏

You're On Your Own.

目录视图

摘要视图

RSS 订阅

个人资料



saltriver

关注

发私信



访问：200854次

积分：2274

等级：BLOG 5

排名：第18742名

原创：95篇 转载：0篇

译文：2篇 评论：26条

文章搜索

Q

文章分类

- Python (23)
- 数学与算法 (22)
- 机器学习 (22)
- 图像处理与OpenCV (14)
- 数据结构与算法 (10)
- 生活随笔 (6)

文章存档

图灵赠书——程序员11月书单

【思考】Python这么厉害的原因竟然是！

感恩节赠书：《深度学习》等异步社区优秀图书和作者评选启动！

每周荐书：京东架构、Linux内核、Python全栈

蒙特卡洛方法到底有什么用

标签：蒙特卡洛

2016-08-12 20:07

5867人阅读

评论(1)

收藏

举报

分类：

数学与算法 (21)

版权声明：无需授权，可任意转载。




蒙特卡洛方法（Monte Carlo method，也有翻译成“蒙特卡罗方法”）是以概率和统计的理论、方法为基础的一种数值计算方法，将所求解的问题同一定的概率模型相联系，用计算机实现统计模拟或抽样，以获得问题的近似解，故又称随机抽样法或统计试验法。上述就是蒙特卡洛方法的基本概念，比较抽象，下面结合实际工作中的理解，谈一谈对蒙特卡洛方法的一些认识。

（1）首先，蒙特卡洛不是个人名，而是个地名，说明该方法与概率有着密切的关联。

蒙特卡洛方法的提出者是大名鼎鼎的数学家冯·诺伊曼，搞计算机的不可能不知道他（计算机之父），冯·诺伊曼在20世纪40年代中期用驰名世界的赌城—摩纳哥的蒙特卡洛来命名这种方法。（大家也别把蒙特卡洛当一个城市，估计和北京的一条街差不了多少，因为摩纳哥（不是非洲的摩洛哥）本身就是个袖珍国家，比我国澳门都小的多）。说明该方法与赌博中的随机性、概率性有着天然而密切的联系。几乎涉及到复杂的、与概率相关的数值计算的领域都有可能用到。比如计算物理、经济金融、统计学、机器学习等。

（2）蒙特卡洛没有什么高深的理论，它只是一种方法或者说策略。


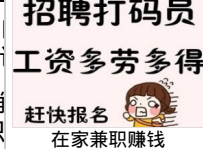
蒙特卡洛方法并没有什么高深的理论支撑，如果一定要说有理论也就只有概率论或统计学中的大数定律了。蒙特卡洛的基本原理简单描述是先大量模拟，然后计算一个事件发生的次数，再通过这个发生次数除以总模拟次数，得到想要的结果。比如投3个骰子，计算3个骰子同时是6的概率，可以模拟投N次（随机样本数），统计同时是6出现的次数C，然后C除以N即是计算结果。



人脸识别

迪拜十星级酒店

（3）蒙特卡洛方法可以应用在很多场合，但求情况下，越接近与真实值，但样本数增加会带来



招聘打码员

工资多劳多得

赶快报名

在家兼职赚钱

移民澳大利亚

蒙特卡洛方法不仅仅是算概率哦，再看一个稍区间的积分，即求如下图所示的红色区域的面积

2017年12月 (12)
2017年10月 (3)
2017年09月 (2)
2017年08月 (4)
2017年07月 (4)

展开

阅读排行

泊松分布的期望和方差推导	(42525)
Python的定时器	(20036)
二项分布均值和方差的简单推导	(14448)
指数分布的期望和方差推导	(13080)
信息熵到底是什么	(10661)
图论（一）基本概念	(6833)
求约束条件下极值的拉格朗日...	(6799)
Python中如何修改字符串的值	(6448)
蒙特卡洛方法到底有什么用	(5861)
K最近邻算法（KNN）	(5717)

评论排行

Python的定时器	(6)
线性回归损失函数为什么要用...	(4)
二项分布均值和方差的简单推导	(3)
Python几个简单好用的基础功能	(2)
指数分布族	(2)
科比罚球，武器命中和霍金劫数	(2)
逻辑回归为什么使用对数损失...	(2)
逻辑回归是个什么逻辑	(1)
泊松分布的期望和方差推导	(1)
Sigmoid函数	(1)

推荐文章

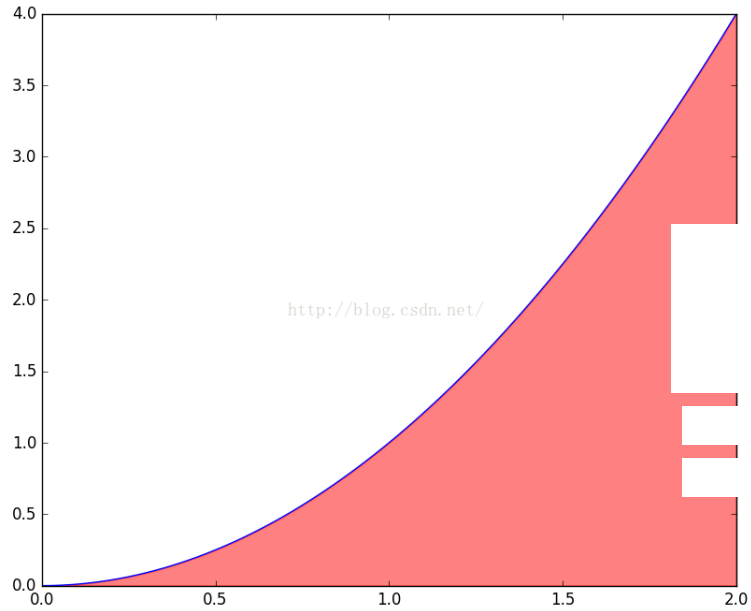
- * 【2017年11月27日】CSDN博客更新周报
- * 【CSDN】邀请您来GitChat赚钱啦！
- * 【GitChat】精选——JavaScript进阶指南
- * 改做人工智能之前，90%的人都没能给自己定位
- * TensorFlow 人脸识别网络与对抗网络搭建
- * Vue 移动端项目生产环境优化
- * 面试必考的计算机网络知识点梳理

最新评论

蒙特卡洛方法到底有什么用
weiwensong : 讲的不错

Sigmoid函数
KookoBaby : 伯努利分布变形式中的 $\log(1-p)$

算更简单精确，这里主要是举例说明下蒙特卡洛方法的使用过程。



绘图代码如下：

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 2, 1000)
y = x ** 2
plt.plot(x, y)
plt.fill_between(x, y, where=(y > 0), color='red', alpha=0.5)
plt.show()
```

该红色区域在一个 2×4 的正方形里面。使用蒙特卡洛方法，随机在这个正方形里面产生大量随机点（数量为 N ），计算有多少点（数量为 count ）落在红色区域内（判断条件为 $y < x^2$ ）， count/N 就是所要求的积分值，也即红色区域的面积。

1.模拟1000个随机点：

$NN = 1000$

```
points = [[xy[0] * 2, xy[1] * 4] for xy in np.random.rand(N, 2)]
plt.scatter([x[0] for x in points], [x[1] for x in points], s=5, c=np.random.rand(N, 1) < 0.5)
plt.show()
```

关闭

)应该改为 $\ln(1-p)$ 吧？

Python的定时器

hack0072005 :@njutli:同问，这样是不是不合理

Python的定时器

我要笑熬浆糊 :谢谢分享~

Python的定时器

我要笑熬浆糊 :@njutli:不好意思==我又测试了一下，我是在IDLE里测试的这段代码；我把pr...

Python的定时器

我要笑熬浆糊 :@njutli:我测试了一下，把那些时间参数都改成0.5s,虽然内存占用空间上升不是很快，但是内存空...

图论（十）最小生成树-Prim算法

现在问题来了_WA_划水 :文章很不错，言简意赅

线性回归损失函数为什么要用平方形式

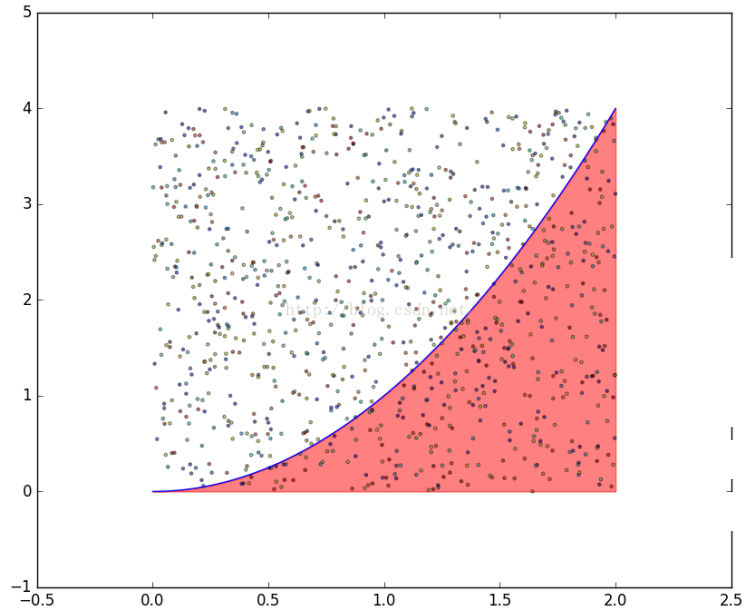
bihanren32 :good!!!

逻辑回归是个什么逻辑

anqinghu :多谢分享，非常详细。

泊松分布的期望和方差推导

Full_Speed_Turbo :赞



2.计算落在红色区域的比重：

```
count = 0
for xy in points:
    if xvf11 < xy[0] ** 2:
        count += 1
print((count / N) * (2 * 4))
```

输出结果：

2.832

这与精确值（2.666666）的差距只有6.2%，而对于更大规模的模拟， $N=100$ 万，输出结果为：2.66528，这与精确值的差距只有0.051975%（万分之五）。可以看出，蒙特卡洛方法有一定的误差，误差的大小与模拟的样本大小直接相关，模拟样本越大，误差越小，但计算量也会大幅上升。

（4）对于简单问题来说，蒙特卡洛是个“笨”办法。但对许多问题来说，它往往是个有效，有时甚至是唯一可行的方法。

对于上面的简单问题，蒙特卡洛方法就显得有点“笨”了。直接用数值积分运算更简单，更精确。

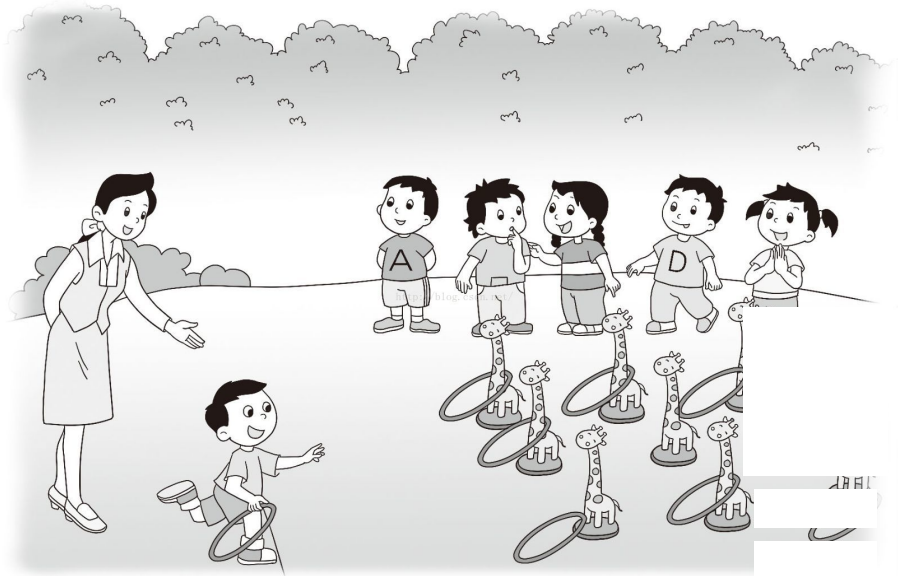
```
print(scipy.integrate.quad(lambda x: x ** 2, 0, 2)[0])
```

输出结果：

2.666666666666667

但对于涉及不可解析函数或概率分布的模拟及计算，蒙特卡洛方法是个有效的力法。

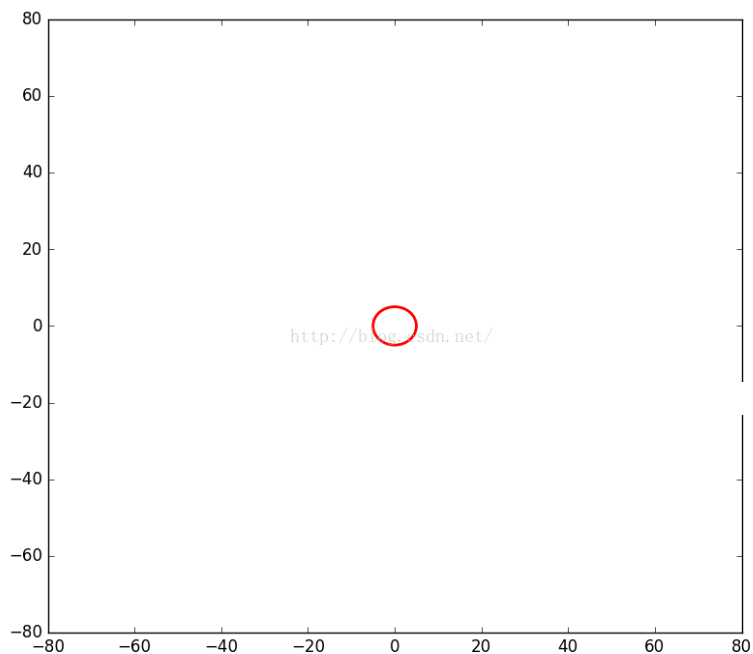
关闭



我们都玩过套圈圈的游戏，想过为什么你总是套不上吗？用蒙特卡洛方法来算一算。

1. 设物品中心点坐标为 $(0,0)$ ，物品半径为 5cm 。

```
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import numpy as np
circle_target = mpatches.Circle([0, 0], radius=5, edgecolor='r', fill=False)
plt.xlim(-80, 80)
plt.ylim(-80, 80)
plt.axes().add_patch(circle_target)
plt.show()
```



关闭

2. 设投圈半径 8cm ，投圈中心点围绕物品中心点呈二维正态分布，均值 $\mu=0\text{cm}$ ，

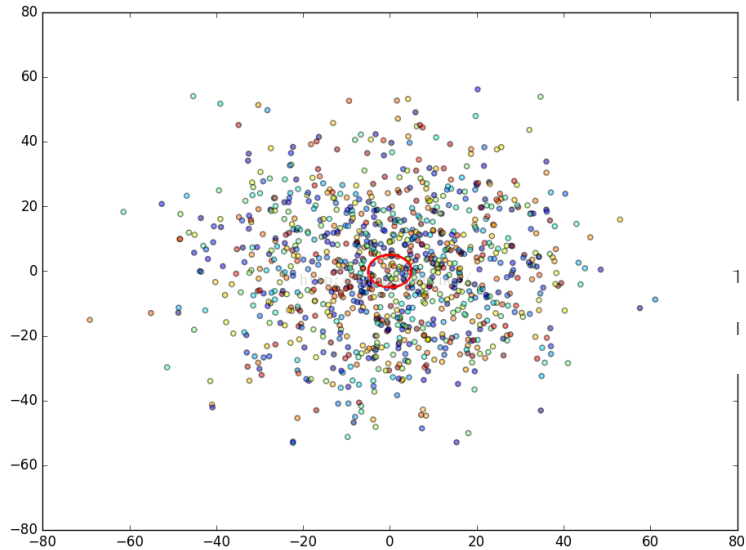
标准差 $\sigma=20\text{cm}$ ，模拟1000次投圈过程。

$N = 1000$

$u, \text{sigma} = 0, 20$

$\text{points} = \text{sigma} * \text{np.random.randn}(N, 2) + u$

$\text{plt.scatter}([x[0] \text{ for } x \text{ in points}], [x[1] \text{ for } x \text{ in points}], c=\text{np.random.rand}(N), \text{alpha}=0.5)$



上图中红圈为物品，散点图为模拟1000次投圈过程中，投圈中心点的位置散布。

3. 计算1000次投圈过程中，投圈套住物品的占比情况。

```
print(len([xy for xy in points if xy[0]**2 + xy[1]**2 < (8-5)**2]) / N)
```

输出结果：0.014，即投1000次，有14次能够套住物品，就是个小概率事件，知道你为什么套不住了吧。

(5) 蒙特卡洛方法本身不是优化方法，与遗传算法、粒子群等优化算法有着本质的区别。

蒙特卡洛方法与遗传算法、粒子群算法等智能优化算法有相似之处，比如都属于随机近似方法，都不能保证得到最优解等，但它们也有着本质的差别。一是层次不一样，蒙特卡洛只能称之为方法，遗传算法等则属于仿生智能算法，比蒙特卡洛方法要复杂。二是应用领域不同，蒙特卡洛是一种模拟统计方法，如果问题可以描述成某种统计量的形式，那么就可以用蒙特卡洛方法来解决；遗传算法等则适用于大规模的组优化问题（选址问题、排班问题、管理调度、路线优化）等，以及复杂函数求最值、参数优化等。

顶 踩

4

0

关闭

- 上一篇 [Python的定时器](#)
- 下一篇 [Python的操作符重载](#)

相关文章推荐



kuka机器人



加速度传感器



免费的云主机



人脸识别



客户管理系统

查看评论



weiwensong

讲的不错

1楼 2017-12-25 21:10发表

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司
京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved 

关闭

第6页 共6页

2018/1/2 上午10:33