

我是8位的

I am 8 bits, what about you?

随笔 - 205, 文章 - 0, 评论 - 103, 阅读 - 101万

导航

博客园
首页
新随笔
联系
订阅
管理

2022年3月						
日	一	二	三	四	五	六
27	28	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

公告

你的支持是我的动力
欢迎关注微信公众号“我是8位的”



昵称：我是8位的
园龄：4年7个月
粉丝：288
关注：5
+加关注

盖楼抽奖

#她的梦想在发光#

HWD科技女性故事有奖征集

分享最打动你的科技女性故事

活动时间: 2022年3月8日-3月18日

马上参与

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

积分与排名

积分 - 457097
排名 - 1198

概率统计18——再看大数定律

在对不了解概率的人解释期望时，我总是敷衍地将期望解释为均值。这种敷衍的说法之所以行得通，正是由于大数定律起了作用。

人们在实践中发现，尽管每个随机变量的取值不同，但当随机变量大量出现时，它们的均值却相对恒定，这个规律就是大数定律。

一个公平的骰子

我们有一个公平的骰子，每个点数出现的概率都是1/6，如果只投掷一次，完全无法预测它的点数，但是如果把连续投掷20次看作一次试验，却发现每次试验的点数的均值总是在3.5附近徘徊。每次试验投掷的次数越多，点数的均值越稳定。

```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(10, 5))
plt.subplots_adjust(hspace=0.5, wspace=0.3)

m = 100 # 试验的次数
for i in range(1, 5):
    n = 2 * 10 ** i # 每次试验投掷n次骰子
    mean_list = [np.random.randint(1, 7, n).mean() for i in range(1,
ax = fig.add_subplot(2, 2, i)
ax.plot(list(range(1, m + 1)), mean_list, label='均值')
plt.xticks(list(range(0, 7))) # 重置y轴的坐标
ax.set_xlabel('试验次数')
ax.set_ylabel('均值')
ax.set_title('每次试验投{}次骰子'.format(n))
ax.legend(loc='upper right')

plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
plt.show()
```

随笔分类 (211)

- ★★资源下载★★(1)
- Java并发编程(1)
- 程序员的数学(24)
- 单变量微积分(31)
- 多变量微积分(24)
- 概率(24)
- 机器学习(27)
- 软件设计(1)
- 数据分析(6)
- 数据结构与算法(27)
- 随笔(5)
- 线性代数(34)
- 项目管理(2)
- 转载(4)

随笔档案 (205)

- 2021年2月(1)
- 2020年3月(2)
- 2020年2月(6)
- 2020年1月(4)
- 2019年12月(7)
- 2019年11月(15)
- 2019年9月(3)
- 2019年8月(6)
- 2019年7月(1)
- 2019年6月(8)
- 2019年5月(3)
- 2019年4月(5)
- 2019年3月(7)
- 2019年2月(3)
- 2019年1月(7)
- 更多

阅读排行榜

- 1. 使用Apriori进行关联分析 (一) (29768)
- 2. 线性代数笔记12——列空间和零空间 (28772)
- 3. FP-growth算法发现频繁项集 (一)——构建FP树(24430)
- 4. 寻找“最好” (2) ——欧拉-拉格朗日方程(23099)
- 5. 多变量微积分笔记3——二元函数的极值(22772)

评论排行榜

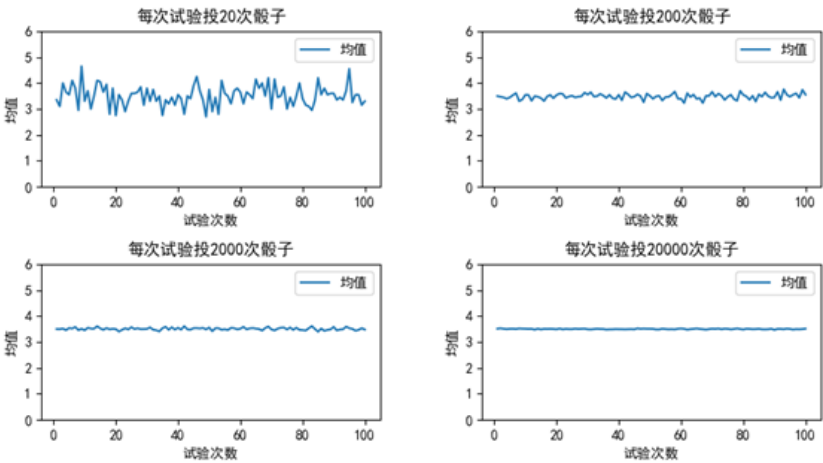
- 1. 隐马尔可夫模型 (一) (8)
- 2. 线性代数笔记12——列空间和零空间 (7)
- 3. 线性代数笔记3——向量2 (点积) (7)
- 4. FP-growth算法发现频繁项集 (一)——构建FP树(5)
- 5. 寻找“最好” (2) ——欧拉-拉格朗日方程(4)

推荐排行榜

- 1. 寻找“最好” (2) ——欧拉-拉格朗日方程(7)
- 2. FP-growth算法发现频繁项集 (一)——构建FP树(7)
- 3. 线性代数笔记3——向量2 (点积) (6)
- 4. FP-growth算法发现频繁项集 (二)——发现频繁项集(5)
- 5. 隐马尔可夫模型 (一) (5)

最新评论

- 1. Re:线性代数笔记3——向量2 (点积)
如果点积小于0, 即夹角小于90°, 这个写错了吧。应该是夹角大于90°
--猫猫猫猫大人



均值的期望和均值的方差

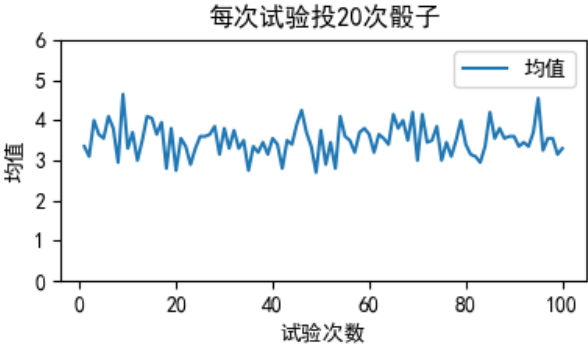
我们用随机变量 X_1 表示第一次投骰子的结果, X_2 表示第二次....., 一个公平的骰子可以得到下面的结论:

$$\begin{aligned}P(X_1 = 1) &= P(X_2 = 1) = \cdots = P(X_n = 1) = \frac{1}{6} \\P(X_1 = 2) &= P(X_2 = 2) = \cdots = P(X_n = 2) = \frac{1}{6} \\&\vdots \\P(X_1 = 6) &= P(X_2 = 6) = \cdots = P(X_n = 6) = \frac{1}{6}\end{aligned}$$

连续投掷后将形成一个随机变量序列 $\{X_1, X_2, \dots, X_n\}$, 这个序列的均值是:

$$\bar{X} = \frac{1}{n}(X_1 + X_2 + \cdots + X_n)$$

序列中的每个变量都是随机的, 因此它们的和及均值也是随机的, 也就是说 $\bar{X}$ 然是随机的。以 $n=20$ 为例:



而均值的期望则不同:

$$\begin{aligned}E[\bar{X}] &= E\left[\frac{1}{n}(X_1 + X_2 + \cdots + X_n)\right] \\&= \frac{1}{n}E[X_1 + X_2 + \cdots + X_n] \\&= \frac{1}{n}(E[X_1] + E[X_2] + \cdots + E[X_n])\end{aligned}$$

每次掷骰子的结果都是独立同分布的, 它们有相同的期望和方差。用 μ 和 σ^2 表示随机变量的期望和方差, 于是我们得到了均值的期望:

$$E[\bar{X}] = \frac{1}{n}(E[X_1] + E[X_2] + \cdots + E[X_n]) = \frac{n\mu}{n} = \mu$$

2. Re:线性代数笔记10——矩阵的LU分解写的很好，不过LU分解的前提是错的，LU分解只需要第三个条件，如果允许行置换就是下面写到的PLU，可以分解所有矩阵

--wiki3D

3. Re:单变量微积分笔记20——三角替换1 (sin和cos) 很nice

--尹保棕

4. Re:线性代数笔记24——微分方程和exp(At)

有些图片挂了呢

--ccchendada

5. Re:寻找“最好” (2) ——欧拉-拉格朗日方程

提个issue，最速降线中

$v = \{2gh\}^{1/2}$ 与配图不一致，建议以起点为原点，向右伸出x轴，向下伸出y轴建立坐标系

--trustInU

均值的期望等于整体的期望，是一个定值。对于公平的骰子来说，整体的期望与整体的均值相等。

均值的方差是：

$$\begin{aligned} \text{Var}(\bar{X}) &= \text{Var}\left(\frac{1}{n}(X_1 + X_2 + \dots + X_n)\right) \\ &= \frac{1}{n^2} \text{Var}(X_1 + X_2 + \dots + X_n) \\ &= \frac{1}{n^2} (\text{Var}(X_1) + \text{Var}(X_2) + \dots + \text{Var}(X_n)) \\ &= \frac{n\sigma^2}{n^2} \\ &= \frac{\sigma^2}{n} \end{aligned}$$

辛钦大数定律和伯努利大数定律

n 个样本均值的方差是总体方差的 $1/n$ ，随着 n 的增大，该方差也越来越小，逐渐趋近于0。方差为0表示随机变量 $\bar{X}$ 相对于 $E[\bar{X}]$ 的波动0，即 $\bar{X}$ 不含随机性。换句话说，随着 n 的增大，样本的均值逐渐收敛于 μ ，这就是所谓的大数定律。这回有意思了，随着 n 的增大，作为均值的随机变量居然逐渐变得不含随机性。

这里的大数定律也称为辛钦大数定律或弱大数定律。抄一下教科书上的定义：

设随机变量 $X_1, X_2, \dots, X_n, \dots$ 相互独立，服从同一分布且具有数学期望 $E(X_k) = \mu (k=1, 2, \dots)$ ，则序列 $\bar{X} = \frac{1}{n} \sum_{k=1}^n X_k$ 依概率收敛于 μ ，即 $\bar{X} \xrightarrow{P} \mu$ 。

样本均值的标准差是 $\sigma/\sqrt{n}$ ，标准差表示随机变量与总体之间的差异度。想要把差异度缩减10倍，那么 n 的数量要增加 10^2 倍。

伯努利大数定律是辛钦大数定律的一个重要推论。

设 f_A 是 n 次独立重复试验中事件 A 发生的次数， p 是事件 A 在每次试验中发生的概率，则对于任意正数 $\varepsilon > 0$ ，有：

$$\lim_{n \rightarrow \infty} P\left\{\left|\frac{f_A}{n} - p\right| < \varepsilon\right\} = 1 \quad \text{或} \quad \lim_{n \rightarrow \infty} P\left\{\left|\frac{f_A}{n} - p\right| \geq \varepsilon\right\} = 0 \quad ①$$

别被定义唬住了，它想解释的问题很简单：当 n 足够大时， f_A/n 与 p 的差值趋近于0，也就是频率趋近于概率，这也是我们用比例估计作为点估计量的基础。在实际问题中，当试验次数很大时，便可以用事件的频率来代替事件的概率。

其实辛钦大数定律也可以用①的方式表示：

$$\lim_{n \rightarrow \infty} P(|\bar{X} - \mu| < \varepsilon) = 1 \quad \text{或} \quad \lim_{n \rightarrow \infty} P(|\bar{X} - \mu| \geq \varepsilon) = 0$$

对于概率的符号，有时候有大括号，有时候用小括号，这个还是别太纠结，实际上没什么区别，用大括号只是为了强调事件是一个集合。

不存在的期望

大数定律并不是对所有问题都生效，比如在不期望的情况下。

现在我们把骰子换成一枚公平的硬币，参加一个关于硬币第几次正面朝上的游戏。把随机变量 $X=x$ 看作在第 x 次投硬币时会得到第1次正面朝上的结果：

如果第1次就得到正面朝上的结果 ($X=1$)，得2分

如果第2次才得到正面朝上的结果 ($X=2$)，得4分

如果第3次才得到正面朝上的结果 ($X=3$)，得8分

.....

如果第 n 次才得到正面朝上的结果 ($X=n$)，得 2^n 分

现在我们需要计算得分 $Y=2^X$ 的期望。显然 X 符合几何分布，每个 X 又和 Y 一一对应，因此 Y 出现的概率等同于 X 出现的概率：

x 的值	Y 的值	x 和 Y 值出现的概率
1	2	$1/2$
2	4	$1/4$
3	8	$1/8$
4	16	$1/16$
...	...	...

$$E[Y] = 2 \times \frac{1}{2} + 4 \times \frac{1}{4} + \dots = 1 + 1 + \dots$$

这个期望是发散的，即没法得出具体的数值，也没法求极限， $E[Y]=\infty$ 表示不存在期望。

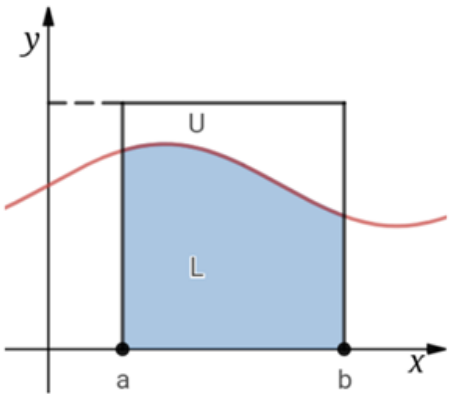
蒙特卡罗积分法

假设我们遇到了一个难以计算的积分：

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

包括换元法、三角替换、分部积分在内的各种方法都无济于事，此时可以求助于暴力的大数定律。

我们知道积分的几何意义是曲线与 x 轴围成的面积：



曲线 $f(x)$ 把矩形分割成两部分， U 是曲线上方的面积， L 是阴影部分的面积， R 是整个矩形的面积，是已知量。现在我们从这个矩形内随机选择一点 $r_i=(x, y)$ ，那么根据几何概型，这个点落入 L 中的概率是 L/R 。既然积分的是

在计算面积 L ，我们就可以借助积分对落在 L 中的某一点的概率进行精确的表达：

$$p = P(r_i \in L) = \frac{L}{R} = \frac{1}{R} \int_a^b f(x) dx$$

上式同时附带得出积分的结果：

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

R 是已知量，只要求得 p ，就能得到积分的结果。于是积分问题变成了概率问题。

现在让每个随机点 r_i 都对应一个随机变量 X_i ， X_i 满足下面的特性：

$$X_i = \begin{cases} 1, & \text{if } r_i \in L \\ 0, & \text{else} \end{cases}$$

每个随机变量的期望是：

$$E[X_i] = 1 \times P(r_i \in L) + 0 \times P(r_i \in U) = p$$

将大数定律应用于 X_i 上，当 $n \rightarrow \infty$ 时， X_i 的均值依概率收敛于 p ，即：

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n X_i \rightarrow p$$

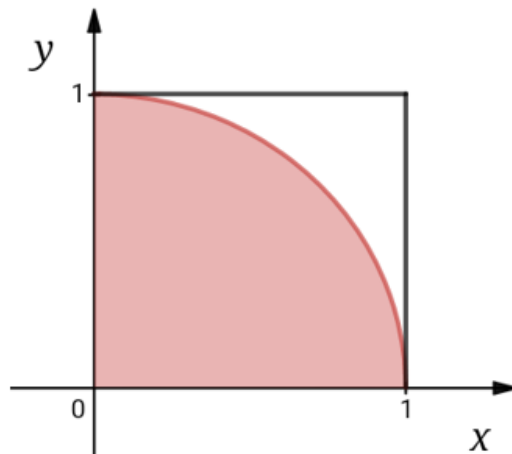
n 是落在矩形中的随机点的数量，如果随机点落在 L 中， $X_i=1$ ，否则 $X_i=0$ ，这个结果实际是告诉我们，当 n 足够大时，落在 L 中的点的概率趋近于落在 L 中点的数量与落在矩形中的总点数的比值。

将这个结果代入积分式：

$$\int_a^b f(x) dx = pR = \frac{R}{n} \sum_{i=1}^n X_i$$

于是我们可以借助物理实验和计算机的模拟求解这个难以对付的积分。

我们用程序模拟蒙特卡罗积分法，计算下面四分之一圆的面积：



圆的曲线是 $x^2 + y^2 = 1$ ，如果一个随机点 (x_i, y_i) 满足 $x^2 + y^2 < 1$ ，则表示该点落在了圆内。



```
import matplotlib.pyplot as plt
import numpy as np

def create_ri(n):
    ''' 成 n 个随机点 '''
    xs, ys = np.random.random_sample(n), np.random.random_sample(n)
    return xs, ys

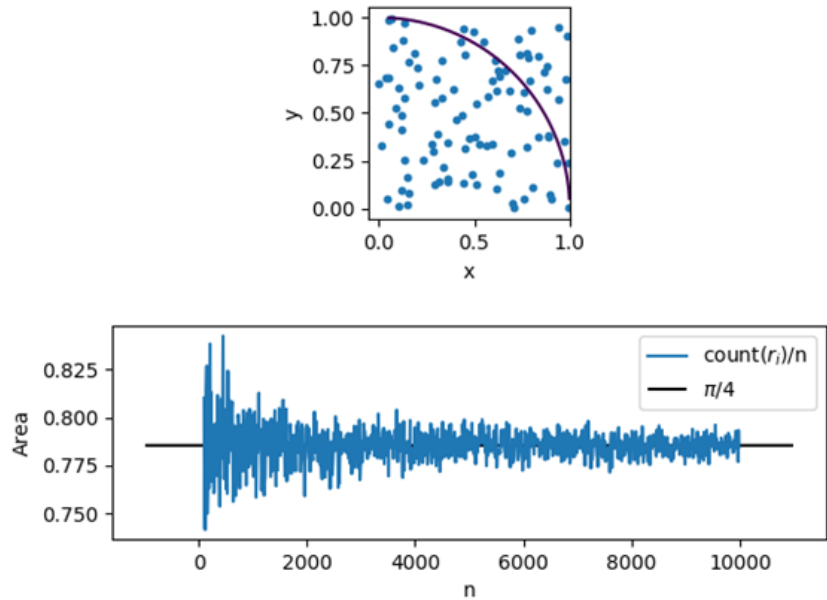
def cnt_circle(xs, ys):
    ''' 圆中随机点的数量 '''
    ps = xs ** 2 + ys ** 2
    return ps[ps < 1].size

fig = plt.figure()
plt.subplots_adjust(hspace=0.5)
ax = fig.add_subplot(2, 1, 1)
x = y = np.arange(0, 1, 0.001)
x, y = np.meshgrid(x, y)
ax.contour(x, y, x**2 + y**2, [1])
xs, ys = create_ri(100)
ax.scatter(xs, ys, s=10)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.axis('scaled')

xl, yl = [], []
n_max = 10000
for n in range(100, n_max, 10):
    xs, ys = create_ri(n)
    cnt = cnt_circle(xs, ys)
    Area = cnt / n
    xl.append(n)
    yl.append(Area)
    n *= 10

ax = fig.add_subplot(2, 1, 2)
ax.hlines(np.pi / 4, xmin=-1000, xmax=n_max + 1000, label='$\pi/4$')
ax.plot(xl, yl, label='count($r_i$)/n')
ax.legend(loc='upper right')
ax.set_xlabel('n')
ax.set_ylabel('Area')
plt.show()
```





随着 n 的增加，蓝色曲线的波动越来越小，计算的结果越来越接近真实值。

正像前面提到的，如果想让计算结果的精度提升10倍，需要增加100倍的试验次数，因此蒙特卡罗法的效率并不高。

尽管大数定律下的蒙特卡罗法很好用，但仍然需要小心谨慎，关键之处在于大数定律没有为我们明确指出到底什么才是“大数”，到底需要多少次试验才能充分接近我们所期待的极限。无论 n 有多大，我们仍然不能否认存在这样的情况：所抛出的硬币全部正面朝上，尽管这种情况发生的概率很小。确定这个概率的方式是中心极限定理的内容。

出处：微信公众号“我是8位的”

本文以学习、研究和分享为主，如需转载，请联系本人，标明作者和出处，非商业用途！

扫描二维码关注作者公众号“我是8位的”



```
ps = xs ** 2 + ys ** 2
return ps[ps < 1].size
```

随笔

分类: 概率

标签: 伯努利大数定律, 蒙特卡罗积分法, 辛钦大数定律

好文要顶

关注我

收藏该文





我是8位的

关注 - 5

粉丝 - 288

+ 加关注

0

推荐

0

反对

« 上一篇: [概率统计17——点估计和连续性修正](#)

» 下一篇: [概率统计19——中心极限定理](#)

posted on 2020-02-12 18:05 我是8位的 阅读(1557) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) [博客园首页](#)

- 【推荐】[华为 HWD 2022 故事征集, 分享最打动你的科技女性故事](#)
- 【推荐】[华为开发者专区, 与开发者一起构建万物互联的智能世界](#)

kuaidaili.com

动态/静态/自动
切换/完全独享
IP

打开 >

编辑推荐:

- 革命性创新, 动画杀手铜 @scroll-timeline
- 戏说领域驱动设计 (十二) —— 服务
- ASP.NET Core 6框架揭秘实例演示[16]: 内存缓存与分布式缓存的使用
- .Net Core 中无处不在的 Async/Await 是如何提升性能的?
- 分布式系统改造方案 —— 老旧系统改造篇

#她的梦想在发光#
HWD科技女性故事有奖征集
活动时间: 2022年3月8日-3月18日



最新新闻:

- 乔布斯的创业搭档: 他缺乏工程师才能, 不得不锻炼营销能力来弥补
 - 美国大厂码农薪资曝光: 年薪18万美元, 够养家, 不够买海景房
 - 两张照片就能转视频! Google提出FLIM帧插值模型
 - Android 再推 “杀手级” 功能, 可回收 60% 存储空间
 - 溺在理财暴雷潮的投资人: 本金63万, 月兑25元不够卖菜
- » 更多新闻...

Powered by:
博客园
Copyright © 2022 我是8位的
Powered by .NET 6 on Kubernetes