

昵称： 光彩照人
园龄： 8年1个月
粉丝： 140
关注： 7
[+加关注](#)

2025年3月						
<	日	一	二	三	四	五
	23	24	25	26	27	28
	2	3	4	5	6	7
	9	10	11	12	13	14
	16	17	18	19	20	21
	23	24	25	26	27	28
	30	31	1	2	3	4

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](#)

随笔分类

[keras\(18\)](#)

[python\(44\)](#)

[pytorch\(11\)](#)

[TensorFlow\(11\)](#)

[比赛学习\(12\)](#)

[大数据平台\(3\)](#)

[对话系统\(3\)](#)

[机器学习\(56\)](#)

[计算广告\(3\)](#)

[架构之美\(5\)](#)

[金融科技\(23\)](#)

[决策树\(树形模型\)\(15\)](#)

[强化学习\(3\)](#)

[设计模式\(9\)](#)

推荐算法之Thompson(汤普森)采样

AI IDE Trae

如果想理解贝叶斯统计推断，就必须先熟悉「解贝塔分布」。

一、Beta(贝塔)分布

Beta分布是一个定义在[0,1]区间上的连续概率分布族，它有两个正值参数，称为形状参数，一般用α和β表示，Beta分布的概率密度函数形式如下：

$$\begin{aligned} f(x; \alpha, \beta) &= \text{constant} \cdot x^{\alpha-1} (1-x)^{\beta-1} \\ &= \frac{x^{\alpha-1} (1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1} (1-u)^{\beta-1} du} \\ &= \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} \\ &= \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} \end{aligned}$$

这里的Γ表示gamma函数。

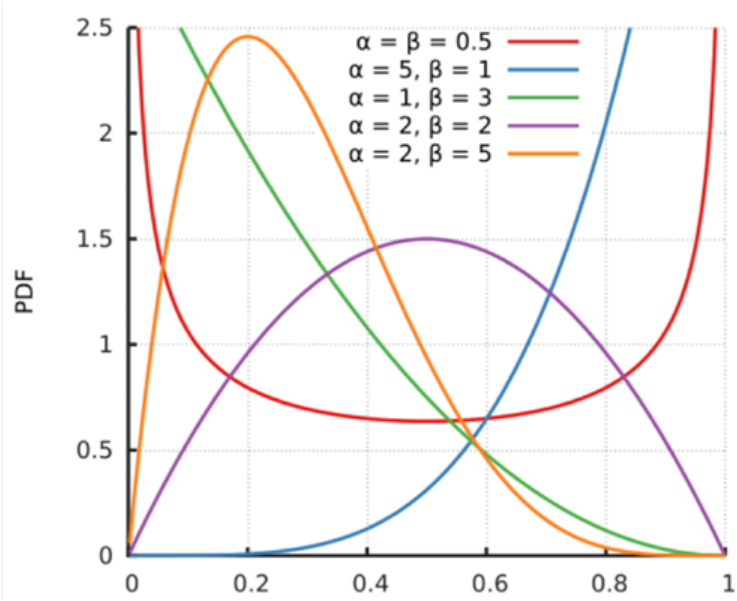
Beta分布的均值是：

$$\frac{\alpha}{\alpha + \beta}$$

方差：

$$\frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

Beta分布的图形(概率密度函数)：



从Beta分布的概率密度函数的图形我们可以看出，Beta分布有很多种形状，但都是在0-1区间内，因此Beta分布可以描述各种0-1区间内的形状（事件）。因此，它特别适合为某件事发生或者成功的概率建模。同时，当α=1，β=1的时候，它就是一个均匀分布。

贝塔分布主要有 α和 β两个参数，这两个参数决定了分布的形状，从上图及其均值和方差的公式可以看出：

摄影(2)
更多

随笔档案

2024年6月(1)
2023年5月(1)
2023年3月(1)
2022年11月(1)
2022年9月(1)
2022年8月(1)
2022年7月(6)
2022年6月(5)
2022年5月(6)
2022年4月(3)
2022年1月(2)
2021年9月(1)
2021年8月(5)
2021年7月(3)
2021年4月(6)

更多

阅读排行榜

1. Python数据可视化–seaborn(104974)
2. 随机森林（Random Forest）详解（转）(56506)
3. Spring Boot中配置文件application.pro perties使用(49347)
4. 图示详解BERT模型的输入与输出(4671 1)
5. 两个重要极限公式(45880)

评论排行榜

1. 拉格朗日对偶理解(5)
2. 图示详解BERT模型的输入与输出(3)
3. 深入理解KS(3)
4. Python数据可视化–seaborn(3)

- 1) $\alpha/(\alpha+\beta)$ 也就是均值，其越大，概率密度分布的中心位置越靠近1，依据此概率分布产生的随机数也多说都靠近1，反之则都靠近0。
- 2) $\alpha+\beta$ 越大，分布的集中度越高，这样产生的随机数更接近中心位置，从方差公式上也能看出。

二、举例理解Beta分布

贝塔分布可以看作是一个概率的分布，当我们不知道一个东西的具体概率是多少时，它给出了所有概率出现的可能性大小，可以理解为概率的概率分布。

以棒球为例子：

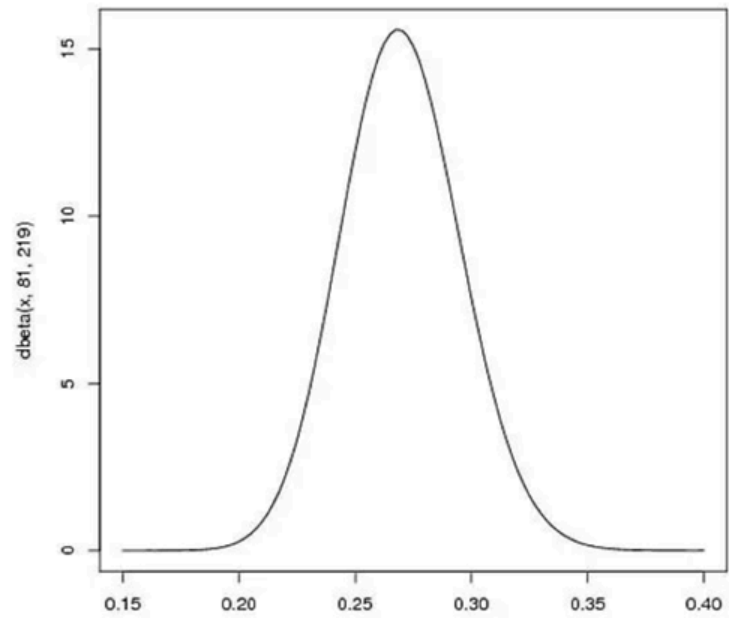
棒球运动的一个指标就是棒球击球率，就是用一个运动员击中的球数除以总的击球数，一般认为0.27是一个平均的击球水平，如果击球率达到0.3就会认为非常优秀了。如果我们要预测一个棒球运动员，他整个赛季的棒球击球率，怎么做呢？你可以直接计算他目前的棒球击球率，用击中数除以击球数。但是，这在赛季开始阶段时是很不合理的。假如这个运动员就打了一次，还中了，那么他的击球率就是100%；如果没中，那么就是0%，甚至打5、6次的时候，也可能运气爆棚全中击球率100%，或者运气很糟击球率0%，所以这样计算出来的击球率是不合理也是不准确的。

为什么呢？

当运动员首次击球没中时，没人认为他整个赛季都会一次不中，所以击球率不可能为0。因为我们有先验期望，根据历史信息，我们知道击球率一般会在0.215到0.36之间。如果一个运动员一开始打了几次没中，那么我们知道他可能最终成绩会比平均稍微差一点，但是一般不可能偏离上述区间，更不可能为0。

如何解决呢？

一个最好的方法来表示这些先验期望（统计中称为先验（prior））就是贝塔分布，表示在运动员打球之前，我们就对他的击球率有了一个大范围预测。假设我们预计运动员整个赛季的击球率平均值大概是0.27左右，范围大概是在0.21到0.35之间。那么用贝塔分布来表示，我们可以取参数 $\alpha=81$ ， $\beta=219$ ，因为 $\alpha/(\alpha+\beta)=0.27$ ，图形分布也主要集中在0.21~0.35之间，非常符合经验值，也就是我们在不知道这个运动员真正击球水平的情况下，我们先给一个平均的击球率的分布。



假设运动员一次击中，那么现在他本赛季的记录是“1次打中；1次打击”。那么我们更新我们的概率分布，让概率曲线做一些移动来反应我们的新信息。

$$\text{Beta}(\alpha_0+\text{hits}, \beta_0+\text{misses})$$

注： α_0 ， β_0 是初始化参数，也就是本例中的81，219。hits表示击中的次数，misses表示未击中的次数。

5. python进行机器学习（一）之数据预处理(3)

推荐排行榜

- 1. 图示详解BERT模型的输入与输出(15)
- 2. Python数据可视化-seaborn(14)
- 3. 推荐算法之Thompson(汤普森)采样(8)
- 4. 残差网络(Residual Network)(5)
- 5. 一文彻底搞懂attention机制(3)

最新评论

1. Re:推荐算法之Thompson(汤普森)采样

写的真好，赞

--安道龙
2. Re:图示详解BERT模型的输入与输出

输入一般是embedding+positional embedding吧。这里多出来的文本向量怎么理解，能举个例子吗

--禾禾斗斗
3. Re:拉格朗日对偶理解

谢谢博主，我是在Boyd的凸优化上也看见了这个先增后减的图，觉得有点问题，搜索到了您的博客。

--土松混合茉莉
4. Re:拉格朗日对偶理解

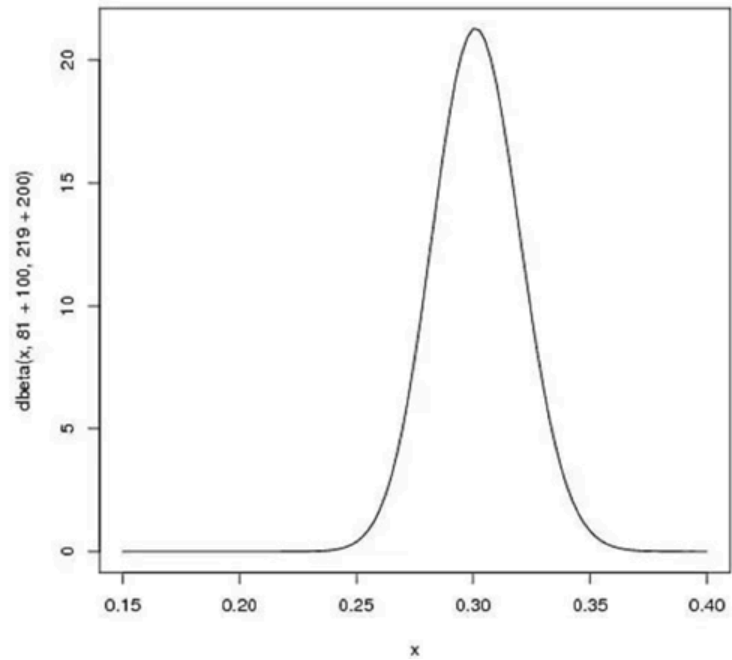
@土松混合茉莉 第三幅图是单独重新举例说明，不管原问题是否凹凸，对偶问题都是凹问题，可以直接转换成凸问题求解，只是为了解释这个问题。...

--光彩照人
5. Re:拉格朗日对偶理解

楼主您好，第三幅图我存在一些疑惑，为何实线是先增后减，从第一幅图中可以看出，lambda较小时，在约束条件成立的区间内，极小值仍是1.54

--土松混合茉莉

击中一次，则新的贝塔分布为Beta(81+1,219)，一次并不能反映太大问题，所以在图形上变化也不大，不画示意图了。然而，随着整个赛季运动员逐渐进行比赛，这个曲线也会逐渐移动以匹配最新的数据。由于我们拥有了更多的数据，因此曲线（击球率范围）会逐渐变窄。假设赛季过半时，运动员一共打了300次，新的贝塔分布是Beta(81+100,219+200)，如下图：



可以看出，曲线更窄而且往右移动了（击球率更高），由此我们对于运动员的击球率有了更好的了解。新的贝塔分布的期望值为0.303，比直接计算100/(100+200)=0.333要低，是比赛季开始时的预计0.27要高，所以贝塔分布能够抛出掉一些偶然因素，比直接计算击球率更能客观反映球员的击球水平。

总结：

这个公式就相当于给运动员的击中次数添加了“初始值”，相当于在赛季开始前，运动员已经有81次击中219次不中的记录。因此，在我们事先不知道概率是什么但又有一些合理的猜测时，贝塔分布能够很好地表示为一个概率的分布。

三、汤普森采样

汤普森采样的背后原理正是上述所讲的Beta分布，你把贝塔分布的 a 参数看成是推荐后用户点击的次数，把分布的 b 参数看成是推荐后用户未点击的次数，则汤普森采样过程如下：

- 1、取出每一个候选对应的参数 a 和 b；
- 2、为每个候选用 a 和 b 作为参数，用贝塔分布产生一个随机数；
- 3、按照随机数排序，输出最大值对应的候选；
- 4、观察用户反馈，如果用户点击则将对应候选的 a 加 1，否则 b 加 1；

注：实际上在推荐系统中，要为每一个用户都保存一套参数，比如候选有 m 个，用户有 n 个，那么就要保存 2 m n个参数。

汤普森采样为什么有效呢？

- 1) 如果一个候选被选中的次数很多，也就是 a+b 很大了，它的分布会很窄，换句话说这个候选的收益已经非常确定了，就是说不管分布中心接近0还是1都几乎比较确定了。用它产生随机数，基本上就在中心位置附近，接近平均收益。
- 2) 如果一个候选不但 a+b 很大，即分布很窄，而且 a/(a+b) 也很大，接近 1，那就确定这是个好的候选项，平均收益很好，每次选择很占优势，就进入利用阶段。反之则有可能平均分布比较接近与 0，几乎再无出头之日。
- 3) 如果一个候选的 a+b 很小，分布很宽，也就是没有被选择太多次，说明这个候选是好是坏还不太确定，那么分布就是跳跃的，这次可能好，下次就可能坏，也就是还有机会存在，没有完全抛弃。那

么用它产生随机数就有可能得到一个较大的随机数，在排序时被优先输出，这就起到了前面说的探索作用。

python代

AI IDE Trae

```
choice = numpy.argmax(pymc.rbeta(1 + self.wins, 1 + self.trials - self.wins))
```

分类: [推荐系统](#)

[好文要顶](#)[关注我](#)[收藏该文](#)[微信分享](#)



[光彩照人](#)
[粉丝 - 140](#) [关注 - 7](#)

[+加关注](#)

80

[升级成为会员](#)

« 上一篇: [推荐算法之E&E](#)
» 下一篇: [推荐系统之构建排行榜](#)

posted @ 2019-07-21 08:56 光彩照人 阅读(31777) 评论(1) 编辑 收藏 举报

[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】还在用 ECharts 开发大屏？试试这款永久免费的开源 BI 工具！
- 【推荐】国内首个AI IDE，深度理解中文开发场景，立即下载体验Trae
- 【推荐】编程新体验，更懂你的AI，立即体验豆包MarsCode编程助手
- 【推荐】抖音旗下AI助手豆包，你的智能百科全书，全免费不限次数
- 【推荐】轻量又高性能的 SSH 工具 IShell：AI 加持，快人一步



编辑推荐：

- Linux系列：如何调试 malloc 的底层源码
- AI与.NET技术实操系列：基于图像分类模型对图像进行分类
- go语言实现终端里的倒计时
- 如何编写易于单元测试的代码
- 10年+ .NET Coder 心语，封装的思维：从隐藏、稳定开始理解其本质意义

阅读排行：

- Apifox不支持离线，Apipost可以！
- 零经验选手，Compose 一天开发一款小游戏！
- 历时 8 年，我冲上开源榜前 8 了！
- Trae 开发工具与使用技巧
- 通过 API 将Deepseek响应流式内容输出到前端