

砸金蛋中奖概率杂谈

作者: [washmore](#)

原文链接: <https://ld246.com/article/1502955341414>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

前言:帕拉丁级

在微信端展开的一系列营销活动项目中,砸金蛋涉及到中奖概率的设置以及实现,不排除后续项目中也涉及到中奖概率,因此将砸金蛋中的抽奖算法分享出来与大家一同讨论和分析如何改进以及衍生其他定化变种!

逻辑:克洛诺斯级

话不多说上干货:

1.首先进行前置操作:筛选掉不可用奖项,出于某些考虑,奖项列表中某些单项并不能被实际领取比如奖太贵重不给中,又比如奖品数量有限已发完

2.进入正题开始罗列实际可用的奖项并计算各个奖项的中奖概率!!!!

- A.前台设置的中奖概率形式为 $1/X$,其中 X 为可变量.因此第一步将各奖项的 X 相乘得到公分母baseRandom,同时将各奖项 $1/X$ 累加得到综合中奖率totalChance;
- B.若综合中奖率totalChance小于等于1,则中奖基数coefficient即为totalChance本身;若totalChance大于1,则中奖基数coefficient为 $1/\text{totalChance}$,所有奖项在计算中奖率的时候都要按coefficient系数行压缩以保证综合中奖率小于等于1
- C.将 $[0,1)$ 左闭右开区间(方便java随机数实现)分为baseRandom份,则每个奖项的实际中奖率为其他项 X 乘积/baseRandom
- D.为方便理解,举个栗子:

现有1234等奖分别中奖概率为 $1/10, 1/5, 1/2, 1/1$,则baseRandom= $10 \times 5 \times 2 \times 1 = 100$,totalChance= $0.1 + 0.2 + 0.5 + 1 = 1.8$,coefficient = $1/1.8$;将 $[0,1)$ 等比划分为 100×1.8 份,其中:

一等奖中将区间为 $[0, 5 \times 2 \times 1 / 180)$

二等奖中将区间为 $[5 \times 2 \times 1 / 180, 5 \times 2 \times 1 / 180 + 10 \times 2 \times 1 / 180)$

三等奖中将区间为 $[5 \times 2 \times 1 / 180 + 10 \times 2 \times 1 / 180, 5 \times 2 \times 1 / 180 + 10 \times 2 \times 1 / 180 + 10 \times 5 \times 1 / 180)$

四等奖中将区间为 $[5 \times 2 \times 1 / 180 + 10 \times 2 \times 1 / 180 + 10 \times 5 \times 1 / 180, 5 \times 2 \times 1 / 180 + 10 \times 2 \times 1 / 180 + 10 \times 5 \times 1 / 180 + 10 \times 5 \times 2 / 180)$

稍微计算后分别为 $[0, 1/18), [1/18, 3/18), [3/18, 8/18), [8/18, 1)$

- E.至此计算用户一次砸蛋行为是否中奖,中几等奖就清晰明了了.使用随机数生成器生成一个 $[0,1)$ 之间double类型数字,位于哪个中将区间则中几等奖

总结:魔像级

在大数据量的支撑下每个用户的中奖概率是公平准确的,并且此种做法可以很好的兼容综合中奖率大于的场景,**不足之处在于每个用户每次单独的砸蛋行为不再变得独立.**各位有什么想说的吗?

后文:恶狼级

这是很久以前(几年前)的一份文档了,最近不知道从哪冒了出来,后来也有考虑(参考)过其他的中奖概率相关问题,有一种流行的做法是把时间同中奖概率绑定起来,随着时间推移,中奖率会快速提高,到达一个峰后,在随时间推移缓慢下降直至降低到时间对中奖率再无影响,或许后续有机会再来设计中奖相关内容时候,可以再想想!

