

【得物技术】如何测试概率性事件 - 二项分布置信区间

 SmartCo...

关注

9 人赞同了该文章

前言

日常开发测试可能会遇到这样一种情况，有一个接口或方法概率触发，那么需要多少次抽样，落在一个什么区间内，才能断定是否按照设定概率进行呢？

本文将以二项分布作为研究手段，分两种情况求解此类问题的置信区间范围，并结合实际案例进行分析。

背景

某一天，测试同学在验证一个接口时遇到了一个问题。

该接口设定为50%概率触发，测试同学写了自动化脚本进行多次调用。

但是问题来了，他并不知道应该调用多少次，然后落在一个什么区间内才算测试通过。

极大的扩大样本容量，然后给一个模糊的范围边界确实能解决这个问题，但是测试同学并不满足于此，他要一个精确的数字！

因此我只能满足他任性的要求，提笔拯救测试同学。

解题思路

在这种情况下，触发 or 不触发，此类概率问题可以看作一个二项分布。我们的目的在于经过一定量



那么就需要考虑这几种极端情况：

1. 概率太小
2. 概率太大
3. 样本数很小
4. 样本数很大

在二项分布中，可以将样本数与事件概率综合起来考虑，合并为以下两种情况

- (a) 概率比较正常或样本数较大 —— $np > 5$ and $n(1-p) > 5$
- (b) 概率比较极端或样本数较小 —— $np \leq 5$ or $n(1-p) \leq 5$

(n：样本总数，p：概率)

二项分布的置信区间估计方法常用的有两种，一是正态分布近似方法，即 Normal Approximation Method；二是精确置信区间法，即 Clopper-Pearson Method。

对于这两种情况可分别应用这两种方法进行求解。

Normal Approximation Method

很容易可以想象出当样本数足够大的情况下，二项分布的曲线分布会愈发趋向于近似正态分布。

因此在情况a中适用于正态分布近似。

置信区间计算公式：

$$p \pm z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n}}$$

因此使用了正态分布近似，因此Z value用的也是正态分布的Z value。

p: 样本中所测得的事件发生概率

a: I类错误率

Z: Z value常量，可查表得知

n: 样本总数

该方法的优点在于简单易理解，但是在极端情况中精度会很差。

Clopper-Pearson Method

在情况b中适用于精确置信区间法

置信区间计算公式（贼复杂，网上大多数贴的都是这个）：



$$\sum_{k=0}^k \binom{n}{k} p_{UB}^k (1 - p_{UB})^{n-k} = \frac{\alpha}{2}$$

$$\sum_{k=x}^n \binom{n}{k} p_{LB}^k (1 - p_{LB})^{n-k} = \frac{\alpha}{2}$$

但是可以发现上述公式是基于Binomial Cumulative Distribution Function，可以通过Beta Distribution来计算，因此经过简化可得如下置信区间计算公式。

置信区间计算公式（经过简化）：

$$p_{ub} = 1 - \text{BetaInv}\left(\frac{\alpha}{2}, n - k, k + 1\right)$$

$$p_{lb} = 1 - \text{BetaInv}\left(1 - \frac{\alpha}{2}, n - k + 1, k\right)$$

n: 样本总数

k: 成功数量

a: I类错误率

BetaInv: 一个算法函数，完全不用理解具体细节，找个别人实现的直接调用即可（包括excel）

这样一来就简单的多了，我甚至可以拿excel解出来。

该方法的优点就在于可以处理极端情况，p是0或1的情况也阔以。

实际案例

案例一（正态分布近似法）

问题：一个50%概率的接口，测试50次，成功28次，判断是否正常。

$$p = 28/50 = 0.56$$

$$np = 0.56 * 50 = 28 > 5$$

$$n(1-p) = 0.44 * 50 = 22 > 5$$

因此使用正态分布近似法。



$$p \pm z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n}}$$

置信度假设为95%，因此查表得知Z值为1.96

$$Z = 1.96$$

$$p = 28/50 = 0.56$$

$$n = 50$$

代入可得，置信区间为[0.56 - 0.14, 0.56 + 0.14]这个范围，因此0.5确实落入这个置信区间，所以可以暂时认为这个我这个接口没问题！

案例二（精确置信区间法）

问题：一个17%概率的接口，测试50次，成功3次，判断是否正常。

$$p = 3 / 50 = 0.06$$

$$np = 0.06 * 50 = 3 < 5$$

适用于精确置信区间法。

$$p_{ub} = 1 - \text{BetaInv}\left(\frac{\alpha}{2}, n - k, k + 1\right)$$

$$p_{lb} = 1 - \text{BetaInv}\left(1 - \frac{\alpha}{2}, n - k + 1, k\right)$$

假设置信度为95%

$$\alpha = 0.05$$

$$n = 50$$

$$k = 3$$

使用Excel或其他方法



O	P	Q
0.16548195	0.01254859	

公式生成器

显示所有函数

BETAINV

Probability = 0.025

0.025

Alpha = 47

47

Beta = 4

4

A = 数值

B = 数值

结果: 0.165481947

完成

BETAINV

计算可得：

Pub = 0.1655

Plb = 0.0125

置信区间为[0.0125, 0.1655]，所以17%概率的接口没有落在置信区间内，可以认为在95%置信度的情况下，该接口出现了问题。

接着有请测试同学发言=。=!

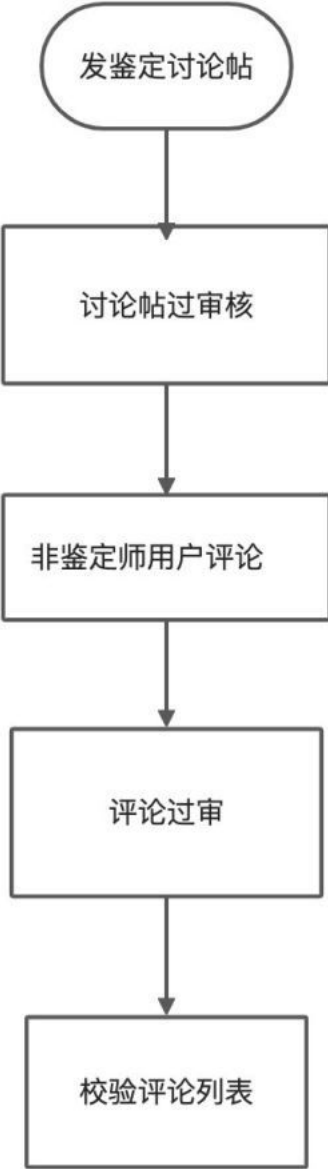
测试的实践

实践的功能与背景

基于鉴定460版ai鉴定师智能回复这个功能。简单来说这个功能就是，你对鉴定贴进行回复时，你的回复内容完全匹配到回复词库时，就会触发ai自动回复。而触发ai自动回复是有一个概率控制的，在测试的时候想不单单校验返回值，也想同时校验这个概率功能的准确性。

测试执行

首先遇到这个功能，我想到的测试步骤就是：



目前就是这个测试步骤，然后通过接口的返回值来判断做判定：

- 1.检查回复中有没有子评论
- 2.检查子评论的回复人是不是ai鉴定师

校验的代码如下

```
if (replyListResponse.getJSONObject("data").getJSONObject("simpleReply").
    getJSONArray("list").getJSONObject(j).getJSONObject("childReply").
    getJSONArray("list")!=null){
    DuAssert.getInstance().assertTrue(replyListResponse.
        getJSONObject("data").getJSONObject("simpleReply").
        getJSONArray("list").getJSONObject(j).getJSONObject("childReply").
        getJSONArray("list").getJSONObject(0).getString("userName").equa
```

有人会问，为什么没有检查回复的信息的正确性呢？首先，测试服的数据不可控因素比较多，比如今天校验了他的回复信息，但是过两天回复信息在验收之后改了，那就还需要对用例进行改动。其次，在正常的测试流程中肯定会覆盖不同的入参信息与不同的返回信息对应情况，只要验证一次这个逻辑，剩下这个数据准确性的验证反而觉得有一些冗余。写针对这么长的测试用例更像是为了保



证整个流程：从发帖到审核，从回复到审核这两个老流程的准确性保证；非鉴定师触发ai鉴别这个基础流程的保证情况。

概率验证

进行到概率验证，做法很简单，比如我们随机50次，然后有子评论，且评论用户为ai鉴别，则n+1。但是这样又会引入一个新问题，这个最后叠加出的n肯定是在一定区间内波动的。比如服务端现在设置为50%回复，那n就是25左右，那这个具体要怎样来设置这个区间，才能较为准确的验证这个概率的正确性，从而在数值没有落在区间内的时候，可以光明正大的给开发提bug，说你的概率算的不准确呢？

从上文中我们得知，需要测试的事件概率为50%，尝试的次数为50次，所以：

$np=50 \times 50\%$;

$n(1-p)=50 \times 50\%$,

均大于5，由此可知概率应符合正态分布曲线，所以用来计算这个置信区间的公式为：

$$p \pm z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n}}$$

代码实现

```
public static double getProbability(int times,double targetP) {  
  
    double z=1.96;  
    return Math.sqrt(targetP*(1-targetP)/times)*z;  
  
}
```

很简单，一行代码就搞定了。有同学可能会问这个1.96是哪来的，实际上这个是查表查的，暂定要求的准确度为95%，于是根据下方的表格查得0.975对应的横纵坐标为0.06和1.9，于是相加得1.96



z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	0.5	0.504	0.508	0.512	0.516	0.52	0.524	0.528	0.532	0.536
0.1	0.54	0.544	0.548	0.552	0.556	0.56	0.564	0.568	0.571	0.575
0.2	0.58	0.583	0.587	0.591	0.595	0.599	0.603	0.606	0.61	0.614
0.3	0.618	0.622	0.626	0.63	0.633	0.637	0.641	0.644	0.648	0.652
0.4	0.655	0.659	0.663	0.666	0.67	0.674	0.677	0.681	0.684	0.688
0.5	0.692	0.695	0.699	0.702	0.705	0.709	0.712	0.716	0.719	0.722
0.6	0.726	0.729	0.732	0.736	0.74	0.742	0.745	0.749	0.752	0.755
0.7	0.758	0.761	0.764	0.767	0.77	0.773	0.776	0.779	0.782	0.785
0.8	0.788	0.791	0.794	0.797	0.8	0.802	0.805	0.808	0.811	0.813
0.9	0.816	0.819	0.821	0.824	0.826	0.829	0.832	0.834	0.837	0.839
1	0.841	0.844	0.846	0.849	0.851	0.853	0.855	0.858	0.85	0.862
1.1	0.864	0.867	0.869	0.871	0.873	0.875	0.877	0.879	0.881	0.883
1.2	0.885	0.887	0.889	0.891	0.893	0.894	0.896	0.898	0.9	0.902
1.3	0.903	0.905	0.907	0.908	0.91	0.912	0.913	0.915	0.916	0.918
1.4	0.919	0.921	0.922	0.924	0.925	0.927	0.928	0.929	0.931	0.932
1.5	0.933	0.935	0.936	0.937	0.938	0.939	0.941	0.942	0.943	0.944
1.6	0.945	0.946	0.947	0.948	0.95	0.951	0.952	0.953	0.954	0.955
1.7	0.955	0.956	0.957	0.958	0.959	0.96	0.961	0.962	0.963	0.963
1.8	0.964	0.965	0.966	0.966	0.967	0.968	0.969	0.969	0.97	0.971
1.9	0.971	0.972	0.973	0.973	0.974	0.974	0.975	0.976	0.976	0.977
2	0.977	0.978	0.978	0.979	0.979	0.98	0.98	0.981	0.981	0.982
2.1	0.982	0.983	0.983	0.983	0.984	0.984	0.985	0.985	0.985	0.986
2.2	0.986	0.986	0.987	0.987	0.988	0.988	0.988	0.988	0.989	0.989
2.3	0.989	0.99	0.99	0.99	0.99	0.991	0.991	0.991	0.991	0.992
2.4	0.992	0.992	0.992	0.993	0.993	0.993	0.993	0.993	0.993	0.994
2.5	0.994	0.994	0.994	0.994	0.995	0.995	0.995	0.995	0.995	0.995
2.6	0.995	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.996
2.7	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997

最后用例代码添加

```
double realP=n/50.00;
double max =Math.ceil(50*realP+50* Probability.getProbability(50,realP);
double min =Math.ceil(50*realP-50* Probability.getProbability(50,realP);
if (25<=max &&25>min){
    DuAssert.getInstance().assertEquals("在区间内， 概率可信","在区间内， 概率
}
else{
    DuAssert.getInstance().assertEquals("不在区间内， 不可信， 提bug! ","在区
}
```

综上所述，我们基于统计的原理校验了这个功能的准确性。

总结

测试概率是测试过程中一个比较模棱两可的事情，如何进行概率事件的测试并有效的发现问题是非常必要的。

因此在上文中主要将此类问题模拟成二项分布进行求解，求得置信区间从而进行较为准确的判断。

但是有一点要注意的是在上文两个案例中都是以95%置信度作为前提，实际上是存在发生I类错误的可能性，所以测出了问题只能说大概率可能出现了问题，而不能立马给一个绝对性的结论，这样是不科学的。

建议是根据统计学经验，先测30+次，能用正态分布近似覆盖就尽量先使用正态分布近似。

发布于 2020-12-25 16:47

测试 概率性事件 二项分布



评论千万条，友善第一条

1 条评论

默认 时间



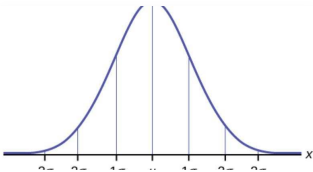
夜雨听风

懂数学 多是一件美事啊👍

2021-12-31

回复 赞

推荐阅读



如何通俗理解置信区间和置信水平

炊烟

数理统计（3）——参数点估计及其优良性

前言数理统计的主要任务是通过样本信息来推断总体的特征，即统计推断工作，总体是由随机变量及其分布函数来刻画的。在实际问题中，可以根据问题本身的背景，确定该随机现象的总体所具有的分...

乔胤博

导致内生性的四个原因

1.模型设定的错误（遗漏变量）举例：经验和教育对工资的影响 解决办法：代理变量；（用IQ来代理智商）工具变量；面板数据。（面板数据是追踪调查的） 2.测量误差只要是非随机的错（系统性...

君小涵

【概率论】 - (1)区间估计

在数据分析过程中，一个完整的环节是从数据中得到洞察，根据洞察得到某种假设，通过实验检验这一假设。实验环节中会涉及到一些概率论知识，比如统计推断中重要的两类问题，区间估计和假设...

茵茵的聪聪 发表于数据分析