

Brunner-Munzel 検定 (exact test)

青木繁伸

2020年3月17日

1 目的

正確な p 値を計算する Brunner-Munzel 検定 (正確確率検定) である。

データ (分割表) によっては計算量が多くなり実用的な時間内で計算が終了できないこともあるので、そのような場合にはモンテカルロ法による近似計算もできる。

周辺和を固定した全ての分割表において Brunner-Munzel 検定の p 統計量と生起確率を求め、実際に観察された分割表の p 統計量より小さいか等しい分割表の生起確率を合計したものを p 値とするものである。

2 使用法

```
import sys
sys.path.append("statlib")
from exact import exact_bm_test
exact_bm_test(x, y=NULL, exact=TRUE, hybrid=FALSE, loop=10000, verbose=True)
```

2.1 引数

| | |
|----------------------|---|
| <code>x</code> | 分割表 (合計を含まない) または、第 1 群のデータベクトル |
| <code>y</code> | <code>x</code> がデータベクトルの場合は、第 2 群のデータベクトル <code>x</code> が分割表の場合には無視される |
| <code>exact</code> | 正確な p 値を求める場合、またはシミュレーションにより近似的な p 値を求めるときには <code>True</code> (デフォルト) <code>False</code> の場合には t 分布による漸近近似のみを行う |
| <code>hybrid</code> | <code>True</code> を指定すれば、シミュレーションにより近似的な p 値を計算する |
| <code>loop</code> | シミュレーションの回数 |
| <code>verbose</code> | 必要最小限のプリント出力をする |

2.2 戻り値の名前

| | |
|-----------------------|----------------------|
| <code>"exactP"</code> | 正確検定による p 値 |
| <code>"simP"</code> | モンテカルロ法による p 値 |
| <code>"t"</code> | 検定統計量 (t 分布にしたがう) |

"df" 自由度
"asymP" 漸近検定による p 値

3 使用例

3.1 分割表を与える場合

```
x = [[5,3,2,1], [2,3,1,2]]  
  
import sys  
sys.path.append("statlib")  
from exact import exact_bm_test  
  
a = exact_bm_test(x)
```

```
Brunner-Munzel test  
t = 0.90861, df = 14.94, p value = 0.37798, estimate = 0.61932  
exact p value = 0.38374  
91 tables are inspected.
```

```
import numpy as np  
np.random.seed(123) # 乱数の種 (通常は不要)  
b = exact_bm_test(x, hybrid=True)
```

```
Brunner-Munzel test  
t = 0.90861, df = 14.94, p value = 0.37798, estimate = 0.61932  
simulated p value = 0.38130  
based on 10000 times simulation.
```

3.2 二群のデータベクトルを与える場合

```
x = [78, 64, 75, 45, 82]  
y = [110, 70, 53, 51]  
c = exact_bm_test(x, y)
```

```
Brunner-Munzel test  
t = 0.20927, df = 6.0242, p value = 0.84113, estimate = 0.45  
exact p value = 0.90476  
126 tables are inspected.
```

```
d = exact_bm_test(x, y, hybrid=True)
```

```
Brunner-Munzel test  
t = 0.20927, df = 6.0242, p value = 0.84113, estimate = 0.45  
simulated p value = 0.90800  
based on 10000 times simulation.
```