DT_proof

Josh Soboil

13/09/2020

An incoherent, Dutch Book series of bets:

```
p1 <- .3
p2 <- .1
p3 <- .4
c(p1, p2, p3)
```

```
## [1] 0.3 0.1 0.4
```

The set of probabilities do not meet the criterion $\mathrm{P}(A\cap, ..., \cap A) = 1$:

```
sum(p1 + p2 + p3)
```

```
## [1] 0.8
```

In matrix form:

```
R <- matrix(c(1 - p1,   - p2,   - p3,
               - p1, 1 - p2,   - p3,
               - p1,   - p2, 1 - p3),
            nrow = 3)
R
```

```
##      [,1] [,2] [,3]
## [1,]  0.7 -0.3 -0.3
## [2,] -0.1  0.9 -0.1
## [3,] -0.4 -0.4  0.6
```

and that we can invert the matrix because it is not singular:

```
matrixcalc::is.singular.matrix(R)
```

```
## [1] FALSE
```

```r
matlib::inv(R)
```

```
##      [,1] [,2] [,3]
## [1,]  2.5  1.5  1.5
## [2,]  0.5  1.5  0.5
## [3,]  2.0  2.0  3.0
```

Notice matrix determinant > 0:

```r
det(R)
```

```
## [1] 0.2
```

Here we give some arbitrary numerical stakes to each event with probability:

```r
s <- c(5, 2, 1)
s
```

```
## [1] 5 2 1
```

Accordingly, if we solve for the system of linear equations, the gains for each bet equal:

```r
G <- solve(a = inv(R), b = s)
G
```

```
## [1]  2.6  1.2 -2.2
```

Notice the sum have positive gains of 2:

```r
round(sum(G))
```

```
## [1] 2
```

Because this series of bets is inconsistent, we can solve for s again:

```r
s_solved <- solve(a = R, b = G)
s_solved
```

```
## [1] 5 2 1
```

and this will work for any arbitrary size of stakes (bets):

```r
s <- c(20291, 213, 13512)
s
```

```
## [1] 20291   213 13512
```

Accordingly, if we solve for the system of linear equations, the gains for each bet equal:

```r
G <- solve(a = inv(R), b = s)
G
```

```
## [1] 10086.2 -3188.6   -94.4
```

Again, notice the sum have positive gains:

```r
round(sum(G))
```

```
## [1] 6803
```

A coherent series of bets:

```r
p1 <- .3
p2 <- .3
p3 <- .4
c(p1, p2, p3)
```

```
## [1] 0.3 0.3 0.4
```

The set of probabilities do meet the criterion $\mathrm{P}(A\cap, ..., \cap A) = 1$:

```r
sum(p1 + p2 + p3)
```

```
## [1] 1
```

In matrix form:

```r
R <- matrix(c(1 - p1,   - p2,   - p3,
              - p1, 1 - p2,   - p3,
              - p1,   - p2, 1 - p3),
            nrow = 3)
R
```

```
##      [,1] [,2] [,3]
## [1,]  0.7 -0.3 -0.3
## [2,] -0.3  0.7 -0.3
## [3,] -0.4 -0.4  0.6
```

and we cannot invert the matrix, because it is a singular matrix with a determinant of zero:

```r
matrixcalc::is.singular.matrix(R)
```

```
## [1] TRUE
```

```
det(R)
```

```
## [1] -6.661338e-17
```

Here we give assign numerical stakes to each event with probability pn:

```
s <- c(5, 2, 1)
s
```

```
## [1] 5 2 1
```

Accordingly, if we solve for the system of linear equations, the gains for each bet equal (we even have to revert to the %*% matrix operator)

```
G <- R%*%s
G
```

```
##      [,1]
## [1,]  2.6
## [2,] -0.4
## [3,] -2.2
```

Notice the net overall gains are 0:

```
round(sum(G))
```

```
## [1] 0
```

... and theefore we cannot solve "for s = R^-1 * g"

This will always result in a net gain of 0 for any arbitrary size of stakes (bets):

```
s <- c(20291, 213, 13512)
s
```

```
## [1] 20291   213 13512
```

Accordingly, if we solve for the system of linear equations, the gains for each bet equal:

```
G <- R%*%s
G
```

```
##         [,1]
## [1,] 10086.2
## [2,] -9991.8
## [3,]   -94.4
```

Again, notice the net overall gains are 0:

```r
round(sum(G))
```

```
## [1] 0
```