
J Book

Jason Corderoy

1 Item Associations

1. d = data. Rows: baskets, columns: products, where 1=product in basket
2. pb = % baskets containing each product
3. ep = Expected % baskets containing each pair of products
4. ap = actual % baskets containing each pair of products
5. Calculate lift: ap/ep

```

nn=:4
]< d=:(,~nn)$ ?2#~ *~nn

|1 1 0 1|
|0 1 1 0|
|0 1 1 1|
|0 1 1 0|

]pb=:(+/% #)"2 d
0.25 1 0.75 0.5
]<ep=:(pb * =/~ i.nn) >. pb *"0 1 pb

|0.25 0.25 0.1875 0.125|
|0.25 1 0.75 0.5|
|0.1875 0.75 0.75 0.375|
|0.125 0.5 0.375 0.5|

]<ap=:>{(+/% #) */"1 y {"1 _1 d}} each { ;~ i.nn

|0.25 0.25 0 0.25|
|0.25 1 0.75 0.5|
|0 0.75 0.75 0.25|
|0.25 0.5 0.25 0.5|

]<lift=:ap%ep

|1 1 0 2|
|1 1 1 1|
|0 1 1 0.666667|
|2 1 0.666667 1|

```

2 Optimisation

1. x_0 = Options
2. x_s = Random selection from options (x_0)
3. x_p = Problem to solve, which is the column sum of x_s
4. Solve it. Solve knowing only x_p and x_0 . Being blind to x_s

```
nn=:4
]<x0=:8* ( ] % +/"1) ( ,~nn) $ ?2#~*~nn
```

2.66667	2.66667	0	2.66667
4	4	0	0
2	2	2	2
0	0	4	4

2	2	2	2
2	2	2	2
2	2	2	2


```
6 6 6 6
```

```
xt=(x0,0) {~ ?20#nn NB. rando solve incl all 0 option
```

```
eval=:3 : '+/ | xp - +/"2 y'
```

```
bs=:3 : '({:xt) ,~ (x0,0){~ ( ] i. <./) {{eval y, } : xt}}"1 x0, 0'
```

```
NB. best solve
```

```
solver=: 3 : 0
```

```
xt=:bs 1
```

```
eval xt
```

```
)
```

```
solver"0 i.25
```

```
128 120 112 104 96 88 80 72 64 56 48 40 32 24 16 13.3333 8 4 4 0 0 0 0
0 0
```


2	2	2	2
0	0	4	4
4	4	0	0