3.3 Illustration of the productConfig package

Diego Aviles

2016-03-28

First, let us look at the data:

```
tail.matrix(camera_data)
```

```
##       cid usid round atid   selected selectable
## 1823 1835   62     1    4 0.01944444          1
## 1824 1836   62     2    4 0.16805556          1
## 1825 1837   63     0    1 1.00000000          2
## 1826 1838   63     0    2 1.00000000          1
## 1827 1839   63     0    3 1.00000000          1
## 1828 1840   63     0    4 0.16805556          1
```

As you can see our data displays 1828 rows with around 63 different users in a rather complex format which makes it practically difficult to work with. This is the reason we need the basic function cluster `GetFunctions`. For example, it is quite necessary to know how many attributes there are in out data:

```
get_attrs_ID(dataset=camera_data)
```

```
## [1] 1 2 3 4
```

Given that our functions are mostly vectorized and assuming all users have the same attribtues, we can ask for the unique values of each `attr`.

```
getAttrValues(dataset=camera_data, attr = c(1,2,3,4))
```

```
lapply(temp, unique)
```

```
## $`1`
## [1] 3 0 2 1
##
## $`2`
## [1] 0 3 2 1
##
## $`3`
## [1] 0 3 2 1
##
## $`4`
## [1]  0.16805556 -0.27777778 -0.12916667  0.01944444  0.46527778  0.31666667
## [7]  0.61388889
```

Now that we know how many attributes there are, we also know how many columns the decision matrices have. The number of rows depends on how much each user interacted with the product configurator and again, since we can calculate the number of rows for all users using `getRoundsById`.

```
all.rounds <- getRoundsById(camera_data, userid = getAllUserIds(camera_data))
head(all.rounds,3)
```

```
## $`6`
##   [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
##  [24] 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
##  [47] 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
##  [70] 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91
##  [93] 92 93 94 95 96 97 98 99
##
## $`9`
## [1] 0 1 2 3
##
## $`10`
## [1] 0 1 2 3 4
```