Simultaneous Adversarial Knapsack

Ben Ewing

November 25, 2018

Setup

Libraries.

```
# For data manipulation
library(dplyr)
library(purrr)
library(tidyr)
# For kurtosis and skew functions
library(e1071)
# RcppAlgos::comboGeneral is ~80x faster than utils::combn in my quick test, it also seems to beat
# data.table::CJ. It can also run in parallel, but this doesn't seem to help on Windows. Might be
# worth trying on the Econ linux machines?
library(RcppAlgos)
# For compiling functions to bytecode, this usually nets a tiny speedup
library(compiler)
# For linear programming
library(ROI)
library(ROI.plugin.glpk)
library(ompr)
library(ompr.roi)
```

Load knapsack functions.

```
source("../r/knapsack_functions_mat.R")
```

Simulation settings.

```
# number of draws from each items distribution, used to estimated utility of the item
n_draws <- 100000
```

Premise

There are n players (indexed by i), each with budget b_i. There are m items (indexed by j) with price p_j and value v_j, which is a random variable with a known distribution. Unless otherwise stated, we assume that all v_j are uncorrelated. Any additional beliefs players hold about an item's value are captured in an error term $\epsilon_{i,j}$. Unless otherwise stated, we assume the error for all players and items has mean 0.

The player who forms the knapsack with the highest value wins. Players only care about winning: utility of 1 for winning, 0 otherwise. Players form knapsacks simultaneously, and values are only revealed after formation.

Case 1

- $n = 2$ (two players)
    - $b_i = 1$ for all i (all have a budget of one)

- $m > 1$ (more than one item)

    - all v_j are equal in distribution (all items are the same)
    - $p_j = 1$ for all j (all items have a price of one)

```r
n_players <- 2
knap_size <- 1

tmp <- matrix(c(
  rnorm(n_draws, 0, 1),
  rnorm(n_draws, 0, 1)
), ncol = 2) %>%
  form_knapsacks(knap_size) %>%
  estimate_knapsack_utility(n_players)

tmp %>%
  group_by(knap_id) %>%
  summarise_if(is.numeric, mean) %>%
  arrange(utility)
```

```
## # A tibble: 2 x 7
##   knap_id win_prop  utility      mean    sd  skewness kurtosis
##     <int>    <dbl>    <dbl>     <dbl> <dbl>     <dbl>    <dbl>
## 1       2    0.249 -0.00117 -0.000990 1.000  -0.00915   0.0279
## 2       1    0.251  0.00117  0.00163  1.00    0.000817  0.00217
```

Because all items are equally likely to take on the same value, players will just try not to choose the same item. In other words, players will play a mixed strategy where each item has an equal chance of being selected: $\frac{1}{m}$.

Case 2

- $n = 2$ (two players)

    - $b_i = 1$ for all i (all have a budget of one)

- $m > 1$ (more than one item)

    - All v_j have the same expectation, different variance, and are symmetric.
    - $p_j = 1$ for all j (all items have a price of one)

```r
n_players <- 2
knap_size <- 1

tmp <- matrix(c(
  rnorm(n_draws, 0, 1),
  rnorm(n_draws, 0, 3)
), ncol = 2) %>%
  form_knapsacks(knap_size) %>%
  estimate_knapsack_utility(n_players)

tmp %>%
  group_by(knap_id) %>%
  summarise_if(is.numeric, mean) %>%
  arrange(utility)
```

```
## # A tibble: 2 x 7
##   knap_id win_prop   utility      mean    sd skewness kurtosis
##     <int>    <dbl>     <dbl>     <dbl> <dbl>    <dbl>    <dbl>
## 1       1    0.250 -0.000930 -0.000343  1.00 -0.00589  0.00240
## 2       2    0.250  0.000930 -0.00267   2.99 -0.00573  0.0195
```

With two players, variance does not matter so long as expectation is approximately the same. Players will again play a mixed strategy in an attempt to avoid choosing the same items.

Case 3

- $n = 2$ (two players)
    - $b_i = 1$ for all i (all have a budget of one)
- $m > 1$ (more than one item)
    - All v_j have different expectation, different variance, and are symmetric.
    - $p_j = 1$ for all j (all items have a price of one)

```r
n_players <- 2
knap_size <- 1

tmp <- matrix(c(
  rnorm(n_draws, 0, 1),
  rnorm(n_draws, 5, 3),
  rnorm(n_draws, 10, 5)
), ncol = 2) %>%
  form_knapsacks(knap_size) %>%
  estimate_knapsack_utility(n_players)

tmp %>%
  group_by(knap_id) %>%
  summarise_if(is.numeric, mean) %>%
  arrange(utility)
```

```
## # A tibble: 2 x 7
##   knap_id win_prop utility  mean    sd skewness kurtosis
##     <int>    <dbl>   <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1       1   0.0465  -0.407  1.67  3.04     1.28    0.987
## 2       2   0.454    0.407  8.33  5.01     0.337   -0.106
```

```r
strat_minimax_2(tmp)$solution
```

```
##   row_pr[1]   row_pr[2] row_utility
##           0           1           0
```

Players will choose the item with the highest expectation, variance does not matter.

Case 4

- $n = 3$ (three players)
    - $b_i = 1$ for all i (all have a budget of one)
```

- $m > n$ (more than one item)

  - All $v_j$ have the same expectation, different variance, and are symmetric.
  - $p_j = 1$ for all $j$ (all items have a price of one)

```
n_players <- 3
knap_size <- 1

tmp <- matrix(c(
  rnorm(n_draws, 5, 1),
  rnorm(n_draws, 5, 3),
  rnorm(n_draws, 5, 5),
  rnorm(n_draws, 5, 7),
  rnorm(n_draws, 5, 9)
), ncol = 5) %>%
  form_knapsacks(knap_size) %>%
  estimate_knapsack_utility(n_players)

tmp %>%
  group_by(knap_id) %>%
  summarise_if(is.numeric, mean) %>%
  arrange(utility)
```

```
## # A tibble: 5 x 7
##    knap_id win_prop utility  mean    sd skewness  kurtosis
##      <int>    <dbl>   <dbl> <dbl> <dbl>    <dbl>     <dbl>
## 1        3    0.368   0.156  5.01  4.99 0.00565    0.0170
## 2        1    0.368   0.156  5.00  1.00 0.000767  -0.0127
## 3        4    0.369   0.158  5.04  6.99 0.0184    -0.000515
## 4        2    0.369   0.159  5.01  3.00 0.00335   -0.0358
## 5        5    0.369   0.160  5.04  8.98 0.00183   -0.00819
```

```
strat_minimax_3(tmp)$solution
```

```
##     row_pr[1]     row_pr[2]     row_pr[3]     row_pr[4]     row_pr[5]
##   0.3342091566  0.0000000000  0.3338238647  0.0000000000  0.3319669787
##    row_utility
## -0.0002487839
```

No idea. It looks like players randomize over every item, except that with the highest expectation?

Case 5

- $n = 3$ (three players)

  - $b_i = 1$ for all i (all have a budget of one)

- $m > n$ (more than one item)

  - All v_j have different expectation, different variance, and are symmetric.
  - $p_j = 1$ for all j (all items have a price of one)

4

```
n_players <- 3
knap_size <- 1

tmp <- matrix(c(
  rnorm(n_draws, 5, 1),
  rnorm(n_draws, 6, 3),
  rnorm(n_draws, 7, 5),
  rnorm(n_draws, 8, 7),
  rnorm(n_draws, 9, 3)
), ncol = 5) %>%
  form_knapsacks(knap_size) %>%
  estimate_knapsack_utility(n_players)

tmp %>%
  group_by(knap_id) %>%
  summarise_if(is.numeric, mean) %>%
  arrange(utility)
```

```
## # A tibble: 5 x 7
##   knap_id win_prop utility  mean    sd  skewness kurtosis
##     <int>    <dbl>   <dbl> <dbl> <dbl>     <dbl>    <dbl>
## 1       1    0.214  -0.304  5.00 0.997 -0.000627  0.0294
## 2       2    0.312 -0.0128  6.00 3.01   0.00485   0.0198
## 3       3    0.375   0.177  6.99 4.98   0.0114    0.00415
## 4       4    0.418   0.306  8.00 6.99  -0.00195  -0.00537
## 5       5    0.524   0.624  9.00 3.00   0.000863 -0.0105
```

```
strat_minimax_3(tmp)$solution
```

```
##   row_pr[1]   row_pr[2]   row_pr[3]   row_pr[4]   row_pr[5] row_utility
## 0.31261218 0.00000000 0.18043201 0.35347134 0.15348447 -0.07620366
```