Matching students to advisers 2019-09-02 data

This is an R Markdown Notebook.

Algorithm on the "real" data.

Needed software

```
source("hungar.R")
source("do_match.R")
```

Read in data

Use a separate data conversion

```
source("cvt_2019_09_02.R")
students<- student_preferences[,1:2]
```

Post processing until we have a useful weights table

Added "tracks" and some refinements. Allocated 11 slots per adviser. Ran many simulations but it was clear that the workgroup languages should be EN, NL, NL, NL, EN.

Writing as a function

We want workgroups with one language, so later on we have to run the matching multiple times with different choices for the workgroups. So lets write a function that presents those results.

```
LANGUAGE_MISMATCH = -10
LANGUAGE_BONUS = 2
LANGUAGE_BONUS2 = 5
TOO_MANY_STUDENTS = -100 ## overflow per advisor
FORBIDDEN_COMBINATION = -1000
TRACK_MISMATCH = -10
SQUEEZE_POWER = 2 ## see below for more explanations

fixed_lang <- function(d2,d3,d4, topics){
  topics[3:8, 4] <- c(d2,d2,d3,d3,d4,d4)
  student_topic<- student_preferences[, 3:11]
    # tracks
    a<- topics$track == "D"
    b<- student_preferences$track == "F"
    student_topic[b,a] <- student_topic[b,a] + TRACK_MISMATCH

    a<- topics$track == "F"
    b<- student_preferences$track == "D"
    student_topic[b,a] <- student_topic[b,a] + TRACK_MISMATCH

    a<- topics$scr_lang == "EN"
  b<- student_preferences$scr_lang == "NL"
  student_topic[b,a] <- student_topic[b,a] + LANGUAGE_MISMATCH

  a<- topics$scr_lang == "NL"
  b<- student_preferences$scr_lang == "EN"
  student_topic[b,a] <- student_topic[b,a] + LANGUAGE_MISMATCH

  a<- topics$wg_lang == "EN"
  b<- student_preferences$wg_lang == "NL"
  student_topic[b,a] <- student_topic[b,a] + LANGUAGE_MISMATCH

  a<- topics$wg_lang == "NL"
  b<- student_preferences$wg_lang == "EN"
  student_topic[b,a] <- student_topic[b,a] + LANGUAGE_MISMATCH

  # bonus point for language preferences

  a<- topics$scr_lang == "EN"
  b<- student_preferences$scr_lang %in%  c("EN", "VE")
  student_topic[b,a] <- student_topic[b,a] + LANGUAGE_BONUS
  a<- topics$scr_lang == "NL"
  b<- student_preferences$scr_lang %in%  c("NL", "VN")
  student_topic[b,a] <- student_topic[b,a] + LANGUAGE_BONUS
  a<- topics$wg_lang == "EN"
  b<- student_preferences$wg_lang %in%  c("EN", "VE")
  student_topic[b,a] <- student_topic[b,a] - LANGUAGE_BONUS
  a<- topics$wg_lang == "NL"
  b<- student_preferences$wg_lang %in%  c("NL", "VN")
  student_topic[b,a] <- student_topic[b,a] - LANGUAGE_BONUS

 # negative bonus point for language preferences

  a<- topics$scr_lang == "NL"
  b<- student_preferences$scr_lang %in%  c("EN", "VE")
  student_topic[b,a] <- student_topic[b,a] - LANGUAGE_BONUS2
  a<- topics$scr_lang == "EN"
  b<- student_preferences$scr_lang %in%  c("NL", "VN")
  student_topic[b,a] <- student_topic[b,a] - LANGUAGE_BONUS2
  a<- topics$wg_lang == "NL"
  b<- student_preferences$wg_lang %in%  c("EN", "VE")
  student_topic[b,a] <- student_topic[b,a] - LANGUAGE_BONUS2
  a<- topics$wg_lang == "EN"
  b<- student_preferences$wg_lang %in%  c("NL", "VN")
  student_topic[b,a] <- student_topic[b,a] - LANGUAGE_BONUS2

  # collapse per advisor; not generic code yet

  cost<- student_topic[, 1:5] # just for the size
  names(cost)<- advisers$ID
  mem_s_t <- cost #idem

  for (j in 1:4){
    range2 <- 2*j
```

```
      range1 <- range2-1
      up <- pmax(student_topic[, range1],
                         student_topic[, range2])
      cost[, j ] <- up
      mem_s_t[, j] <- ifelse(
        up== student_topic[, range1], range1, range2)

    }
    cost[, 5] <- student_topic[,9]
    mem_s_t[, 5] <- 9

    # force unpopular docet1 to get 8 students
    # cost<- cost-1
    # cost[,1]<- cost[,1] +5
    #
    match_table<- do_match(cost, advisers$max,
                  penalty = TOO_MANY_STUDENTS,
                  squeeze_power = SQUEEZE_POWER)
    match_table$topic = sapply(match_table$nr, function(i)
       mem_s_t[i, match_table$adviser[i]])
    match_table$input= student_topic

    return(match_table)
}
```

Now, later we will uncover the best language choice, but here just show the best result.

There is one more parameter at play here: SQUEEZE_POWER. Explanation:

Giving almost all students their prefered advisor but one student his/her least prefered advisor night raise some eyebrows, even though it maximizes the sum of values. If that effect is too gross, the high values can be "squeezed down" a little bit so that for instance 8 eights will weight more than 7 nines and 1 one. SQUEEZE_POWER of 1 is normal behaviour, use 2-10 for strong(er) equlizing effects.

```
match_table<- fixed_lang("NL", "NL","NL", topics)

final_result<- data.frame(
  nr= students$nr[match_table$nr],
  ID= students$ID[match_table$nr],
  adviser= advisers$ID[match_table$adviser],
  topic = match_table$topic,
  weight = match_table$value
)
```

You might want to print or save the result

```
# print or save final result
# write.csv(final_result, "fr.csv", row.names= FALSE, quote= FALSE)

print(sum(final_result$weight))
```

```
## [1] 371
```

```
f_result <- final_result[order(final_result$weight),]
```

```
knitr::kable(f_result, col.names= names(f_result), row.names=FALSE, caption="Best Matching, ordered by
```

Table 1: Best Matching, ordered by weight

nr	ID	adviser	topic	weight
11	2001105	104	7	5
2	1255574	102	3	6
10	2000815	105	9	6
26	2004509	104	7	6
29	2009603	101	2	6
30	2009735	104	7	6
34	2012967	104	7	6
3	1270230	103	6	7
6	1279954	103	5	7
8	2000435	102	3	7
15	2002339	103	6	7
24	2003951	105	9	7
25	2004318	105	9	7
28	2008878	104	7	7
31	2011116	103	6	7
32	2011891	104	7	7
35	2013063	103	6	7
36	2013126	105	9	7
44	2027800	103	5	7
1	154379	101	2	8
4	1278484	104	7	8
5	1279806	105	9	8
13	2002094	102	4	8
16	2002559	102	3	8
27	2004929	101	1	8
37	2013587	104	7	8
7	1279982	105	9	9
9	2000527	101	1	9
12	2001204	101	1	9
14	2002268	101	2	9
17	2002568	103	6	9
19	2002950	102	3	9
20	2003132	103	5	9
21	2003419	102	3	9
22	2003715	103	6	9
23	2003734	105	9	9
33	2012759	103	6	9
41	2015272	105	9	9
42	2015457	101	1	9
45	2031867	103	6	9
18	2002685	105	9	10
40	2013982	104	8	10
38	2013651	104	8	11
39	2013829	105	9	11
43	2015984	104	8	11
46	12000167	102	3	11

Diagnostics

Next look at some other details of the matching.

```
diag1 <- aggregate(final_result[,3],
                   by=list(final_result$adviser),
                   FUN=length)
names(diag1)<- c("ID", "nr_students")
diag1<- merge(advisers, diag1, by="ID")

knitr::kable(diag1, caption="Adviser load")
```

Table 2: Adviser load

ID	max	nr_students
101	11	7
102	11	7
103	11	11
104	11	11
105	11	10

```
##### check languages

diag2<- final_result[order(final_result$adviser,
                           final_result$topic,
                           final_result$weight),
                     c(3:4, 1:2,5)]
diag2$scr_lang <- student_preferences$scr_lang[diag2$nr]
diag2$wg_lang <- student_preferences$wg_lang[diag2$nr]

knitr::kable(diag2, row.names=FALSE,
             caption= "Best matching, by adviser")
```

Table 3: Best matching, by adviser

adviser	topic	nr	ID	weight	scr_lang	wg_lang
101	1	27	2004929	8	EN	EN
101	1	9	2000527	9	EN	
101	1	12	2001204	9	EN	EN
101	1	42	2015457	9	VE	
101	2	29	2009603	6	EN	EN
101	2	1	154379	8	EN	
101	2	14	2002268	9	EN	EN
102	3	2	1255574	6	VN	
102	3	8	2000435	7	NL	NL
102	3	16	2002559	8	VN	
102	3	19	2002950	9	NL	
102	3	21	2003419	9	NL	NL
102	3	46	12000167	11	NL	
102	4	13	2002094	8	NL	NL
103	5	6	1279954	7	EN	NL
103	5	44	2027800	7	NL	NL
103	5	20	2003132	9	NL	

adviser	topic	nr	ID	weight	scr_lang	wg_lang
103	6	3	1270230	7	NL	NL
103	6	15	2002339	7	NL	NL
103	6	31	2011116	7	NL	NL
103	6	35	2013063	7	NL	NL
103	6	17	2002568	9	NL	
103	6	22	2003715	9	VN	
103	6	33	2012759	9	VN	
103	6	45	2031867	9	NL	
104	7	11	2001105	5	NL	NL
104	7	26	2004509	6	NL	NL
104	7	30	2009735	6	EN	NL
104	7	34	2012967	6	VN	NL
104	7	28	2008878	7	VN	
104	7	32	2011891	7	VN	
104	7	4	1278484	8	NL	
104	7	37	2013587	8	VN	
104	8	40	2013982	10	VN	
104	8	38	2013651	11	VN	
104	8	43	2015984	11	VN	
105	9	10	2000815	6	EN	EN
105	9	24	2003951	7	EN	EN
105	9	25	2004318	7	EN	EN
105	9	36	2013126	7	EN	EN
105	9	5	1279806	8	EN	EN
105	9	7	1279982	9	EN	EN
105	9	23	2003734	9	EN	
105	9	41	2015272	9	VE	
105	9	18	2002685	10	EN	
105	9	39	2013829	11	EN	

Running the above without specifying the workgroup languages led to chaos. That's why we have to fix them, and the code below shows that in this case "EN", "NL", "NL" is the choice with the hoghest score, although some others come close.

```
for (d2 in c("EN", "NL"))
for (d3 in c("EN", "NL"))
for (d4 in c("EN", "NL")){
    matched <- fixed_lang(d2, d3, d4, topics)
    print(c(d2,d3,d4, sum(matched$value)))
  }
```

```
## [1] "EN"  "EN"  "EN"  "204"
## [1] "EN"  "EN"  "NL"  "322"
## [1] "EN"  "NL"  "EN"  "316"
## [1] "EN"  "NL"  "NL"  "355"
## [1] "NL"  "EN"  "EN"  "266"
## [1] "NL"  "EN"  "NL"  "349"
## [1] "NL"  "NL"  "EN"  "357"
## [1] "NL"  "NL"  "NL"  "371"
```

So that settles the distribution of languages