Matching students to advisers 2019-09-06 data

Processed 2019-09-09

This is an R Markdown Notebook.

Algorithm on the “real” data 0f 2019-09-06.

Load software and set parameters

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

TRACK_MISMATCH = -25        ## when the topic is not fit
                            ## for the student track
LANGUAGE_MISMATCH = -10     ## when the student is NL or EN only
                            ## but not supported by topic
LANGUAGE_BONUS = 0          ## topic in studentss pref lang
LANGUAGE_MALUS = -7         ## topic NOT in studentss pref lang
TOO_MANY_STUDENTS = -100    ## overflow penalty per adviser
FORBIDDEN_COMBINATION = -1000 ## what it says

## standard we just optimize sum of weights, but for advanced
## purposes we can use other functions for which
## this is a parameter

SQUEEZE_POWER = 1 ## default - does nothing
```

Read in data

The spreadsheet out of the students administartion system was not entirely correct, so had to be corrected manually. We read the csv files that already contain the corrections.

```r
# edited advisers manually
advisers <- read.csv("./data_2019_09_06/advisers.csv", sep=";",
                     stringsAsFactors = FALSE)

# copied teachers tab in xls to csv
# the name is a bit misleading because it is per topic
df<- read.csv("./data_2019_09_06/teachers.csv", sep=",",
              stringsAsFactors = FALSE)
# transfer to useful format
topics <- data.frame(nr=1:9,
       ID= substr(df$Topic, 1, 21),
       scr_lang=  c("EN", "NL", "")[
         match(df$Language, c("English", "Dutch", "Both"))],
       wg_lang=c("EN", "EN", rep("",6), "EN"),
       docent= df$Teacher_ID-100,
       full_text= df$Topic,
       track= c("D", "F", "")[
         match(df$Program, c("Developmental", "Forensic", "Both"))],
       stringsAsFactors = FALSE
)

## --------- student preferences
df<- read.csv("./data_2019_09_06/prefs.csv", sep=",",
              stringsAsFactors = FALSE)
## that table is not yet in a usable format
topics2 <- as.data.frame(matrix(0, nrow=nrow(df), ncol=9))
names(topics2) <- paste0("topic_", 1:9)
d1<- unique(df$Thesis_write_language)
d1<- d1[order(d1)]; d1T <- c("NL", "EN", "VN", "VE")
d2<- unique(paste0(df$Dutch_wg, df$Englis_wg))
d2T<- c("", "NL", "EN")
d3<- unique(df$Track); d3T <- c("FE", "DN", "FN", "DE")
student_preferences <- data.frame(
  nr= 1:nrow(df),
  ID= df$Studentnr,
  topics2,
  scr_lang=  d1T[match(df$Thesis_write_language, d1)],
  wg_lang = d2T[match(paste0(df$Dutch_wg, df$Englis_wg), d2)],
  track= d3T[match(df$Track, d3)],
  stringsAsFactors = FALSE
)

for ( i in 1:nrow(df)) {
  indx= as.integer(substring(df[i, 2:10],1,1))
  student_preferences[i, 2+indx] <- (9:1)
}
```

Check data rules.

There are some inconsistencies, f.i. students in an English language track indicating preferences for a thesis in Dutch. The following code applies all known rules.

```
# ------ "english tracks are only allowed to write in english analysis"
change1 <- student_preferences$nr[student_preferences$wg_lang == "NL" &
                         student_preferences$track %in% c("FE", "DE")]

corset1<- student_preferences[change1,c(1,2,12,13,14)]


student_preferences$wg_lang[change1] <- ""

# # ------- "only-lamguage for wg and scr cannot conflict"
change2 <- student_preferences$nr[student_preferences$scr_lang == "EN" &
                                    student_preferences$wg_lang == "NL"]
change3 <- student_preferences$nr[student_preferences$scr_lang == "NL" &
                                    student_preferences$wg_lang == "EN"]
change4 <- student_preferences$nr[student_preferences$scr_lang == "VN" &
                                    student_preferences$wg_lang == "EN"]
student_preferences$wg_lang[change4] <- ""
change5 <- student_preferences$nr[student_preferences$scr_lang == "VE" &
```

```
                                    student_preferences$wg_lang == "NL"]

# student 2009735 cant do number 9
changes6 <- student_preferences$nr[student_preferences$ID == 2009735]
student_preferences$topic_9[changes6] <-FORBIDDEN_COMBINATION


# ------ Finally remove language from track code
student_preferences$track<- substring(student_preferences$track, 1, 1)
```

Some attrubutes were changed.

```
knitr::kable(corset1, row.names = F, caption="Corrected entries")
```

Table 1: Corrected entries

nr	ID	scr_lang	wg_lang	track
6	1279954	EN	NL	FE
29	2009735	EN	NL	DE

The matching function

We want workgroups with one language, so we have to run the matching multiple times with different choices for the workgroups.

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

 # 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)
}
```

Eight language options for all workgroups

```
matched <- list()
i<- 0
for (d2 in c("EN", "NL"))
for (d3 in c("EN", "NL"))
for (d4 in c("EN", "NL")){
    i<- i+1
    m <- fixed_lang(d2, d3, d4, topics)
    cat("\n", i, "-", d2,d3,d4, "-", sum(m$value))
    matched[[i]]<- list(index=paste0(d2,d3,d4), matched=m)
  }
```

```
##
##  1 - EN EN EN - 179
##  2 - EN EN NL - 291
##  3 - EN NL EN - 279
##  4 - EN NL NL - 293
##  5 - NL EN EN - 230
##  6 - NL EN NL - 330
##  7 - NL NL EN - 326
##  8 - NL NL NL - 332
```

Discussion and overview of result

We see that the last result is the best. For the three middle workgroups Dutch is the best workgroup language. There are various ways to check the robustness of the result, f.i. by reordering the student preferences and to see if somebody else will get assigned to a low weight topic.

A second way is to alter the measured objective. That is now the sum of weights, but it could be any additive fucntion. F.i. it could be the sum of sqr roots of weights (with negative weights unchanged), etc. The function do_match can be given alternative functions. We have tried a couple, but in this case there is not much of a difference.

So let's look at the reults.

```
match_table<- matched[[8]]$matched

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

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

knitr::kable(f_result, col.names= names(f_result), row.names=FALSE, caption="Individul weights for best
```

Table 2: Individul weights for best matching

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

nr	ID	adviser	topic	weight
44	12000167	102	3	9

Per adviser

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", align="l")
```

Table 3: Adviser load

ID	max	nr_students
101	9	9
102	9	9
103	8	8
104	9	9
105	9	9

```
##### 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= "Students by adviser", align="l")
```

Table 4: Students by adviser

adviser	topic	nr	ID	weight	scr_lang	wg_lang
101	1	9	2000527	7	EN	
101	1	40	2015457	7	VE	
101	1	26	2004929	8	EN	EN
101	1	12	2001204	9	EN	EN
101	2	1	154379	6	EN	
101	2	22	2003734	6	EN	
101	2	28	2009603	6	EN	EN
101	2	29	2009735	6	EN	
101	2	14	2002268	9	EN	EN
102	3	2	1255574	4	VN	
102	3	15	2002559	6	VN	
102	3	8	2000435	7	NL	NL

adviser	topic	nr	ID	weight	scr_lang	wg_lang
102	3	18	2002950	7	NL	
102	3	20	2003419	9	NL	NL
102	3	44	12000167	9	NL	
102	4	38	2013982	2	VN	
102	4	41	2015984	5	VN	
102	4	13	2002094	8	NL	NL
103	5	19	2003132	9	NL	
103	6	3	1270230	9	NL	NL
103	6	16	2002568	9	NL	
103	6	21	2003715	9	VN	
103	6	30	2011891	9	VN	
103	6	31	2012759	9	VN	
103	6	33	2013063	9	NL	NL
103	6	43	2031867	9	NL	
104	7	11	2001105	7	NL	NL
104	7	27	2008878	7	VN	
104	7	42	2027800	7	NL	NL
104	7	4	1278484	8	NL	
104	7	25	2004509	8	NL	NL
104	7	32	2012967	8	VN	NL
104	7	35	2013587	8	VN	
104	7	39	2015272	9	VE	
104	8	36	2013651	9	VN	
105	9	10	2000815	6	EN	EN
105	9	6	1279954	7	EN	
105	9	23	2003951	7	EN	EN
105	9	24	2004318	7	EN	EN
105	9	34	2013126	7	EN	EN
105	9	5	1279806	8	EN	EN
105	9	17	2002685	8	EN	
105	9	7	1279982	9	EN	EN
105	9	37	2013829	9	EN	

Lets repeat this final overview per workgroup.

```r
wg_list <- list(1:2, 3:4, 5:6, 7:8, 9)
wg_lang <- c("EN", rep("NL", 3), "EN")

for ( jj in 1:5){
  tabl <- final_result[final_result$topic %in% wg_list[[jj]], c("nr", "ID")]
  names(tabl)[2] <- "Stud_ID"
  tabl$topic <- topics$ID[final_result$topic[tabl$nr]]
  tabl$pref_scr_lang <- student_preferences$scr_lang[tabl$nr]

  tabl<- tabl[order(tabl$topic, tabl$nr), ]
  capt=sprintf("Workgroup %d, lang= %s", jj, wg_lang[jj])
  print(knitr::kable(tabl, row.names=FALSE, caption = capt, align="l"))

}
```

Table 5: Workgroup 1, lang= EN

nr	Stud_ID	topic	pref_scr_lang
9	2000527	1. Do Irrational Deci	EN
12	2001204	1. Do Irrational Deci	EN
26	2004929	1. Do Irrational Deci	EN
40	2015457	1. Do Irrational Deci	VE
1	154379	2. Personality Traits	EN
14	2002268	2. Personality Traits	EN
22	2003734	2. Personality Traits	EN
28	2009603	2. Personality Traits	EN
29	2009735	2. Personality Traits	EN

Table 6: Workgroup 2, lang= NL

nr	Stud_ID	topic	pref_scr_lang
2	1255574	3. Reken maar! Wat te	VN
8	2000435	3. Reken maar! Wat te	NL
15	2002559	3. Reken maar! Wat te	VN
18	2002950	3. Reken maar! Wat te	NL
20	2003419	3. Reken maar! Wat te	NL
44	12000167	3. Reken maar! Wat te	NL
13	2002094	4. Rekenangst, alleen	NL
38	2013982	4. Rekenangst, alleen	VN
41	2015984	4. Rekenangst, alleen	VN

Table 7: Workgroup 3, lang= NL

nr	Stud_ID	topic	pref_scr_lang
19	2003132	5. Forensic psychopat	NL
3	1270230	6. The Dark Side of T	NL
16	2002568	6. The Dark Side of T	NL
21	2003715	6. The Dark Side of T	VN
30	2011891	6. The Dark Side of T	VN
31	2012759	6. The Dark Side of T	VN
33	2013063	6. The Dark Side of T	NL
43	2031867	6. The Dark Side of T	NL

Table 8: Workgroup 4, lang= NL

nr	Stud_ID	topic	pref_scr_lang
4	1278484	7. Sociale positie, s	NL
11	2001105	7. Sociale positie, s	NL
25	2004509	7. Sociale positie, s	NL
27	2008878	7. Sociale positie, s	VN
32	2012967	7. Sociale positie, s	VN
35	2013587	7. Sociale positie, s	VN
39	2015272	7. Sociale positie, s	VE
42	2027800	7. Sociale positie, s	NL

nr	Stud_ID	topic	pref_scr_lang
36	2013651	8. Sociale interactie	VN

Table 9: Workgroup 5, lang= EN

nr	Stud_ID	topic	pref_scr_lang
5	1279806	9. How important are	EN
6	1279954	9. How important are	EN
7	1279982	9. How important are	EN
10	2000815	9. How important are	EN
17	2002685	9. How important are	EN
23	2003951	9. How important are	EN
24	2004318	9. How important are	EN
34	2013126	9. How important are	EN
37	2013829	9. How important are	EN