

March 07 tensor clustering

Yuchen Zeng

March 2019

1 The bottleneck of the `sparse.choosekrl()`

Listed the trace in the appendix A, what we can see is that the long running time of this function comes from the complex iterations. Therefore, what we can do to speed up this function is:

- using Rcpp to improve the speed of each iteration;
- change the for loop into `apply()`.

2 Two problems with `chooseLambda()`

2.1 Problem 1: the range of lambda

Sometimes I found it is pretty hard to find a suitable range for lambda. Instead, we have to try many times to find a suitable range for us to find the best lambda. Therefore, we may delete the parameter to input the range of lambda into this function, instead, we can use `optimize()` to find the best lambda which is different with `sparse.choosekrl()`. (I already tried this method, and it would super slow:(the result looks better but it may depends on the initial value we take.) The new result was listed in the appendices.

2.2 Problem 2: the penalty of non-zeros parameters

I tries many cases, I found that the lambda with the lowest BIC is always 0. I don't know whether it is because of the range I select is not good or the penalty is too small.

3 New problem about the definition of `plot_tensor()`

Now we already can get the same plot if the input `k,r,l` are totally the same with the true `k,r,l` by using a good definition of quantile. But what if the input `k,r,l` is different with the true `k,r,l`. Will the ouput color be totally different from the color of true mus?

4 A reminder for me to change the way to calculate the CER in sparse clustering

5 The results of clustering `dnations.mat`

The code has been put into appendices. My result is listed as follows. First I tried `sparse.choosekrl()`, the range is 2:5, 2:5, 2:5, and the result is 5,5,5. The result below is obtained by using `k,r,l=5`. To avoid the

estimated k, r, l is bounded by 5, then I enlarged the range, but then the program is too slow. And I still have not got the updated k, r, l .

5.1 Clusters of countries

```
cluster 1 :
[1] "Egypt" "India" "Israel"
cluster 2 :
[1] "UK" "USA"
cluster 3 :
[1] "Burma" "Indonesia" "Jordan"
cluster 4 :
[1] "Brazil" "Netherlands"
cluster 5 :
[1] "China" "Cuba" "Poland" "USSR"
```

5.2 Clusters of relationship

```
cluster 1 :
[1] "economicaid" "releconomicaid" "officialvisits" "warning"
      "violentactions"
[6] "militaryactions" "duration" "negativebehavior" "severdiplomatic"
      "expeldiplomats"
[11] "boycottembargo" "aidenemy" "negativecomm" "accusation"
      "protests"
[16] "unofficialacts" "attackembassy" "nonviolentbehavior" "timesincewar"
      "lostterritory"
[21] "dependent" "commonbloc0"
cluster 2 :
[1] "treaties" "conferences" "weightedunvote" "unweightedunvote"
      "intergovorgs"
[6] "ngo" "timesinceally" "independence" "
      blockpositionindex"
cluster 3 :
[1] "relintergovorgs" "relngo" "intergovorgs3" "ngoorgs3"
cluster 4 :
[1] "embassy" "reldiplomacy" "commonbloc1"
cluster 5 :
[1] "reltreaties" "exportbooks" "relexportbooks" "
      booktranslations"
[5] "relbooktranslations" "tourism" "reltourism" "tourism3"
[9] "emigrants" "relemigrants" "emigrants3" "students"
[13] "relstudents" "exports" "relexports" "exports3"
[17] "militaryalliance" "commonbloc2"
```

A Rprofile of sparse.choosekrl()

```
$by.self
      self.time self.pct total.time total.pct
tensorsparse.R#196 393.28 43.17 762.68 83.72
tensorsparse.R#87 269.92 29.63 269.92 29.63
tensorsparse.R#90 72.06 7.91 72.06 7.91
tensorsparse.R#186 54.16 5.94 54.16 5.94
tensorsparse.R#82 32.94 3.62 32.94 3.62
tensorsparse.R#148 27.46 3.01 27.46 3.01
tensorsparse.R#89 25.20 2.77 25.20 2.77
tensorsparse.R#99 19.26 2.11 19.26 2.11
tensorsparse.R#128 6.08 0.67 6.08 0.67
```

tensorsparse.R#86	2.22	0.24	2.22	0.24
tensorsparse.R#79	1.78	0.20	1.78	0.20
tensorsparse.R#160	1.38	0.15	1.38	0.15
tensorsparse.R#80	1.28	0.14	1.28	0.14
tensorsparse.R#225	0.82	0.09	0.82	0.09
tensorsparse.R#151	0.80	0.09	0.80	0.09
tensorsparse.R#121	0.44	0.05	0.44	0.05
tensorsparse.R#123	0.36	0.04	0.36	0.04
tensorsparse.R#122	0.34	0.04	0.36	0.04
tensorsparse.R#142	0.30	0.03	0.30	0.03
tensorsparse.R#145	0.10	0.01	0.10	0.01
tensorsparse.R#143	0.08	0.01	34.80	3.82
tensorsparse.R#206	0.08	0.01	23.36	2.56
tensorsparse.R#211	0.08	0.01	22.06	2.42
simulation.R#46	0.06	0.01	911.02	100.00
tensorsparse.R#201	0.06	0.01	19.58	2.15
tensorsparse.R#129	0.06	0.01	0.06	0.01
tensorsparse.R#204	0.04	0.00	2.56	0.28
tensorsparse.R#202	0.04	0.00	2.02	0.22
tensorsparse.R#138	0.04	0.00	0.04	0.00
tensorsparse.R#293	0.04	0.00	0.04	0.00
tensorsparse.R#303	0.04	0.00	0.04	0.00
tensorsparse.R#235	0.02	0.00	910.84	99.98
tensorsparse.R#212	0.02	0.00	5.82	0.64
tensorsparse.R#215	0.02	0.00	4.52	0.50
tensorsparse.R#207	0.02	0.00	2.76	0.30
tensorsparse.R#298	0.02	0.00	0.04	0.00
tensorsparse.R#124	0.02	0.00	0.02	0.00
tensorsparse.R#125	0.02	0.00	0.02	0.00
tensorsparse.R#130	0.02	0.00	0.02	0.00
tensorsparse.R#140	0.02	0.00	0.02	0.00
tensorsparse.R#159	0.02	0.00	0.02	0.00
tensorsparse.R#162	0.02	0.00	0.02	0.00

\$by.total

	total.time	total.pct	self.time	self.pct
simulation.R#46	911.02	100.00	0.06	0.01
tensorsparse.R#235	910.84	99.98	0.02	0.00
tensorsparse.R#316	910.84	99.98	0.00	0.00
tensorsparse.R#196	762.68	83.72	393.28	43.17
tensorsparse.R#87	269.92	29.63	269.92	29.63
tensorsparse.R#90	72.06	7.91	72.06	7.91
tensorsparse.R#186	54.16	5.94	54.16	5.94
tensorsparse.R#143	34.80	3.82	0.08	0.01
tensorsparse.R#82	32.94	3.62	32.94	3.62
tensorsparse.R#148	27.46	3.01	27.46	3.01
tensorsparse.R#89	25.20	2.77	25.20	2.77
tensorsparse.R#206	23.36	2.56	0.08	0.01
tensorsparse.R#211	22.06	2.42	0.08	0.01
tensorsparse.R#201	19.58	2.15	0.06	0.01
tensorsparse.R#99	19.26	2.11	19.26	2.11
tensorsparse.R#128	6.08	0.67	6.08	0.67
tensorsparse.R#212	5.82	0.64	0.02	0.00
tensorsparse.R#215	4.52	0.50	0.02	0.00
tensorsparse.R#210	2.90	0.32	0.00	0.00
tensorsparse.R#207	2.76	0.30	0.02	0.00
tensorsparse.R#204	2.56	0.28	0.04	0.00
tensorsparse.R#86	2.22	0.24	2.22	0.24
tensorsparse.R#209	2.10	0.23	0.00	0.00
tensorsparse.R#202	2.02	0.22	0.04	0.00
tensorsparse.R#79	1.78	0.20	1.78	0.20

tensorsparse.R#214	1.52	0.17	0.00	0.00
tensorsparse.R#160	1.38	0.15	1.38	0.15
tensorsparse.R#205	1.34	0.15	0.00	0.00
tensorsparse.R#80	1.28	0.14	1.28	0.14
tensorsparse.R#193	1.20	0.13	0.00	0.00
tensorsparse.R#225	0.82	0.09	0.82	0.09
tensorsparse.R#203	0.82	0.09	0.00	0.00
tensorsparse.R#151	0.80	0.09	0.80	0.09
tensorsparse.R#208	0.48	0.05	0.00	0.00
tensorsparse.R#121	0.44	0.05	0.44	0.05
tensorsparse.R#123	0.36	0.04	0.36	0.04
tensorsparse.R#122	0.36	0.04	0.34	0.04
tensorsparse.R#142	0.30	0.03	0.30	0.03
tensorsparse.R#213	0.12	0.01	0.00	0.00
tensorsparse.R#145	0.10	0.01	0.10	0.01
tensorsparse.R#129	0.06	0.01	0.06	0.01
tensorsparse.R#138	0.04	0.00	0.04	0.00
tensorsparse.R#293	0.04	0.00	0.04	0.00
tensorsparse.R#303	0.04	0.00	0.04	0.00
tensorsparse.R#298	0.04	0.00	0.02	0.00
tensorsparse.R#124	0.02	0.00	0.02	0.00
tensorsparse.R#125	0.02	0.00	0.02	0.00
tensorsparse.R#130	0.02	0.00	0.02	0.00
tensorsparse.R#140	0.02	0.00	0.02	0.00
tensorsparse.R#159	0.02	0.00	0.02	0.00
tensorsparse.R#162	0.02	0.00	0.02	0.00

\$by.line

	self.time	self.pct	total.time	total.pct
simulation.R#46	0.06	0.01	911.02	100.00
tensorsparse.R#79	1.78	0.20	1.78	0.20
tensorsparse.R#80	1.28	0.14	1.28	0.14
tensorsparse.R#82	32.94	3.62	32.94	3.62
tensorsparse.R#86	2.22	0.24	2.22	0.24
tensorsparse.R#87	269.92	29.63	269.92	29.63
tensorsparse.R#89	25.20	2.77	25.20	2.77
tensorsparse.R#90	72.06	7.91	72.06	7.91
tensorsparse.R#99	19.26	2.11	19.26	2.11
tensorsparse.R#121	0.44	0.05	0.44	0.05
tensorsparse.R#122	0.34	0.04	0.36	0.04
tensorsparse.R#123	0.36	0.04	0.36	0.04
tensorsparse.R#124	0.02	0.00	0.02	0.00
tensorsparse.R#125	0.02	0.00	0.02	0.00
tensorsparse.R#128	6.08	0.67	6.08	0.67
tensorsparse.R#129	0.06	0.01	0.06	0.01
tensorsparse.R#130	0.02	0.00	0.02	0.00
tensorsparse.R#138	0.04	0.00	0.04	0.00
tensorsparse.R#140	0.02	0.00	0.02	0.00
tensorsparse.R#142	0.30	0.03	0.30	0.03
tensorsparse.R#143	0.08	0.01	34.80	3.82
tensorsparse.R#145	0.10	0.01	0.10	0.01
tensorsparse.R#148	27.46	3.01	27.46	3.01
tensorsparse.R#151	0.80	0.09	0.80	0.09
tensorsparse.R#159	0.02	0.00	0.02	0.00
tensorsparse.R#160	1.38	0.15	1.38	0.15
tensorsparse.R#162	0.02	0.00	0.02	0.00
tensorsparse.R#186	54.16	5.94	54.16	5.94
tensorsparse.R#193	0.00	0.00	1.20	0.13
tensorsparse.R#196	393.28	43.17	762.68	83.72
tensorsparse.R#201	0.06	0.01	19.58	2.15
tensorsparse.R#202	0.04	0.00	2.02	0.22

tensorsparse.R#203	0.00	0.00	0.82	0.09
tensorsparse.R#204	0.04	0.00	2.56	0.28
tensorsparse.R#205	0.00	0.00	1.34	0.15
tensorsparse.R#206	0.08	0.01	23.36	2.56
tensorsparse.R#207	0.02	0.00	2.76	0.30
tensorsparse.R#208	0.00	0.00	0.48	0.05
tensorsparse.R#209	0.00	0.00	2.10	0.23
tensorsparse.R#210	0.00	0.00	2.90	0.32
tensorsparse.R#211	0.08	0.01	22.06	2.42
tensorsparse.R#212	0.02	0.00	5.82	0.64
tensorsparse.R#213	0.00	0.00	0.12	0.01
tensorsparse.R#214	0.00	0.00	1.52	0.17
tensorsparse.R#215	0.02	0.00	4.52	0.50
tensorsparse.R#225	0.82	0.09	0.82	0.09
tensorsparse.R#235	0.02	0.00	910.84	99.98
tensorsparse.R#293	0.04	0.00	0.04	0.00
tensorsparse.R#298	0.02	0.00	0.04	0.00
tensorsparse.R#303	0.04	0.00	0.04	0.00
tensorsparse.R#316	0.00	0.00	910.84	99.98

```
$sample.interval
[1] 0.02
```

```
$sampling.time
[1] 911.02
```

B The R code to cluster dnations.mat

```
if(!require("rmatio")){
  install.packages("rmatio")
  stopifnot(require("rmatio"))
}
source('tensorsparse.R')

dnations = read.mat("dnations.mat")
country = unlist(dnations$countrynames)
relationship = unlist(dnations$relnnames)
att = unlist(dnations$attnames)
dnation_arr = (2*dnations$R-1)*10
dnation_arr[is.nan(dnation_arr)] = 0

krl = sparse.choosekrl(dnation_arr,2:5,2:5,2:5)
#our result in sparse.choosekrl() is 5,5,5
relationship_label = label2(dnation_arr,5,5,5)
relationship_label$Cs
relationship_label$Ds
relationship_label$Es
for (i in 1:5){
  cat("cluster_",i,":\n")
  print(country[relationship_label$Cs==i])
}
for (i in 1:5){
  cat("cluster_",i,":\n")
  print(relationship[relationship_label$Es==i])
}
```

C The simulation result of chooseLambda2()

The result I got by using chooseLambda() is always 0 (may because the range I selected is always bad). And here is the result I got from chooseLambda2():

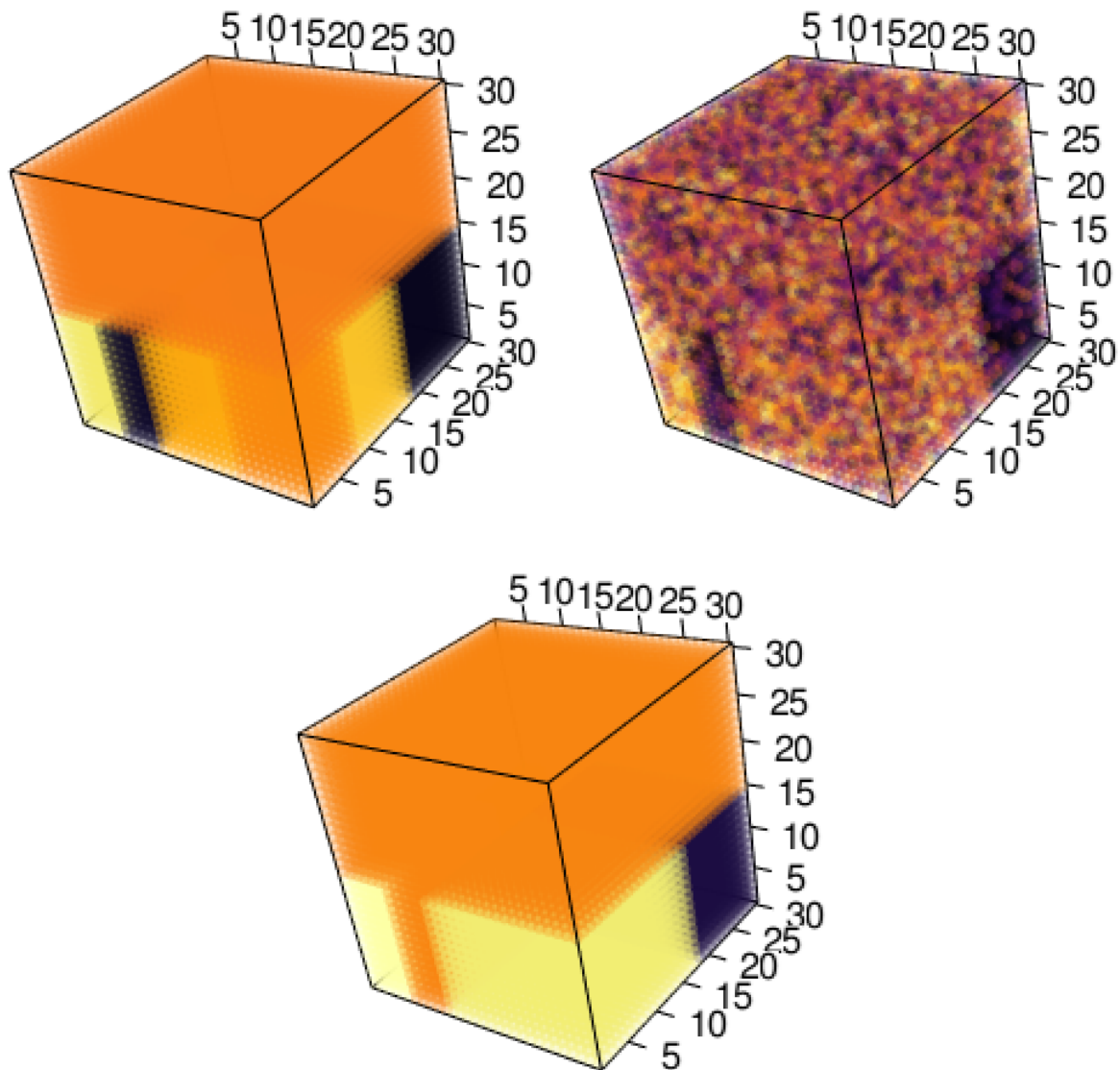


Figure 1: truth,input,output