glm model for binary tensor clustering

Jiaxin Hu

Email: jhu267@wisc.edu

Date: 2019.3.7

1 Construct the glm model

Consider the model:

$$Y = G \times_1 A \times_2 B \times_3 C$$

where $Y \in \mathbb{R}^{2\times3\times4}, G \in \mathbb{R}^{2\times2\times2}$. And $A \in \mathbb{R}^{2\times2}, B \in \mathbb{R}^{2\times3}, C \in \mathbb{R}^{2\times4}$.

Knowing how each element in Y comes from, we have:

$$\begin{aligned} Y[i', j', k'] &= \sum_{ijk} G[i, j, k]A[i, i']B[j, j']C[k, k'] \\ &= G[1, 1, 1]A[1, i']B[1, j']C[1, k'] + \cdots + G[2, 2, 2]A[2, i']B[2, j']C[2, k'] \\ &\textit{where } i' = 1, 2,\ j' = 1, 2, 3\ k' = 1, 2, 3, 4 \end{aligned}$$

From this we can regard every element in G as a coefficient of glm. And for there are 24 element in Y, so there are 24 equations like this.

To simplify the expression of predictor, I use kronecker product rather than outer product to express it. Suppose we have two vector $a = (a_1, \ldots, a_n), b = (b_1, \ldots, b_m)$:

$$a \otimes b = \begin{bmatrix} a_1 b \\ a_2 b \\ \vdots \\ a_n b \end{bmatrix} = \begin{bmatrix} a_1 b_1 \\ \vdots \\ a_1 b_m \\ \vdots \\ a_n b_m \end{bmatrix}$$

So we can use the transport of $A[:,i'] \otimes B[:,j'] \otimes C[:,k']$ as the predictor.

$$A[:,i'] \otimes B[:,j'] \otimes C[:,k'] = \begin{bmatrix} A[1,i']B[1,j']C[1,k'] \\ A[1,i']B[1,j']C[2,k'] \\ A[1,i']B[2,j']C[1,k'] \\ A[1,i']B[2,j']C[2,k'] \\ A[2,i']B[1,j']C[1,k'] \\ A[2,i']B[1,j']C[2,k'] \\ A[2,i']B[2,j']C[1,k'] \\ A[2,i']B[2,j']C[2,k'] \end{bmatrix}_{8\times 1}$$

2 Real data Implement

Using this method to upgrade core tensor, I plug in the real data "dnoation.mat". The result is here:

2.1 Cluster result of country

The cluster result of country in two modes of country is different: In the first mode:

```
Cluster 1 :
"Israel"
Cluster 2 :
"Brazil"  "China"   "Cuba" "Poland"   "USSR"
Cluster 3 :
    "Burma" "Indonesia"    "Jordan"
Cluster 4 :
"Netherlands"          "UK"          "USA"
```

```
Cluster 5 :
"Egypt" "India"
```

In the second mode:

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

2.2 Cluster result of relationship

```
Cluster 1 :
"commonbloc1"
Cluster 2 :
"aidenemy"
Cluster 3 :
       "reltreaties"     "officialvisits"        "conferences"
"exportbooks"
    "relexportbooks"            "warning"     "violentactions"
"militaryactions"
```

```
          "duration"    "severdiplomatic"       "negativecomm"
"protests"
      "unoffialacts" "nonviolentbehavior"            "tourism"
"reltourism"
          "tourism3"           "students"            "exports"
"relexports"
          "exports3"       "intergovorgs"    "relintergovorgs"
"relngo"
     "intergovorgs3"           "ngoorgs3"       "reldiplomacy"
"timesincewar"
     "lostterritory"          "dependent"   "militaryalliance"
Cluster 4 :
"attackembassy"     "emigrants"  "relemigrants"    "emigrants3"
"relstudents"
  "commonbloc2"
Cluster 5 :
        "economicaid"      "releconomicaid"            "treaties"
"booktranslations"
"relbooktranslations"    "negativebehavior"      "expeldiplomats"
"boycottembargo"
         "accusation"      "weightedunvote"    "unweightedunvote"
"ngo"
            "embassy"       "timesinceally"        "independence"
"commonbloc0"
 "blockpositionindex"
```

2.3 Iteration time and Likelihood result

The iteration time to upgrading the core tensor and factor matrices through comparing likelihood function is 4. And the likelihood result is:

```
> test$likelihood
[1] -7607 -2437 -2349  -Inf
```

2.4 The final factorization result:

The final core tensor and factor matrices are:(using cp factorization for initialization and choose $k = 5$)

```
> print(test$modes$U)
, , 1

        [,1]  [,2]   [,3]     [,4]  [,5]
[1,] 14845.1 58179 -46462    160.4  3067
[2,]  -261.1 29540 -39436   9786.1  6908
[3,]  4625.5 24280  -2264 -10861.8 -4957
[4,]  1302.5 -6169   8431  -2386.4 -1866
[5,]  -984.8  3663 -16183  10015.2  1652

, , 2

       [,1]   [,2]   [,3]  [,4]     [,5]
[1,] -10869 -29334 -14534 42102 -15760.4
[2,]  -5956 -22537  28334  5853 -14440.6
```

```
[3,]  -4820   3271 -44594 36710    708.4
[4,]   1584   8067  -3497 -3886   5653.2
[5,]  -1507  -9312   8620  1318  -5320.5

, , 3

         [,1]   [,2]  [,3]  [,4]   [,5]
[1,] -20626.1 -81388 38102 32906 -24237
[2,]   -509.3 -38517 40460  8291  -5686
[3,] -20238.2 -52093 35793 25118 -20899
[4,]  -1485.6   8182 -4783 -4771   2373
[5,]   1965.4 -13540 17862  1422  -3380

, , 4

       [,1]  [,2]    [,3]  [,4]   [,5]
[1,] 7312.0 19410 -7189.8 -5961 1542.5
[2,]  536.4 13091 -7293.3 -4155  218.5
[3,] 7080.9  8013  -294.6 -2805 2030.6
[4,] -277.5 -1834  1118.6  5419  759.9
[5,] -625.8  3626  -588.9 -2553 -836.6

, , 5

        [,1]   [,2]   [,3]    [,4]  [,5]
[1,] -1592.2 -24135   4193 18366.0 -9225
[2,] -1935.9 -16068  15007  3341.0 -5635
```

```
[3,]   457.8  -7031 -12864 15409.2 -2688
[4,]   589.2   4188  -5041   397.3  1849
[5,]  -601.7  -4736   5809  -410.5  2535
```

```
> print(test$modes$M1)
          [,1]     [,2]     [,3]      [,4]     [,5]
 [1,] 0.08813 -0.10965 -0.08715  0.042942  0.08421
 [2,] 0.07523 -0.05696 -0.09269 -0.065791 -0.07977
 [3,] 0.05223 -0.07772 -0.05277 -0.014510  0.06067
 [4,] 0.09497 -0.11899 -0.08441 -0.012048  0.07965
 [5,] 0.07579 -0.10205 -0.07083 -0.010007 -0.04375
 [6,] 0.06963 -0.05196 -0.06408  0.042961 -0.06598
 [7,] 0.04781 -0.06588 -0.05346 -0.074045 -0.05661
 [8,] 0.12710 -0.03418 -0.11028  0.082124 -0.12084
 [9,] 0.05864 -0.01398 -0.08124 -0.051196 -0.07658
[10,] 0.09443 -0.07989 -0.10090  0.220055  0.10278
[11,] 0.13314 -0.16653 -0.10500 -0.001127  0.12463
[12,] 0.08218 -0.11032 -0.06490  0.002397  0.08495
[13,] 0.06396 -0.05017 -0.07588  0.294010  0.08145
[14,] 0.04220 -0.04146 -0.04696  0.169188  0.05929
```

```
> print(test$modes$M2)
           [,1]     [,2]     [,3]     [,4]     [,5]
 [1,] -0.05203 -0.12522 -0.13066 -0.09981  0.11233
 [2,]  0.17844 -0.07801 -0.10456 -0.07511 -0.10088
 [3,]  0.14883 -0.10383 -0.04982 -0.04016  0.07093
 [4,]  0.08356 -0.11185 -0.08382 -0.12406  0.11279
```

```
 [5,]  0.03166 -0.10012 -0.10080 -0.12701 -0.08005
 [6,] -0.02692 -0.05315 -0.07855 -0.09186 -0.05148
 [7,]  0.08362 -0.03758 -0.05602 -0.01056 -0.07253
 [8,] -0.03697 -0.01610 -0.07979 -0.09454 -0.07655
 [9,]  0.20146 -0.02455 -0.05659 -0.05876 -0.09031
[10,] -0.09680 -0.13306 -0.11967 -0.08314  0.11266
[11,]  0.02763 -0.13628 -0.11039 -0.17043  0.13563
[12,]  0.09039 -0.10858 -0.05787 -0.15150  0.08779
[13,] -0.12429 -0.16456 -0.10636 -0.06882  0.11803
[14,] -0.10931 -0.11561 -0.07393 -0.04871  0.08732
```

```
> print(test$modes$M3)
             [,1]       [,2]       [,3]        [,4]       [,5]
 [1,] -0.0126285  0.0034225 -0.022096 -2.096e-02 -0.011693
 [2,] -0.0126285  0.0034225 -0.022096 -2.096e-02 -0.011693
 [3,] -0.0187795  0.0312352 -0.020012 -3.388e-02 -0.005964
 [4,] -0.0156475  0.0244429 -0.014154  6.269e-03  0.005674
 [5,] -0.0048358  0.0131965 -0.014919 -6.734e-03 -0.006203
 [6,] -0.0074858  0.0076286 -0.007677 -7.924e-03  0.009770
 [7,] -0.0190978 -0.0340422 -0.026749 -4.549e-02  0.073399
 [8,] -0.0228967 -0.0356302 -0.026155 -4.591e-02  0.070978
 [9,] -0.0372120  0.0418933 -0.034853 -2.598e-02 -0.028913
[10,] -0.0430219  0.0454082 -0.037686 -2.108e-02 -0.059130
[11,] -0.0128528 -0.0218033 -0.020936  2.613e-02  0.016258
[12,]  0.0012630 -0.0082180 -0.023400  3.409e-02 -0.001411
[13,]  0.0012630 -0.0082180 -0.023400  3.409e-02 -0.001411
[14,]  0.0012630 -0.0082180 -0.023400  3.409e-02 -0.001411
```

```
[15,] -0.0111833  0.0255912 -0.020922 -1.089e-02 -0.017933
[16,] -0.0247709  0.0110586 -0.036225  3.320e-02  0.005762
[17,] -0.0311180  0.0456182 -0.050066 -2.079e-02 -0.022708
[18,] -0.0265015  0.0342071 -0.041909 -2.368e-02 -0.013343
[19,] -0.1975605  0.0252255 -0.193517  3.380e-01  0.417239
[20,] -0.0082539  0.0176492 -0.020043 -2.252e-03  0.003300
[21,] -0.0079407  0.0283345 -0.025231 -8.784e-05 -0.006245
[22,] -0.0092497  0.0075188 -0.019600 -1.703e-02  0.012531
[23,] -0.0224274  0.0014131 -0.030964  5.458e-02  0.047456
[24,] -0.0356421 -0.0242818 -0.046796  6.954e-02  0.138473
[25,] -0.0147423  0.0207512 -0.025962  3.048e-03  0.013516
[26,] -0.0356499  0.0097706 -0.014186 -7.083e-02 -0.053642
[27,] -0.0232979 -0.0057101 -0.008028 -3.154e-02 -0.014435
[28,] -0.0232149 -0.0127704 -0.019894 -3.868e-02  0.026599
[29,] -0.0179329 -0.0144060 -0.017295 -3.544e-02  0.024019
[30,] -0.0247843 -0.0087316 -0.018167 -2.849e-02  0.022834
[31,] -0.0676029 -0.0274707 -0.034429  3.653e-02  0.091536
[32,] -0.1383545 -0.0609962 -0.068613  9.157e-02  0.216511
[33,] -0.0669956 -0.0315785 -0.034876  2.890e-02  0.097344
[34,] -0.0436692 -0.0080430 -0.024399  3.202e-02  0.023658
[35,] -0.0724134 -0.0407276 -0.028547  7.972e-02  0.087258
[36,] -0.0132616  0.0073431 -0.014732 -1.395e-02  0.005304
[37,] -0.0137331  0.0066769 -0.010089  5.917e-03  0.004816
[38,] -0.0157570  0.0099447 -0.012525  5.214e-03 -0.012786
[39,] -0.0282343 -0.0047976 -0.002075 -3.373e-02  0.009052
[40,] -0.0331902 -0.0256068  0.011590  4.636e-02  0.015309
[41,] -0.0479261 -0.0065949 -0.007689 -5.679e-02  0.007276
```

```
[42,] -0.0475929 -0.0129216  0.007553  4.208e-02  0.002537
[43,] -0.0354055  0.0030353  0.000235 -8.854e-03  0.005471
[44,] -0.0550333 -0.0035234  0.003385 -3.675e-03 -0.001383
[45,] -0.0033996  0.0203724 -0.002721 -1.424e-02 -0.022675
[46,] -0.0058231  0.0485326 -0.015959  8.921e-02 -0.002222
[47,] -0.0051393 -0.0007529 -0.012804 -1.265e-03  0.001288
[48,] -0.0212430  0.0613898 -0.021915 -4.413e-02  0.005902
[49,] -0.0006062 -0.0158026 -0.027628  1.829e-02  0.024157
[50,] -0.0316134 -0.0032394 -0.024369  3.225e-02  0.003422
[51,] -0.0079123  0.0570307 -0.025430 -3.831e-02  0.021912
[52,] -0.0256792  0.0626403 -0.045389 -4.329e-02  0.036485
[53,] -0.0104421  0.0394940 -0.022787 -2.660e-02 -0.023146
[54,] -0.0184102 -0.0160770 -0.011991 -7.006e-03  0.039761
[55,] -0.0126853  0.0284074 -0.013651 -7.488e-03 -0.168511
[56,]  0.0011815 -0.0657542 -0.003013  5.370e-02  0.234597
```

3 Real data Implement 2

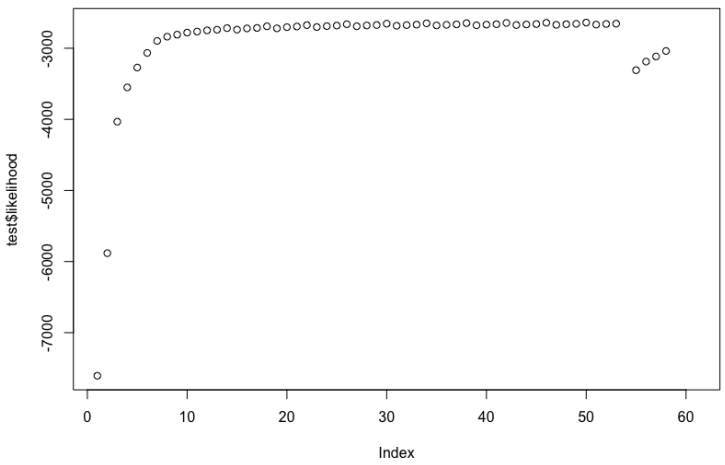
In this part, I use *tucker* to do the initialization of the factor matrices and core tensor. The dimension of the core tensor of each mode are $3, 3, 5$.

3.1 Likelihood

The iteration time is 14 and I printed out each likelihood result when a factor matrix or core tensor is upgraded. So there are total $4 * 14 + 1 = 57$ likelihood results. The numerical result is here:

```
[1] -7605.654 -5882.085 -4033.025 -3551.010
 [5] -3271.709 -3065.977 -2897.183 -2838.021
 [9] -2809.175 -2779.908 -2765.267 -2749.011
[13] -2739.845 -2717.424 -2738.187 -2721.586
[17] -2715.191 -2691.319 -2720.357 -2703.647
[21] -2694.584 -2675.059 -2702.236 -2689.608
[25] -2681.531 -2662.357 -2691.006 -2680.226
[29] -2673.520 -2654.955 -2684.273 -2674.649
[33] -2668.389 -2650.219 -2679.814 -2670.894
[37] -2664.758 -2646.826 -2676.740 -2668.202
[41] -2662.249 -2644.399 -2674.101 -2665.684
[45] -2660.141 -2642.209 -2670.828 -2662.386
[49] -2657.454 -2639.424 -2667.570 -2658.981
[53] -2654.457      -Inf -3308.733 -3187.815
[57] -3115.858
```

And the plot is :



Issues:

- I don't know why the likelihood can suddenly drop to negative infinity and come back after another upgrading.
- It seems this methods depends on the choice of core tensor's dimension. $k = c(3, 3, 5)$ is one choice with a relatively good result.
- It seems *tucker* always give the same factorization result when given dimensions of core tensor.
- Actually the final result of likelihood is not better than using *cp* factorization.

3.2 Clustering result

I just put the cluster result of country. The first mode result is:

```
Cluster 1: Brazil Netherlands UK USA
Cluster 2: Burma Israel Jordan
Cluster 3: China Cuba Egypt India Indonesia Poland USSR
```

The second mode result is:

```
Cluster 1: China Cuba Poland USSR
Cluster 2: Brazil Egypt India Israel Netherlands UK USA
Cluster 3: Burma Indonesia Jordan
```