TopicModel in C++

Original C code: www..cs.princeton.edu/~blei/lda-c/
My C++ implementation hosted on github: github.com/chui0107/TopicModel

1. Code Execution:

The “main” function of the program is in TopicModel.cpp, where settings for topic estimate and topic inference are packed into two structs “TopicModelEstimate” and “TopicModelInference” that are declared in “TopicModel.h”, their 1-on-1 mappings to the original parameters in the C code are commented in the source code in the “main” function.

```
TopicModelEstimate estimateSettings(initialAlpha, topics, varMaxIter, varConvergence,
emMaxIter, emConvergence, alpha, dataPath, topicInit, outputPath);

TopicModelInference inferenceSettings(varMaxIter, varConvergence, emMaxIter,
emConvergence, alpha, dataPath, modelPath, outputPath);
```

There are two functions in the TopicModel class, “run_em” and “infer” to run the “TopicEstimate” and “TopicInference”, respectively. Simply provide the processed corpus to the functions along with its corresponding settings, where the corpus could be generated by passing the path of the raw data file “*.dat” e.g. ap.dat to the function “read_data” declared in “lda-data.h”

```
corpus* newsCorpus = read_data((char*)estimateSettings.mDataPath.c_str());

topicModel.run_em(newsCorpus, estimateSettings);

topicModel.infer(newsCorpus, inferenceSettings);
```

After the model is saved in the output folder that was generated in “topic estimate”, there is a function called “GetTopNTerms” to print out the top N terms for each topic.

```
std::vector<std::vector<std::string> > ret =
topicModel.GetTopNTerms(topicModelTopTermSettings, n);
```

2. Interesting findings:

1. Running “topic estimate” with the same dataset “ap.dat” and the same set of parameters yield different “final.beta” which contains the log of topic distribution each time under the original C implementation, but it is consistent under the replicated c++ implementation.
2. Running “topic inference” with the same dataset “ap.dat” and the same set of parameters yield consistent results for both the C and C++ implementation.s

Parameter for “topic estimate”

Name	Value
initialAlpha	0.5
topics	30
varMaxIter	20
varConvergence	1e-6
emMaxIter	100
emConvergence	1e-4
alpha	estimate
dataPath	example/ap/ap.dat
topicInit	random
outputPath	output

NOTE: timestamp on the files is to indicate they are different files, generated in different trails. Experiments were done more than ten times but results of only two trails are illustrated.

C++ topic estimate first trail:

```
congs-MacBook-Pro:output conghui$ ls -l | grep final.beta
-rw-r--r--  1 conghui  staff  4800523 Oct  5 15:17 final.beta
```

output from "sed -n 1p final.beta | less

```
 -4.9348995990 -4.8142296392 -27.3051388256 -6.1161898656 -5.1993605261 -5.71142
57254 -6.8761276210 -7.8465598183 -5.8442226774 -35.5341742231 -5.2101796428 -5.
6341652185 -9.8923380071 -10.7252605055 -7.4133340157 -6.6180523665 -88.20348224
91 -8.1243804540 -89.5126999785 -5.8794605813 -6.0439915235 -45.1600950871 -6.34
41330477 -5.9410184415 -7.7741291386 -6.7893930333 -7.5969250661 -15.8570003491
-6.8887953645 -7.4582929655 -39.0347754443 -6.0615133748 -6.5024372550 -6.575956
9218 -7.0706390777 -6.2757840113 -6.1385602923 -7.3478751111 -7.1111916204 -7.00
73006930 -5.5224646674 -8.4072651962 -5.9095929693 -6.3882136903 -5.1763697883 -
51.7988720769 -51.5822727153 -7.2514330102 -23.9321443478 -7.4185584331 -13.7804
240374 -7.2820616015 -5.7235831028 -16.4346852301 -89.0965053116 -7.2119804323 -
7.4911592889 -6.1150259610 -8.5721282805 -6.4183435189 -6.9569234295 -6.44236676
51 -6.3077235445 -6.0452677334 -6.5819918365 -7.4329515369 -8.9279455694 -6.9450
827350 -6.6401285925 -101.4116846060 -12.1889074946 -6.7115196202 -7.4620313429
-22.1290843187 -10.1037682589 -8.8833377993 -14.3612407356 -7.9654956854 -18.406
8954360 -6.4164739736 -92.6163738317 -7.9311168494 -7.7548152512 -35.5330727276
-6.5827092161 -7.1229151161 -7.0894272069 -75.8613257317 -7.1817667173 -6.824216
2698 -41.5514422611 -6.5465477055 -48.7207837075 -66.7951384758 -7.4910595583 -7
.5347723137 -80.6132640774 -7.1966912550 -54.5926572566 -46.2024950992 -6.673720
6930 -7.3447786453 -7.8877569557 -36.9825199267 -8.7380974764 -17.2247366434 -7.
5233672841 -6.4818328256 -13.4117931611 -7.6749169005 -6.7932700181 -7.182391433
0 -6.7643675099 -5.9563228624 -6.9839259009 -9.4341757182 -7.3513278730 -7.79852
38879 -7.5452567412 -78.5028202380 -8.2616022138 -5.7168144776 -26.7961046002 -6
.2475683742 -7.5107909155 -5.5109761702 -7.1300075545 -8.8076337846 -6.013175107
:
```

C++ topic estimate second trail:

```
congs-MacBook-Pro:output conghui$ ls -l | grep final.beta
-rw-r--r--  1 conghui  staff  4800523 Oct  5 15:49 final.beta
```

output from "sed -n 1p final.beta | less

```
 -4.9348995990 -4.8142296392 -27.3051388256 -6.1161898656 -5.1993605261 -5.71142
57254 -6.8761276210 -7.8465598183 -5.8442226774 -35.5341742231 -5.2101796428 -5.
6341652185 -9.8923380071 -10.7252605055 -7.4133340157 -6.6180523665 -88.20348224
91 -8.1243804540 -89.5126999785 -5.8794605813 -6.0439915235 -45.1600950871 -6.34
41330477 -5.9410184415 -7.7741291386 -6.7893930333 -7.5969250661 -15.8570003491
-6.8887953645 -7.4582929655 -39.0347754443 -6.0615133748 -6.5024372550 -6.575956
9218 -7.0706390777 -6.2757840113 -6.1385602923 -7.3478751111 -7.1111916204 -7.00
73006930 -5.5224646674 -8.4072651962 -5.9095929693 -6.3882136903 -5.1763697883 -
51.7988720769 -51.5822727153 -7.2514330102 -23.9321443478 -7.4185584331 -13.7804
240374 -7.2820616015 -5.7235831028 -16.4346852301 -89.0965053116 -7.2119804323 -
7.4911592889 -6.1150259610 -8.5721282805 -6.4183435189 -6.9569234295 -6.44236676
51 -6.3077235445 -6.0452677334 -6.5819918365 -7.4329515369 -8.9279455694 -6.9450
827350 -6.6401285925 -101.4116846060 -12.1889074946 -6.7115196202 -7.4620313429
-22.1290843187 -10.1037682589 -8.8833377993 -14.3612407356 -7.9654956854 -18.406
8954360 -6.4164739736 -92.6163738317 -7.9311168494 -7.7548152512 -35.5330727276
-6.5827092161 -7.1229151161 -7.0894272069 -75.8613257317 -7.1817667173 -6.824216
2698 -41.5514422611 -6.5465477055 -48.7207837075 -66.7951384758 -7.4910595583 -7
.5347723137 -80.6132640774 -7.1966912550 -54.5926572566 -46.2024950992 -6.673720
6930 -7.3447786453 -7.8877569557 -36.9825199267 -8.7380974764 -17.2247366434 -7.
5233672841 -6.4818328256 -13.4117931611 -7.6749169005 -6.7932700181 -7.182391433
0 -6.7643675099 -5.9563228624 -6.9839259009 -9.4341757182 -7.3513278730 -7.79852
38879 -7.5452567412 -78.5028202380 -8.2616022138 -5.7168144776 -26.7961046002 -6
.2475683742 -7.5107909155 -5.5109761702 -7.1300075545 -8.8076337846 -6.013175107
:
```

Conclusion: "topic estimate" under the C++ implementation topic distribution is consistent throughout trails

C topic estimate first trail:

```
congs-MacBook-Pro:output conghui$ ls -l | grep final.beta
-rw-r--r--  1 conghui  staff  4792143 Oct  5 16:09 final.beta
```

output from "sed -n 1p final.beta | less

```
 -6.2579942381 -5.3745717130 -14.9685761529 -5.7149557829 -5.0490433637 -5.86126
13568 -6.1323898735 -6.5404827426 -5.1072867492 -7.0591548451 -5.3858114162 -6.7
078190766 -48.9473276753 -5.5966288989 -7.8433750411 -5.9353861231 -41.755276182
6 -8.4815574651 -57.4355424816 -5.9686405053 -6.5343383450 -7.7656158139 -6.7138
251657 -7.1615449286 -6.8336394411 -6.7694757565 -8.0097505485 -6.4178008305 -5.
2632331283 -6.8565058268 -9.2108272259 -6.1513827902 -8.8133647384 -6.3871842490
 -7.6735333765 -6.8174780174 -8.6538575802 -7.3383781695 -5.9128339584 -6.766538
9678 -6.2569600953 -22.8697586252 -6.2022789643 -5.7823066620 -6.5214645365 -6.8
650722513 -6.5923865498 -5.3548834964 -66.3426143821 -8.5220380495 -10.358707987
2 -6.5620453905 -6.5582489478 -7.3565258753 -41.0665985214 -8.0663150179 -6.2186
584433 -6.2513334445 -5.3773213181 -7.2595202746 -6.9798075366 -6.9596786282 -7.
0207206606 -6.5146378903 -6.8359838138 -6.2884018791 -11.8607298760 -6.996317538
9 -6.3919992036 -8.5672445423 -5.7710716431 -9.5578195693 -12.8793827291 -7.2772
732966 -9.1439932500 -17.6611687977 -7.3985458976 -6.0633723063 -24.9206210604 -
7.4227068147 -8.1806033799 -6.7628927388 -5.3045489768 -9.0460386203 -7.79070415
12 -7.6879351837 -6.6512287682 -7.6813576374 -7.3082978732 -6.8785399091 -179.52
87818859 -6.7686734675 -8.0692230877 -6.7449001621 -6.6571425103 -7.0640380782 -
7.7041049092 -30.0766453182 -7.8154000544 -6.7019737977 -7.0596059034 -5.8894260
945 -8.1708114418 -8.2083719038 -7.4874195483 -7.3376574609 -7.5398936154 -6.991
3949589 -7.2845472711 -7.2694708175 -7.1957205625 -7.4001028213 -7.1131033178 -6
.6761307762 -7.5771978237 -7.3733729651 -6.5881438298 -159.7393769574 -7.3230435
475 -8.9495450152 -7.9646321055 -6.3032918622 -9.0743627920 -7.9854329069 -6.240
4303215 -6.9085290579 -7.1815112614 -8.8796611826 -6.2166733869 -8.5377886175 -1
:
```

C topic estimate second trail:

```
congs-MacBook-Pro:output conghui$ ls -l | grep final.beta
-rw-r--r--  1 conghui  staff  4792614 Oct  5 16:20 final.beta
```

output from "sed -n 1p final.beta | less

```
 -6.0585090506 -8.0614345091 -17.2464587648 -4.9312866783 -7.8724348248 -4.92768
34313 -6.1622800846 -24.0390256785 -5.8058398009 -6.6183797660 -5.9464229063 -6.
8154526265 -7.9699014680 -5.8681379241 -7.8858973312 -4.8127418160 -22.819767503
4 -12.0125737210 -77.8504790343 -6.2993768188 -5.4274643150 -9.9855942067 -5.880
1616478 -5.8535911308 -6.8838039818 -8.0945390718 -5.8740239039 -7.2459068363 -7
.3514335031 -6.8489091233 -70.6541878131 -6.9408946229 -6.0408921261 -6.87562284
71 -6.5629888835 -6.2129339275 -5.7479243187 -6.2320123229 -7.7791822940 -7.0348
213225 -6.4213213822 -19.4088404365 -6.3253201050 -22.4301253274 -26.3828778370
-50.9799008540 -5.9776261117 -5.3885143650 -5.6118764245 -5.9280154836 -12.27232
11031 -6.9489570192 -6.3872058157 -6.7953189621 -12.2362345022 -5.7004436380 -8.
3759963602 -8.6382666495 -8.3503573362 -6.7709110435 -5.4579676066 -6.9597755683
 -7.9527861112 -6.3417287893 -5.9660238850 -9.9832504463 -17.6617561202 -6.93899
34004 -6.7007284803 -5.5849435671 -7.5761987764 -7.0278785947 -7.4188787722 -15.
8011103170 -8.1961808453 -33.1658017237 -9.1328282644 -7.8132087601 -89.91960383
91 -17.8480638191 -80.1006278548 -9.8272798478 -7.7714578667 -57.2131983708 -6.0
020671766 -4.9318790921 -32.5408616803 -53.6428852686 -7.0406382887 -5.933774384
4 -186.9002110926 -7.1037274473 -40.1044295235 -73.8645828709 -6.9534444028 -6.8
411142903 -10.0668002523 -7.5278071719 -7.2350420815 -8.6167935644 -6.6885237539
 -8.6530882647 -7.8029768381 -45.9828482726 -7.9537876570 -7.2197609868 -6.54972
34398 -5.9044604356 -29.1729219810 -13.5333333585 -7.9705090486 -6.6207894910 -5
.9084147830 -7.3069043070 -6.4021167803 -7.2852923499 -7.2441439659 -6.351570213
7 -6.6372977972 -5.2695528088 -7.2128770095 -8.4513145329 -18.2157621402 -8.1542
238440 -7.1833664671 -7.1136426279 -7.5610941890 -14.2137060358 -7.0985647599 -7
:
```

Conclusion: "topic estimate" under the C implementation topic distribution is inconsistent throughout trails, in facts, it is distinctive in each trail.

Parameter for “topic inference”

Name	Value
varMaxIter	-1
varConvergence	1e-6
emMaxIter	100
emConvergence	1e-4
alpha	estimate
dataPath	example/ap/ap.dat
modelPath	output/final
outputPath	inferred

C++ topic inference first trail

```
congs-MacBook-Pro:topicmodel conghui$ ls -l | grep inferred-gamma.dat
-rw-r--r--   1 conghui  staff  882918 Oct  5 15:55 inferred-gamma.dat
```

output from sed -n 1p inferred-gamma.dat | less

```
0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.
0155603029 118.5557554556 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.
0155603029 0.0155603029 0.0155603029 0.0155603029 92.8065824848 0.0155603029 0.0
155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029 51.68
43429683 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029
(END)
```

C++ topic inference second trail

```
congs-MacBook-Pro:topicmodel conghui$ ls -l | grep inferred-gamma.dat
-rw-r--r--   1 conghui  staff  882918 Oct  5 15:57 inferred-gamma.dat
```

output from sed -n 1p inferred-gamma.dat | less

```
0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.
0155603029 118.5557554556 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.
0155603029 0.0155603029 0.0155603029 0.0155603029 92.8065824848 0.0155603029 0.0
155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029 51.68
43429683 0.0155603029 0.0155603029 0.0155603029 0.0155603029 0.0155603029
(END)
```

C topic inference first trail

```
congs-MacBook-Pro:lda-c-master conghui$ ls -l | grep inferred-gamma.dat
-rw-r--r--@  1 conghui  staff  883165 Oct  5 16:23 inferred-gamma.dat
```

output from sed -n 1p inferred-gamma.dat | less

```
0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 53
.5459074451 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0
173160248 0.0173160248 22.8654145572 0.0173160248 0.0173160248 0.0173160248 0.01
73160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173
160248 0.0173160248 0.0173160248 186.6406260719 0.0173160248 0.0173160248
(END)
```

C topic inference second trail

```
congs-MacBook-Pro:lda-c-master conghui$ ls -l | grep inferred-gamma.dat
-rw-r--r--@  1 conghui  staff  883165 Oct  5 16:25 inferred-gamma.dat
```

output from sed -n 1p inferred-gamma.dat | less

```
0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 53
.5459074451 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0
173160248 0.0173160248 22.8654145572 0.0173160248 0.0173160248 0.0173160248 0.01
73160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173
160248 0.0173160248 0.0173160248 186.6406260719 0.0173160248 0.0173160248
(END)
```

Conclusion: The results of running topic inference on both C and C++ implementations are consistent. However, one thing to note is that since the “topic inference” use the model computed from “topic estimate” e.g final.other and final.beta, “topic inference” results would vary for the C implementation if its “topic estimate” runs and produces a different model again. However, since the results of the C++ implementation is consistent, it doesn't have this problem.

Example of varied results of C implementation's "topic inference" after its "topic estimate" runs

1. produce a new model

```
congs-MacBook-Pro:output conghui$ ls -l | grep final.beta
-rw-r--r--  1 conghui  staff  4803014 Oct  5 16:35 final.beta
```

output from "sed -n 1p final.beta | less

```
 -6.2807690809 -5.8501713315 -7.8877723516 -5.9845230566 -5.8923018448 -5.632717
2230 -4.9043922293 -6.7713078009 -5.6320654888 -5.7479563293 -5.4555422918 -6.67
70759068 -7.8458726780 -6.4110947338 -6.8294372343 -6.7227348586 -91.3783931051
-7.1502533369 -138.4545148470 -7.0865800335 -5.7025178528 -6.9813933814 -7.78416
45597 -6.7650093066 -5.8895115508 -6.7262326931 -6.5474547880 -4.8530396126 -7.7
163997703 -6.4831279409 -5.0503027934 -8.2415826367 -6.0985955379 -6.0624984690
-6.9977588671 -6.8289056552 -6.8514457849 -6.9631766466 -6.9436722941 -5.8491660
110 -7.0480091617 -31.0313858876 -8.4358228683 -6.8484777236 -6.1229071736 -7.02
61842877 -14.9844797838 -6.3136299800 -23.4004636517 -7.6436051905 -80.678865607
1 -8.3468899852 -6.7850773629 -8.9731860321 -78.3193748647 -7.3174911897 -6.8886
112268 -8.3598388081 -6.1743537045 -6.6479327076 -8.3331982182 -8.4366014197 -7.
3888611804 -8.8106715457 -6.1963269990 -9.2360263215 -8.0267746728 -7.0268341657
 -7.7840971445 -5.9236429749 -8.6040411976 -35.7449233521 -9.3396802173 -7.59817
89611 -9.4881643853 -30.4660527414 -11.3006321051 -6.8285400397 -8.6264964338 -7
.5759556383 -17.9755939000 -7.0636856797 -6.9323903006 -43.1375358051 -7.0430748
019 -37.2681682403 -5.6406441290 -5.7908076308 -7.4792558584 -6.8909596089 -93.3
151348892 -6.9374979255 -49.3898310892 -5.9371947102 -7.2160286389 -12.742860647
8 -4.8100241889 -6.8028242071 -7.4421676380 -7.6833634593 -7.1712845346 -6.32707
01913 -8.9255018202 -7.5642962876 -68.8476187792 -7.4141224870 -7.3363158721 -6.
7456037462 -6.1772856653 -8.1161750443 -21.9003619292 -7.0459314651 -17.93918137
53 -8.2588385328 -8.0431306795 -8.5132334955 -7.3722156812 -13.0632556179 -13.23
04280302 -20.2455246278 -7.1337311725 -8.4572244692 -9.6213302646 -6.9563233853
-6.2297689939 -6.4700588469 -16.8147559329 -7.8527330069 -8.2834734275 -6.592335
:
```

2. run the inference on it again

```
congs-MacBook-Pro:lda-c-master conghui$ ls -l | grep inferred-gamma.dat
-rw-r--r--@  1 conghui  staff  883041 Oct  5 16:37 inferred-gamma.dat
```

sed -n 1p inferred-gamma.dat | less

```
0.0165441409 2.3207845047 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.
0165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 221.
0354025438 39.6934453741 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.0
165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.016
5441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409
(END)
```

Results varied from the results of the second trail of “topic inference” above

side by side view:

i. results from previous inference

```
0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 53
.5459074451 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0
173160248 0.0173160248 22.8654145572 0.0173160248 0.0173160248 0.0173160248 0.01
73160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173160248 0.0173
160248 0.0173160248 0.0173160248 186.6406260719 0.0173160248 0.0173160248
(END)
```

ii. results from the current inference

```
0.0165441409 2.3207845047 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.
0165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 221.
0354025438 39.6934453741 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.0
165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.016
5441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409 0.0165441409
(END)
```

3.Speculation for different results of the “topic estimate” in two implementations:

The C code was compiled with -03 standard and the C++ code was compiled with -std=c+11 standard under the Mac's version gcc compiler (bundled with Xcode). The different results could be generated from different standards' between two languages, I couldn't safely assume that, for example, a float number is guaranteed to be at least 4 bytes under two languages on my Macbook Pro, implementations might vary, and there might be overflow or underflow in the computation that results in different results from two languages.

Speculation for different results of the “topic estimate” in each trail under the C implementation:

To explained the varied results between each C's topic estimate trails, I first speculated that since in the VEM algorithm from the original paper, it used the “Mersenne Twister” (in cokus.cpp), this might result in different results for the C implementation. However, this theory didn't quite hold since the C++ implementation I ported used the same twister at the same spot, if this would be the reason, the output of the C++ implementation would have varied as well.

After carefully inspecting the code base, the most probable culprit is those uninitialized variables that the original users use in the computation. As far as I remember, the C compiler doesn't automatically initialize those variables that aren't manually initialized, e.g. set 0 to integer type and etc. Thus this might result in garbage values being used which leads to different results in each computation in the C runtime. On the other hands, the C++ compiler does a better job at enforcing an initial value for each declared variables even though they aren't properly initialized by the authors. This explained the consistency in the final beta computation.

However, this speculation couldn't be verified because I don't know the mathematics behind the

algorithm well, in order to maintain the correctness of the algorithm, I minimized my modification to the code and avoided touching any computation related parts.

Parameter for “Top term for each topic”

Name	Value
modelPath	output/final.beta
VocabularyPath	example/ap/vocab.txt
n	10

Output from top 10 terms for each topic:

```
g++ -std=c++11 -Wall -g lda-data.o TopicModel.o lda-model.o utils.o cokus.o lda-alpha.o -o lda -lm
reading data from example/ap/ap.dat
number of docs    : 2246
number of terms   : 10473
topic 0: new i fire york year years john city first two
topic 1: government united china states official drug last people chinese news
topic 2: iraq kuwait iraqi saudi gulf united war military iran oil
topic 3: plant environmental company state years waste year new computer water
topic 4: people state india police two troops government killed moslem army
topic 5: air flight plane space two airlines aircraft pilots airline planes
topic 6: dollar yen late gold new london west bid central rain
topic 7: police people two killed city authorities officials man miles three
topic 8: trade billion percent japan states united year japanese million farmers
topic 9: soviet israel bush president summit united meeting union europe israeli
topic 10: aids panama drug noriega patients new disease virus health system
topic 11: million company billion sales corp year percent new workers inc
topic 12: people work workers computer new percent report jobs years business
topic 13: women network television abortion president rights i people cbs tv
topic 14: students church people pope john south government catholic officials years
topic 15: bill house senate congress committee tax budget rep federal sen
topic 16: school drug students state cocaine board i drugs student attorney
topic 17: i think get dont people like im says years just
topic 18: percent year rate billion last rates months increase first rose
topic 19: i years people milken family home species mrs two last
topic 20: south africa aid government african president rebels contra nicaragua contras
topic 21: stock market index trading exchange million new prices points shares
topic 22: government people party president national military police elections political opposition
topic 23: court judge case attorney trial federal charges state prison law
topic 24: i people city years black nbc two percent first mayor
topic 25: cents cent oil lower futures higher new prices market art
topic 26: bush dukakis campaign i president jackson democratic republican new presidential
topic 27: defense military navy president north pentagon house reagan billion general
```

To-Do

1. The algorithm to print the top N term for each topic could be further optimized.

2. So far I've only tested the algorithm on the original dataset that came with the source code, I will further test the algorithm on a dataset I've collected personally

3. In the original paper, there is a native C++ implementation for the LDA model with Gibbs Sampling, e.g. the current one uses VEM algorithm. I will further dig into that one to come up with a tutorial it if needed.