CS 510 Advanced Information Retrieval

MP3

Problem 1:

(a)

Choose the probability of generating the word from the background model to be zero i.e. let $\boldsymbol{\lambda = 0}$. Then log P (D | $\boldsymbol{\theta, \pi}$) from equation (2) is equal to log P(D | $\boldsymbol{\theta, \pi}$) from equation (1). Hence EM algorithm with background language model will reduce to EM algorithm without the background language model.

(b)

Probability of word using the background language model D is given by:

P (w| D) = $\frac{\sum_{d \in D} c(w,d)}{\sum_{w \in V} \sum_{d \in D} c(w,d)}$ where the denominator is equivalent to the size of the entire corpus.

(c)

Hypothesis: When λ is close to one, topics will have top words as very discriminative words.

When the value of λ is large, topics will have very meaningful top words and when λ is small, topics will have highly frequent stop words as top words.

Experiment: Vary the value of λ from zero to one and compute the top words for each of the topics. Compare the top words for various values of λ. For lower values of λ, top words will have more stop words.

Problem 2:

(a)

E step:

$$\mathrm{P}\left(y_{i,j} = 1 \,\middle|\, D, \theta^{(n)}, \pi^{(n)}\right) = \frac{(1-\lambda)\sum_{k=1}^{K} P\left(z_{i,j}=k \mid \pi_i^{(n)}\right) P(w|\theta^{(k)})}{\lambda P(W|D) + (1-\lambda)\sum_{k=1}^{K} P\left(z_{i,j}=k \mid \pi_i^{(n)}\right) P(w|\theta^{(k)})}$$

Set $q_y\left(y_{i,j} = 1\right) = \mathrm{P}\left(y_{i,j} = 1 \,\middle|\, D, \theta^{(n)}, \pi^{(n)}\right)$

$$\mathrm{P}\left(z_{i,j} = k \,\middle|\, y_{i,j} = 1, \theta^{(n)}, \pi^{(n)}\right) = \frac{P\left(z_{i,j}=k \mid \pi_i^{(n)}\right) P\left(w_{i,j}|\theta_k^{(n)}\right)}{\sum_{k'=1}^{K} P\left(z_{i,j}=k' \mid \pi_i^{(n)}\right) P\left(w|\theta_{k'}^{(n)}\right)}$$

Set $q_{z|y}\left(z_{i,j} = k\right) = \mathrm{P}\left(z_{i,j} = k \,\middle|\, y_{i,j} = 1, \theta^{(n)}, \pi^{(n)}\right)$

Now,

- $n_{d,k} = \sum_{w\in d} c(w,d)\, q_y\left(y_{d,w} = 1\right) q_{z|y}\left(z_{d,w} = k\right)$
- $n_{w,k} = \sum_{d\in D} c(w,d)\, q_y\left(y_{d,w} = 1\right) q_{z|y}\left(z_{d,w} = k\right)$

(b)

M step:

For updating $\pi_d^{(n+1)}$:

$$\mathrm{P}\left(z_{d,w} = k \,\middle|\, \pi_d^{(n+1)}\right) = \frac{n_{d,k}}{\sum_{k'=1}^{K} n_{d,k'}} = \frac{\sum_{w\in d} c(w,d)\, q_y\left(y_{d,w}=1\right) q_{z|y}\left(z_{d,w}=k\right)}{\sum_{k'=1}^{K}\sum_{w\in d} c(w,d)\, q_y\left(y_{d,w}=1\right) q_{z|y}\left(z_{d,w}=k\right)}$$

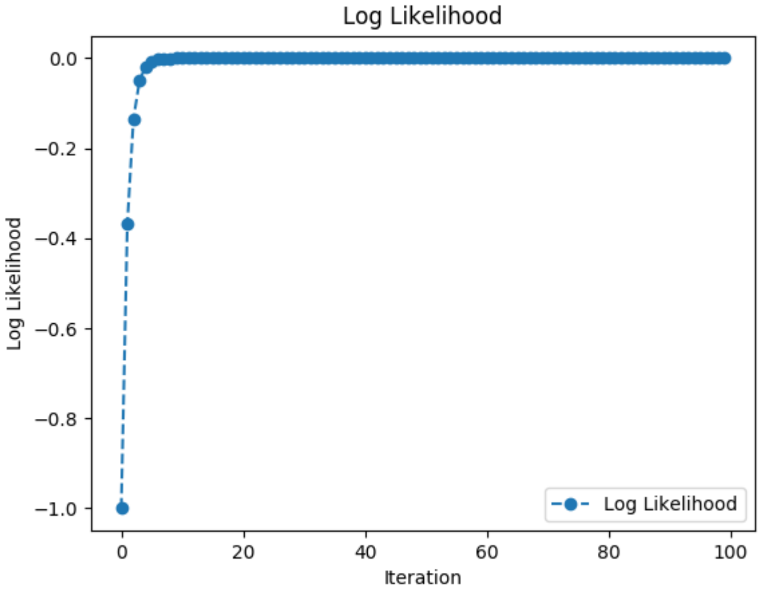
For updating $\theta_k^{(n+1)}$:

$$\mathrm{P}\left(w \,\middle|\, \theta_k^{(n+1)}\right) = \frac{n_{w,k}}{\sum_{w'\in V} n_{w',k}} = \frac{\sum_{d\in D} c(w,d)\, q_y\left(y_{d,w}=1\right) q_{z|y}\left(z_{d,w}=k\right)}{\sum_{w'\in V}\sum_{d\in D} c(w,d)\, q_y\left(y_{d,w}=1\right) q_{z|y}\left(z_{d,w}=k\right)}$$

Problem 3:

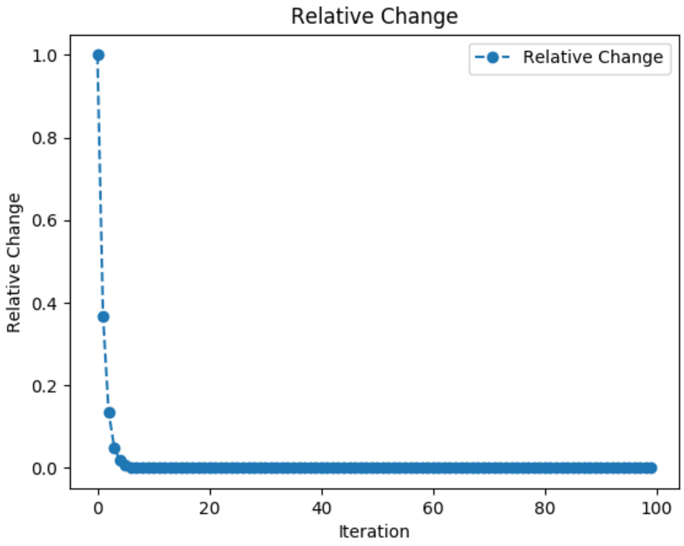
(b)

The value of the log likelihood must increase at each iteration. We expect that the value of the log likelihood would initially increase significantly with each update and then stabilizes when the no. of iterations grow. Below is an estimated graph of change in log likelihood with increase in iteration:



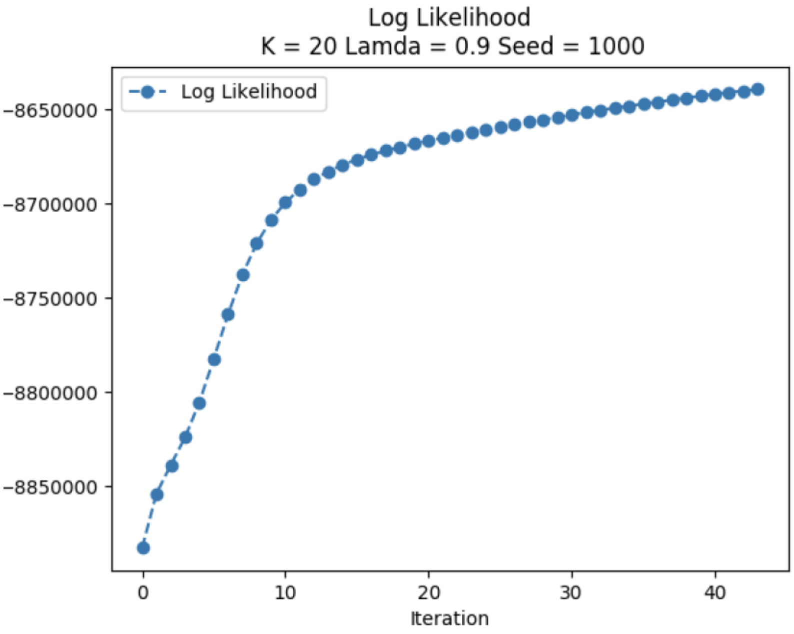
(c)

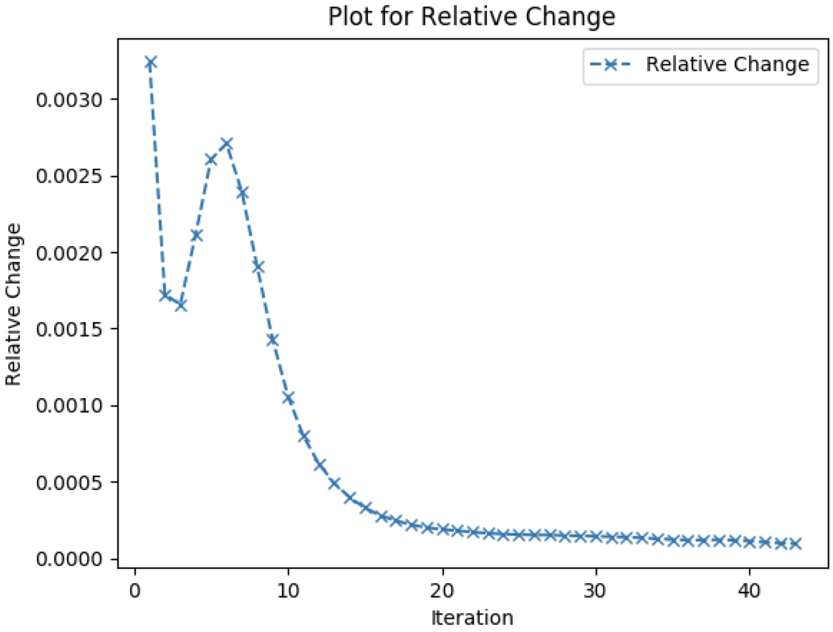
The value of the relative change in log likelihood must decrease at each iteration and as the no. of iterations increase, the decrease in the value must be slower. We expect that the value of the relative change in log likelihood would decrease exponentially with each update. Below is an estimated graph of relative change in log likelihood with change in iteration:



(d)

The plots for the log likelihood values and the relative change in log likelihood values for K = 20 and $\lambda = 0.9$ are given below:





The log likelihood function values in each iteration increase significantly initially with the increase in iteration and then as the no. of iterations grow large, the values stabilize.

The relative change in log likelihood values decrease at each iteration and then stabilizes as the no. of iterations grow. This is because the estimate of π *and* θ improve at each iteration. The plots match our predictions. The relative change increased in the initial iterations unlike the estimated plot. But the plot was similar for lamda = 0.3. After a few changes, the model starts to fit the data well and hence the relative change decreases. Also, the curve we are trying to maximize might be flatter, due to which, the change is smaller than the change with a steeper shaped curve.

(e)

Top 10 words in each of the 20 topics found for $\lambda = 0.9$ are as follows:

Topic 1	['security', 'skyline', 'string', 'key', 'protocol', 'group', 'binding', 'secure', 'flow', 'false']
Topic 2	['languages', 'students', 'uml', 'extraction', 'language', 'corpora', 'corpus', 'programming', 'provenance', 'scale']
Topic 3	['cache', 'caches', 'broadcast', 'misses', 'schemas', 'attack', 'file', 'bandwidth', 'stm', 'max']
Topic 4	['energy', 'register', 'voltage', 'fuzzy', 'consumption', 'approximation', 'log', 'compression', 'scaling', 'graphs']
Topic 5	['sensor', 'networks', 'sensors', 'nodes', 'wireless', 'localization', 'aggregation', 'network', 'probability', 'wsns']
Topic 6	['routing', 'type', 'path', 'awareness', 'program', 'route', 'programs', 'transformation', 'annotations', 'transformations']
Topic 7	['clustering', 'phase', 'clusters', 'proxy', 'rfid', 'measure', 'program', 'statements', 'cluster', 'topic']
Topic 8	['branch', 'product', 'prediction', 'itemsets', 'mining', 'frequent', 'visualization', 'temporal', 'knowledge', 'software']
Topic 9	['patterns', 'mining', 'video', 'pattern', 'sequential', 'sequence', 'sequences', 'array', 'frequent', 'items']
Topic 10	['changes', 'network', 'search', 'internet', 'fault', 'traffic', 'failure', 'message', 'links', 'priority']
Topic 11	['test', 'testing', 'programs', 'garbage', 'collection', 'code', 'symbolic', 'heap', 'software', 'java']
Topic 12	['logic', 'placement', 'probabilistic', 'nearest', 'timing', 'time', 'specifications', 'series', 'complexity', 'neighbor']
Topic 13	['motion', 'security', 'models', 'debugging', 'ontologies', 'recommender', 'traces', 'recommendation', 'trace', 'commerce']
Topic 14	['query', 'xml', 'web', 'queries', 'search', 'documents', 'document', 'retrieval', 'database', 'services']
Topic 15	['network', 'sensor', 'networks', 'privacy', 'security', 'wireless', 'control', 'nodes', 'mobile', 'access']

Topic 16	['memory', 'performance', 'processor', 'chip', 'hardware', 'processors', 'parallel', 'cache', 'instruction', 'shared']
Topic 17	['image', 'classification', 'images', 'feature', 'features', 'classifier', 'training', 'learning', 'methods', 'text']
Topic 18	['surface', 'shape', 'rendering', 'texture', 'image', 'point', 'geometry', 'scene', 'illumination', 'mesh']
Topic 19	['grid', 'resource', 'distributed', 'workflow', 'management', 'dependencies', 'resources', 'service', 'business', 'consistency']
Topic 20	['learning', 'disk', 'supervised', 'multicast', 'player', 'transactional', 'acm', 'algorithms', 'join', 'algorithm']

Top 10 words in each of the 20 topics found for $\lambda = 0.3$ are as follows:

Topic 1	['the', 'of', 'to', 'and', 'we', 'security', 'in', 'that', 'are', 'on']
Topic 2	['the', 'of', 'data', 'to', 'and', 'in', 'is', 'for', 'learning', 'model']
Topic 3	['the', 'of', 'in', 'and', 'to', 'network', 'nodes', 'networks', 'sensor', 'is']
Topic 4	['the', 'to', 'and', 'of', 'network', 'service', 'in', 'traffic', 'is', 'that']
Topic 5	['the', 'of', 'we', 'in', 'is', 'for', 'problem', 'algorithm', 'that', 'and']
Topic 6	['the', 'of', 'and', 'to', 'in', 'we', 'for', 'language', 'is', 'type']
Topic 7	['the', 'of', 'to', 'is', 'and', 'in', 'based', 'on', 'are', 'clustering']
Topic 8	['the', 'and', 'of', 'in', 'to', 'software', 'this', 'for', 'is', 'systems']
Topic 9	['the', 'of', 'in', 'to', 'and', 'patterns', 'mining', 'data', 'that', 'for']
Topic 10	['the', 'of', 'to', 'that', 'and', 'in', 'we', 'is', 'can', 'time']
Topic 11	['the', 'of', 'to', 'and', 'that', 'code', 'program', 'in', 'for', 'is']
Topic 12	['the', 'of', 'and', 'in', 'we', 'to', 'is', 'that', 'prediction', 'for']
Topic 13	['the', 'and', 'of', 'to', 'system', 'performance', 'applications', 'for', 'application', 'on']
Topic 14	['the', 'of', 'query', 'to', 'web', 'and', 'in', 'we', 'queries', 'for']
Topic 15	['the', 'of', 'to', 'and', 'we', 'temporal', 'in', 'that', 'control', 'access']
Topic 16	['the', 'to', 'and', 'of', 'memory', 'in', 'performance', 'cache', 'on', 'power']
Topic 17	['the', 'and', 'of', 'to', 'in', 'user', 'users', 'that', 'with', 'we']
Topic 18	['the', 'of', 'and', 'to', 'we', 'for', 'is', 'in', 'by', 'with']
Topic 19	['the', 'of', 'to', 'in', 'and', 'for', 'on', 'that', 'time', 'is']
Topic 20	['the', 'data', 'of', 'and', 'in', 'to', 'is', 'time', 'for', 'that']

We notice that the top words for each of the 20 topics found by $\lambda = 0.9$ are relevant content words that clearly can define the gist of the topic whereas the top words found be the PLSA algorithm for $\lambda = 0.3$ are mostly the stop irrelevant words with high frequency in the topic.

This matches our expectations since when λ is large, then the background language model is being considered appropriately which balances the tradeoff between the mixture model and the background language model, by not assigning all the weight to the highly frequent stop words. Whereas, for small λ, probability of selecting a word from the background language model is low, hence highly frequent non-relevant words are given more importance and the top words for the topics clearly reflect that.

The trend is observed due to dependence of value of λ on the importance of the background language model. The model attempts to assign high probabilities to high frequent words in the data to collaboratively maximize the likelihood. Component models tend to bet high probabilities on different words to avoid the "competition".

To find topics that have a descriptive top ten word, λ value must be set very high, say $\lambda = 0.95$ would work well.

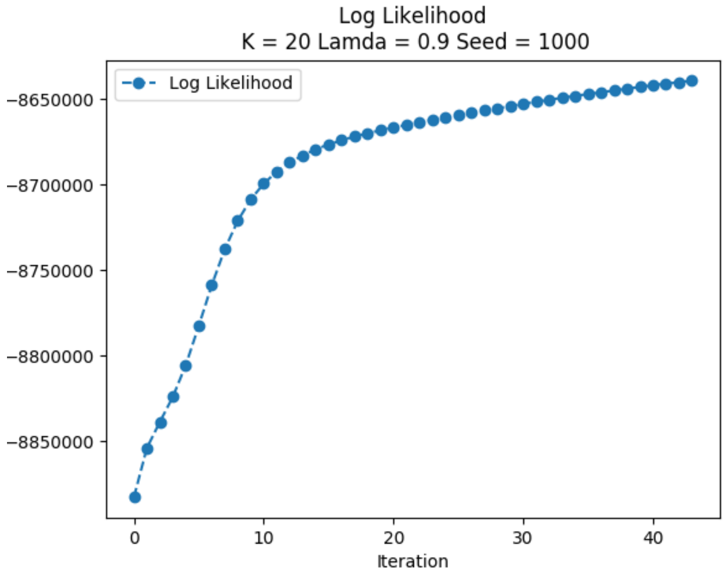
(f)

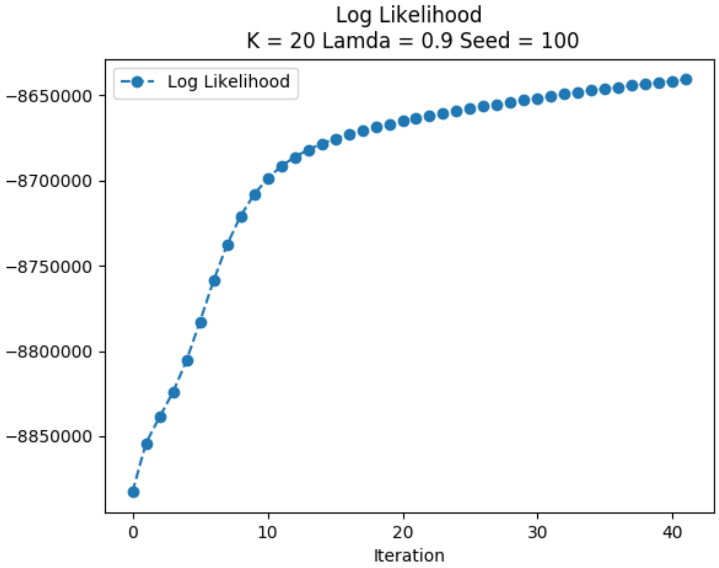
Few of the suggestions are as follows:
1. Maximum Likelihood Estimation is a choice if we don't have any prior knowledge about the topic models. But given a topic, we might have some prior knowledge which may be used to get better starting point for the model.
This prior knowledge can be used as an initial starting point to guide the model which may lead topic models to be near to the predefined facets.
Rather than directly fitting the data with PLSA model, using some domain knowledge to define a prior on the topic models and then estimating the topic models using the Maximum A Posterior (MAP) estimator would seem like a reasonable approach to improve the model. We may want to use the prior topic model given by the user for this purpose. An extension of PLSA model, namely LDA uses prior on the π values, which denotes the coverage of each topic in the documents.

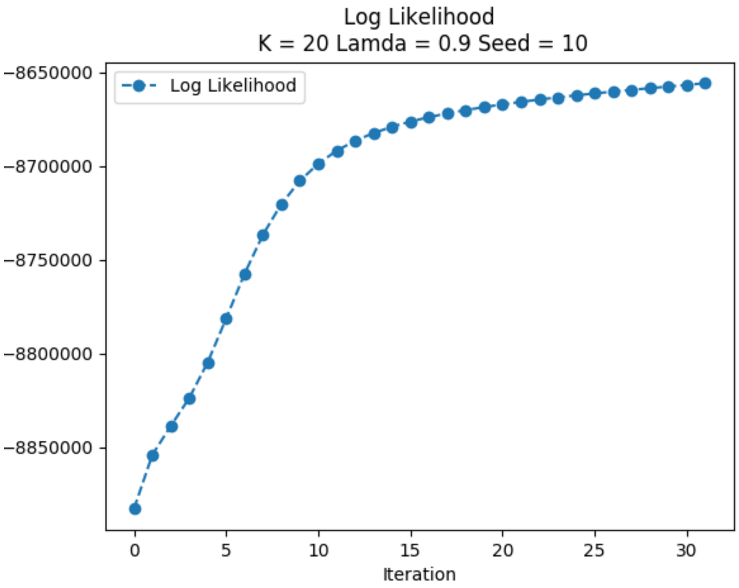
2. Another improvement can be using the TF – IDF weightings in the log likelihood function to penalize the highly frequent stop words in the documents.

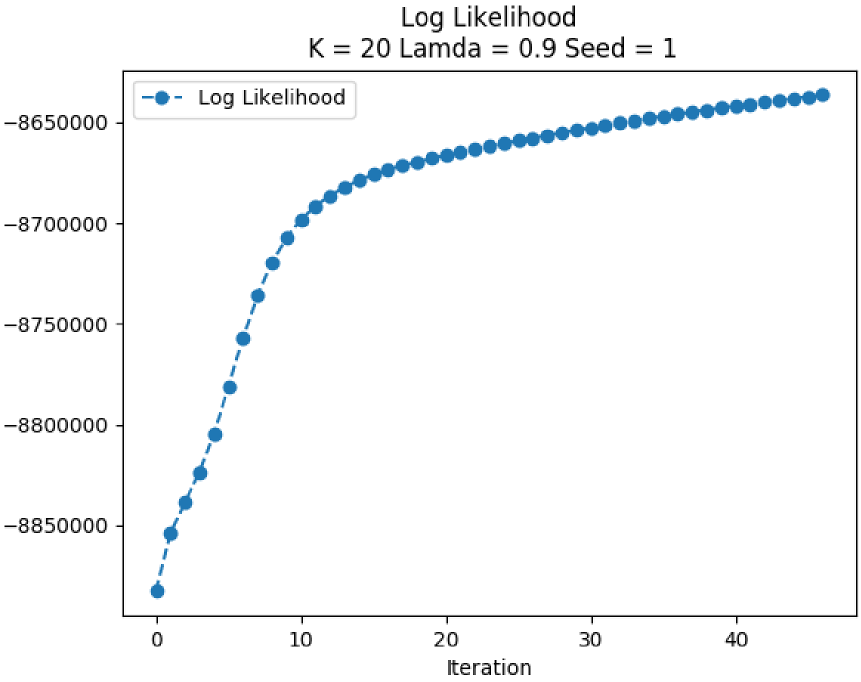
(g)

The loglikelihood versus iteration plots for K=20, $\lambda = 0.9$ with 4 different values of seeds, namely, seed = 1000, 100, 10 and 1 are as follows:


Log Likelihood
K = 20 Lamda = 0.9 Seed = 1000
Log Likelihood
−8650000
−8700000
−8750000
−8800000
−8850000
0
10
20
30
40
Iteration


Log Likelihood
K = 20 Lamda = 0.9 Seed = 100
Log Likelihood
−8650000
−8700000
−8750000
−8800000
−8850000
0
10
20
30
40
Iteration





No, all four runs do not reach the same log likelihood value. For seed = 10, log likelihood was smaller than the values for seed = 100 and seed = 1000.

This is possible in general as well because EM guarantees convergence to local maxima, but not necessarily to global maxima. PLSA / EM is a hill climbing algorithm which can get trapped in local maxima and may not reach the global maxima. Hence the end log likelihood values can be different depending on the maxima reached using the initial random seed.

If someone wanted to achieve the highest possible log likelihood for your model on their data, what would you suggest?

Multiple trials with different random seeds (different starting points) and choosing the model with the highest final likelihood. This will help reaching a global maximum and thus maximize the log likelihood.

Submitted by:

Mihika Dave (mhdave2)

Rachneet Kaur (rk4)