TOPIC MODELING IN R

Finding the best number of topics

Pavel Oleinikov
Associate Director
Quantitative Analysis Center
Wesleyan University

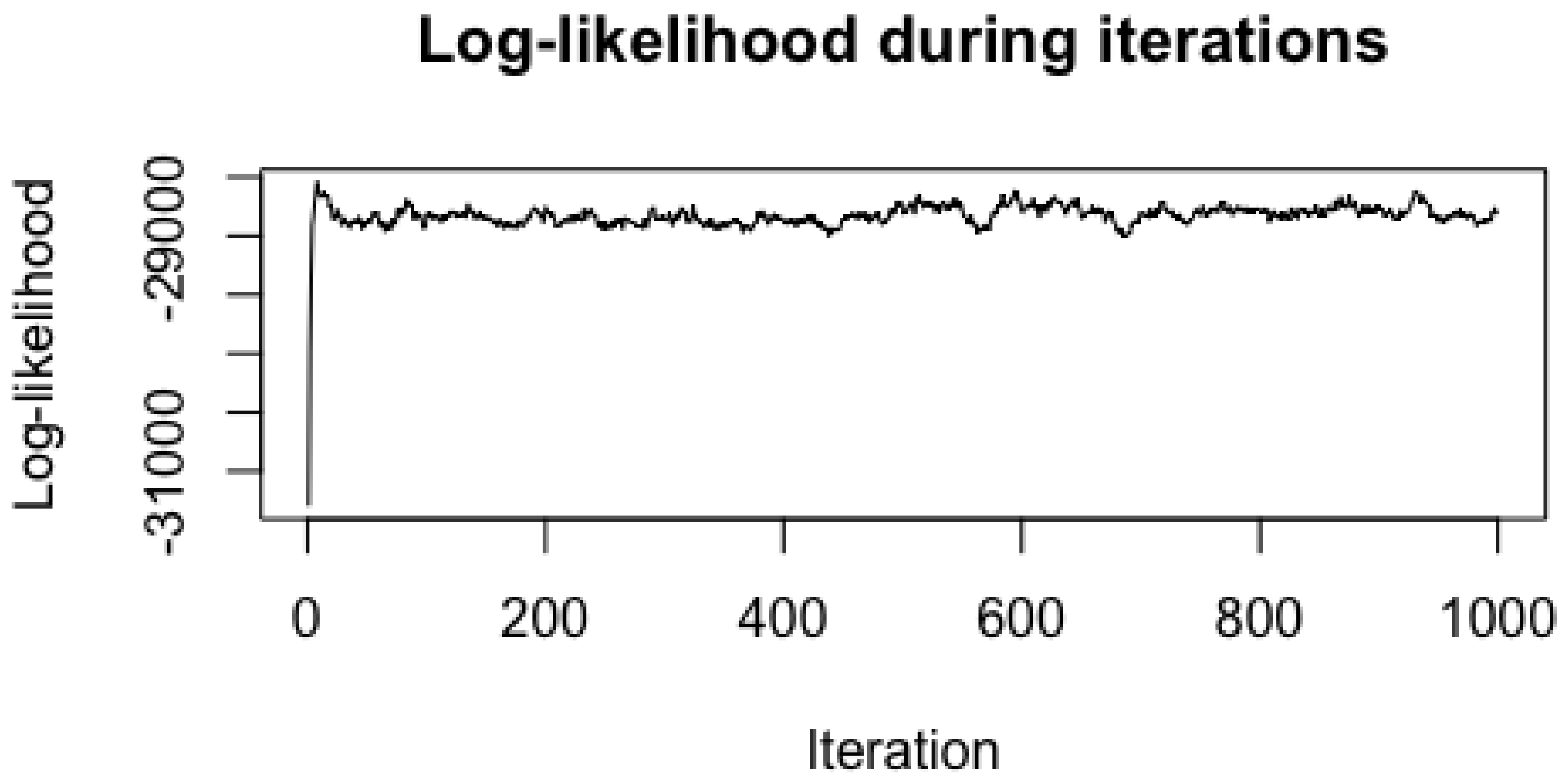
Approaches

- Topic coherence - examine the words in topics, decide if they make sense
  - E.g. site, settlement, excavation, *popsicle* - low coherence.
- Quantitative measures
  - Log-likelihood - how plausible model parameters are given the data
  - Perplexity - model's "surprise" at the data

Log-likelihood

- Likelihood - measure of how plausible model parameters are given the data
- Taking a logarithm makes calculations easier
- All values are negative: when `x<1`, `log(x) < 0`
- Numerical optimization - search for the largest log-likelihood
  - E.g. -100 is better than -105
- Function `logLik` returns log-likelihood of an LDA model

Log-likelihood



Perplexity

- Perplexity is a measure of model's "surprise" at the data
- Positive number
- Smaller values are better
- Function `perplexity()` returns "surprise" of a model (`object`) when presented `newdata`

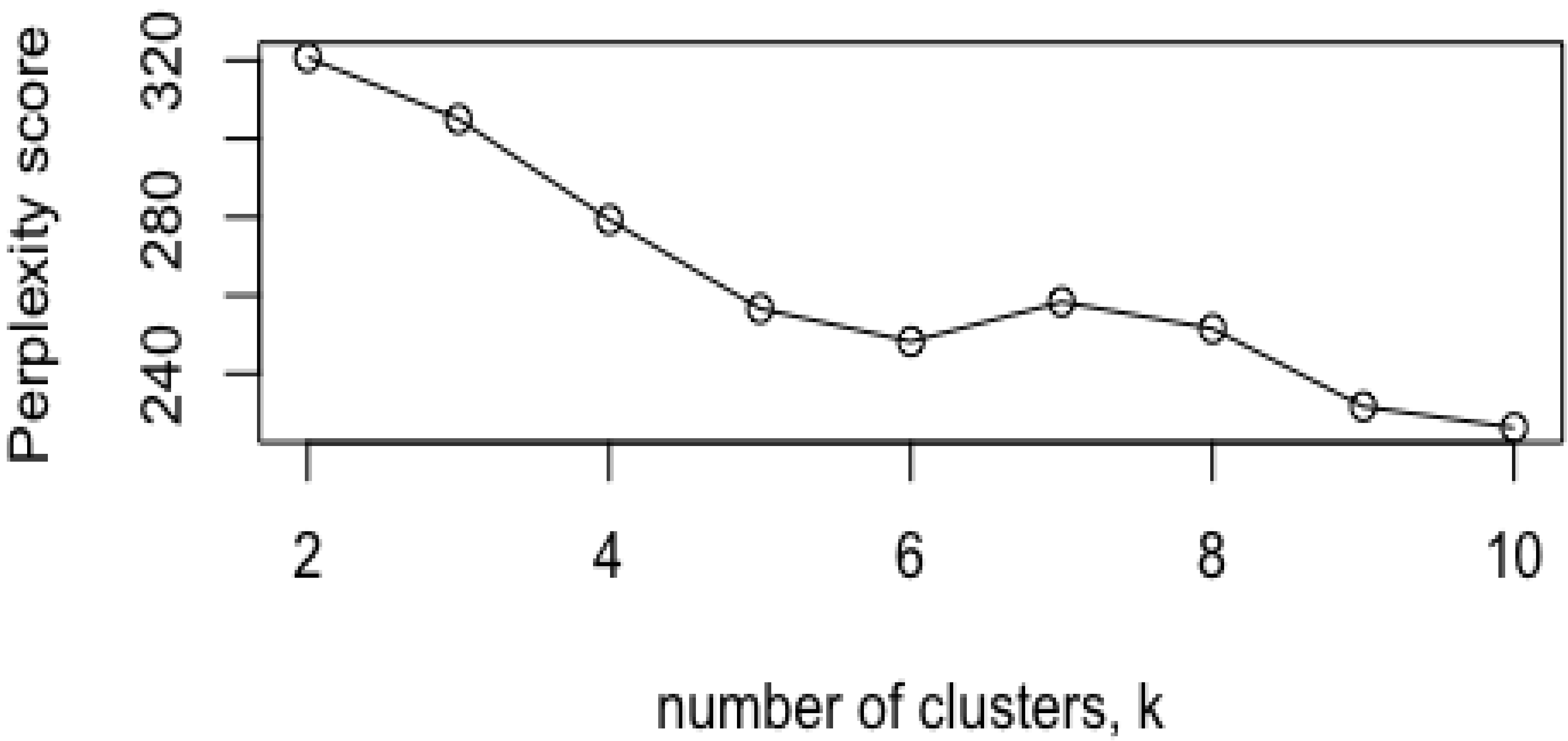
```
perplexity(object=mod, newdata=dtm)
```

```
186.7139
```

Finding the best k

- Fit the model for several values of `k`
- Plot the values
- Pick the one where improvements are small
- Similar to "elbow plot" in k-means clustering

```
mod_log_lik = numeric(10)
mod_perplexity = numeric(10)
for (i in 2:10) {
  mod = LDA(dtm, k=i, method="Gibbs",
            control=list(alpha=0.5, iter=1000, seed=12345, thin=1))
  mod_log_lik[i] = logLik(mod)
  mod_perplexity[i] = perplexity(mod, dtm)
}
```


Perplexity score
240
280
320
2
4
6
8
10
number of clusters, k

Time costs

- Searching for best `k` can take *a lot of time*
- Factors: number of documents, number of terms, and number of iterations
- Model fitting can be resumed
- Function `LDA` accepts an LDA model as an object for initialization

```
# Initial run
mod = LDA(x=dtm, method="Gibbs", k=4,
          control=list(alpha=0.5, seed=12345, iter=1000, keep=1))

# Resumed run
mod2 = LDA(x=dtm, model=mod,
    control=list(thin=1, seed=10000, iter=200))
```

Practice dataset

- A corpus of 90 documents
- Abstracts of projects approved by the US National Science Foundation (NSF)
- Sample from search for four keywords: mathematics, physics, chemistry, and marine biology

```
The study of disease using mathematical models has a long
and rich history.
Much interesting and new mathematics has been motivated by disease,
because the problems are inherently nonlinear and multidimensional.
```

TOPIC MODELING IN R

Let's practice

TOPIC MODELING IN R

Topic model fitted on one document

Pavel Oleinikov
Associate Director
Quantitative Analysis Center
Wesleyan University

Analyzing one (long) novel

- A topic model is used to analyze one long document, e.g. *Moby Dick*
  - JSTOR Labs Text Analyzer, https://www.jstor.org/analyze/analyzer/progress
- Documents are chunks long enough to capture an event or a scene in the plot
- For traditional novels - 1000+ words

Text chunks as chapters

- We had a variable for chapter number

```
corpus %>%
    unnest_tokens(input=text, output=word) %>%
    count(chapter, word)
```

- With text chunks, we need to generate the "chapter number" on our own
- Candidate function: `%/%` - integer division

```
7 %/% 3
25784 %/% 1000

2
25
```

Generating the document number

- Unnest tokens,
- assign sequential number to each word,
- compute document number

```
corpus %>%
   unnest_tokens(input=text, output=word) %>%
   mutate(word_index = 1:n()) %>%
   mutate(doc_number = word_index %/% 1000 + 1) %>%
   count(doc_number, word) %>%
   cast_dtm(term=word, document=doc_number, value=n)
```

Craft vs. science

- Chunk size is a matter of craft
- May vary with writing style
- Solutions:
  - Try different chunk sizes
  - Make sure the text chunk does not span chapter boundary

TOPIC MODELING IN R

Let's practice

TOPIC MODELING IN R

Using seed words for initialization

Pavel Oleinikov
Associate Director
Quantitative Analysis Center
Wesleyan University

Seed for random numbers

- Pseudo-randomness

```
control=list(seed=12345)
```

- Used to ensure reproducibility of results between runs
- LDA performs randomized search through the space of parameters
    - Gibbs sampling
- Topic numbering is unstable

Seed words

- Gibbs method supports initialization with seed words
  - "Lock" topic numbers
  - Specify weights for seed words for topics
- `seedwords` requires a matrix, **k** rows, **N** columns.
  - **k** is number of topics, **N** is vocabulary size
  - Weights get normalized internally so they sum up to 1.

Example

- Tiny dataset: five sentences about restaurants and loans
  - `k` is 2
  - dtm size - 5 rows, 34 columns
- Declare a matrix with 2 rows and 34 columns
- Assign 1 to "restaurant" in row 1, "loans" in row 2

```
seedwords = matrix(nrow=2, ncol=34, data=0)
colnames(seedwords) = colnames(dtm)
seedwords[1, "restaurant"] = 1
seedwords[2, "loans"] = 1
```

Example, continued

Topic model fitted without seedwords

```
lda_mod = LDA(x=dtm, k=2,
    method="Gibbs",
    control=list(alpha=1,
    seed=1234))

tidy(lda_mod, "beta") %>%
  spread(key=topic, value=beta) %>%
  filter(term %in% c("restaurant",
   "loans"))
```

Loans is topic 1, restaurants - topic 2

```
  term           `1`     `2`
1 loans      0.0767 0.00379
2 restaurant 0.0272 0.0795
```

Topic model fitted with seedwords

```
lda_mod = LDA(x=dtm, k=2,
    method="Gibbs",
    seedwords=seedwords,
    control=list(alpha=1,
    seed=1234))

tidy(lda_mod, "beta") %>%
  spread(key=topic, value=beta) %>%
  filter(term %in% c("restaurant",
   "loans"))
```

Loans is topic 2, restaurants - topic 1

```
  term            `1`     `2`
1 loans      0.00379 0.0967
2 restaurant 0.155   0.00236
```

Uses

- Convenient for pre-trained models
  - Training a model involves multiple runs of the algorithm, even for the same `k`
  - Seedwords let us "lock" topic numbers
- Helpful input for training models
  - Speed up algorithm convergence by providing a starting point

TOPIC MODELING IN R

Let's practice

TOPIC MODELING IN R

Final words (and more things to learn)

Pavel Oleinikov
Associate Director
Quantitative Analysis Center
Wesleyan University

Not just words

- LDA topic modeling is a clustering algorithm
  - Soft clustering - probability instead of hard assignment
- Uses counts data
  - Customers attending events
  - Coordinates rounded down, e.g. Fujino et al (2017), Extracting Route Patterns of Vessels from AIS Data by Using Topic Model

Structured topic models - STM

- Variational Expectation-Maximization (VEM) for model estimation
  - Can be applied to correlated topic models
  - Topic proportions follow a multivariate normal distribution
- Package `stm` by Margaret Roberts, Brandon Stewart, Dustin Lingley, and Kenneth Benoit
  - regression modeling of topic proportions and covariates
  - automatic corpus alignment
  - held-out data as omitted words in documents
  - can use result of LDA model as a seed

Deep learning and word embeddings

- `Word2Vec` models:
  - Use deep learning neural network to predict words that occur adjacent to a word, $\pm n$ with $n = 2$, or 4
  - Transform into a vector of smaller dimensions (25, 50, 100)
- **Word windows used in chapter 3 for named entity recognition?**
- `word2vec` models use very large corpora (e.g., 2 billion words)
  - do not make accommodations for multi-word entities
  - take a long time to train
- Experiment with package `wordVectors` created by Ben Schmidt

TOPIC MODELING IN R

Go out and play!