Text Mining and Sentiment Analysis

Prof. Annamaria Bianchi
A.Y. 2024/2025

Lecture 17

6 May 2025

UNIVERSITÀ DEGLI STUDI DI BERGAMO
Dipartimento di Scienze Economiche

Outline

Topic Modeling: document-topic probabilities, by-word assignment

Case study

Packages: **topicmodels**, **tidytext**

Functions: `tidytext::tidy(), tidytext::augment()`

Topic Modeling

We continue the working example on the AssociatedPress dataset (provided by the topicmodels package).

```
> library(tidyverse)
> library(tidytext)
> library(topicmodels)
> data("AssociatedPress")
> ap_lda = LDA(AssociatedPress, k=2, control = list(seed = 1234))
```

Document-Topic Probabilities

Besides estimating each topic as a mixture of words, LDA also models each document as a mixture of topics
Let us extract the per-document-per-topic probabilities, called γ from the model [these correspond to what we called θ_d previously].

```
> ap_documents = tidy(ap_lda, matrix = "gamma")
> ap_documents
# A tibble: 4,492 x 3
   document topic    gamma
      <int> <int>    <dbl>
 1        1     1 0.248
 2        2     1 0.362
 3        3     1 0.527
 4        4     1 0.357
 5        5     1 0.181
 6        6     1 0.000588
 7        7     1 0.773
 8        8     1 0.00445
 9        9     1 0.967
10       10     1 0.147
# ... with 4,482 more rows
```

We obtain a tibble with format: one-document-per-topic-per-row.

For each combination, the model computes the topic proportions. These are estimated as the proportion of words from that document that are generated from that topic.

For example, 24.8% of the words in Document 1 are generated from Topic 1.

Document-Topic Probabilities

Exercise. 1) Verify and convince yourself that γ probabilities represent probability distributions for each document.
2) Can you identify a document that is almost entirely drawn from Topic 1 and a document that is almost entirely drawn from Topic 2?

Document-Topic Probabilities

```
> ap_documents = tidy(ap_lda, matrix = "gamma")
> ap_documents
# A tibble: 4,492 x 3
   document topic    gamma
      <int> <int>    <dbl>
 1        1     1 0.248
 2        2     1 0.362
 3        3     1 0.527
 4        4     1 0.357
 5        5     1 0.181
 6        6     1 0.000588
 7        7     1 0.773
 8        8     1 0.00445
 9        9     1 0.967
10       10     1 0.147
# ... with 4,482 more rows
```

Many of these documents are obtained from a mix of the two topics.

Document 6 is drawn almost entirely from topic 2, while Document 9 is drawn almost entirely from topic 1.

Document-Topic Probabilities

Let us check whether it makes sense that document 6 is drawn almost entirely from topic 2, by looking at the most common words in that document

```
> tidy(AssociatedPress) %>%
+   filter(document == 6) %>%
+   arrange(desc(count))
# A tibble: 287 x 3
   document term           count
      <int> <chr>          <dbl>
 1        6 noriega           16
 2        6 panama            12
 3        6 jackson            6
 4        6 powell             6
 5        6 administration     5
 6        6 economic           5
 7        6 general            5
 8        6 i                  5
 9        6 panamanian         5
10        6 american           4
# ... with 277 more rows
```

Based on the most common words, this article appears to be about the relationship between the American government and Panamanian dictator Manuel Noriega.

Document-Topic Probabilities

Let us now look at the most common words in document 9

```
> tidy(AssociatedPress) %>%
+   filter(document == 9) %>%
+   arrange(desc(count))
# A tibble: 7 x 3
  document term         count
     <int> <chr>        <dbl>
1        9 brush            1
2        9 developments     1
3        9 fires            1
4        9 forest           1
5        9 states           1
6        9 summary          1
7        9 western          1
```

Do you think is makes sense that this article is classified as business or financial news?

By-word assignments

One step of the LDA algorithm consists in assigning each word in each document to a topic. The more words in a document are assigned to that topic, generally, the more weight (probability) will go to that document-topic classification.
You might want to see which words in each document are assigned to which topic. This can be done using the **tidytext**::`augment()` function.

```
augment(x, data, ...)
```

`x`	An LDA (or LDA_VEM) object from the topicmodels package
`data`	The data given to the LDA function, either as a DocumentTermMatrix or as a tidied table with "document" and "term" columns

By-word assignments

Augment returns a tidy data frame with one row per original document-term pair. It adds the extra column .topic, with the topic each term was assigned to within each document.

```
> assignments = augment(ap_lda, AssociatedPress)
> assignments
# A tibble: 302,031 x 4
   document term        count .topic
      <int> <chr>       <dbl>  <dbl>
 1        1 adding          1      2
 2        1 adult           2      1
 3        1 ago             1      2
 4        1 alcohol         1      2
 5        1 allegedly       1      2
 6        1 allen           1      1
 7        1 apparently      2      2
 8        1 appeared        1      2
 9        1 arrested        1      2
10        1 assault         1      2
# ... with 302,021 more rows
```

The extra column added by `augment()` starts with **.** to prevent overwriting existing columns.

By-word assignments

Taking a look at assignments in document 6

```
> assignments %>%
+   filter(document == 6) %>%
+   arrange(desc(count)) %>%
+   View()
```

	document	term	count	.topic
1	6	noriega	16	2
2	6	panama	12	2
3	6	jackson	6	2
4	6	powell	6	2
5	6	administration	5	2
6	6	economic	5	2
7	6	general	5	2

Case study

We use articles from the Guardian newspaper (contained in the file text.rds). The corpus contains all Guardian articles mentioning Pakistan between November 14 2015 and December 1 2015.

Q: How does the Guardian newspaper prioritize articles about Pakistan?

```
> text <- read_rds("text.rds")
> View(text)
```

Topic Modeling

	id	sectionName	body
1	sport/live/2015/nov/27/pakistan-v-england-second-t20-int...	Sport	<div id="block-5658afbae4b03bf40
2	sport/2015/dec/01/england-women-pakistan-odi-cricket	Sport	<p>Englands women are to play Pak
3	technology/2015/nov/30/blackberry-pakistan-government-...	Technology	<p>Smartphone and secure commu
4	sport/live/2015/nov/30/pakistan-v-england-third-t20-intern...	Sport	<div id="block-565ca896e4b0cf03a
5	world/2015/nov/25/last-minute-reprieve-abdul-basit-disabl...	World news	<p>Plans <a href="http://www.theg
6	sport/live/2015/nov/26/pakistan-v-england-first-t20-interna...	Sport	<div id="block-56575a6be4b0e0f1b
7	sport/2015/nov/25/england-pakistan-twenty20-dubai-eoin-...	Sport	<p> </p> <p>

Topic Modeling

We shall remove any text within <>

```
> text=text %>%
+   mutate(body = str_replace_all(body, pattern = "\\<.*?\\>", replacement = ""))
```

	id	sectionName	body
1	sport/live/2015/nov/27/pakistan-v-england-second-t20-int...	Sport	7.34pm GMT England win the ser
2	sport/2015/dec/01/england-women-pakistan-odi-cricket	Sport	Englands women are to play Pakistar
3	technology/2015/nov/30/blackberry-pakistan-government-...	Technology	Smartphone and secure communicat
4	sport/live/2015/nov/30/pakistan-v-england-third-t20-intern...	Sport	7.53pm GMT So England win th
5	world/2015/nov/25/last-minute-reprieve-abdul-basit-disabl...	World news	Plans to execute a a disabled man at

Topic Modeling

Let us also modify the ID variable

```
> text = text %>%
+   mutate(IDn = seq_along(body)) %>%
+   select(IDn, sectionName, body)
> text
# A tibble: 50 x 3
     IDn sectionName body
   <int> <chr>       <chr>
 1     1 Sport       "  7.34pm GMT    England win the series    That puts Englan~
 2     2 Sport       "Englands women are to play Pakistan in one-day internation~
```

Topic Modeling

We move to the tidy text format and preprocess the data by dropping stop words

```
> ga.td = text %>%
+   unnest_tokens(word, body) %>%
+   anti_join(stop_words)
Joining with `by = join_by(word)`
> ga.td
# A tibble: 53,744 × 3
     IDn sectionName word
   <int> <chr>       <chr>
 1     1 Sport       7.34pm
 2     1 Sport       gmt
 3     1 Sport       england
 4     1 Sport       win
 5     1 Sport       series
 6     1 Sport       england
```

Topic Modeling

Exercise. Trasform the data in the format required for applying an LDA model.

Topic Modeling

Apply LDA with 4 topics

```
> ga_lda4 = LDA(ga.dtm, k=4, control = list(seed = 1123))
> ga_topics4 = tidy(ga_lda4, matrix = "beta")
> ga_topics4
# A tibble: 46,464 x 3
  topic term      beta
  <int> <chr>    <dbl>
1     1 0     5.33e-39
2     2 0     1.05e- 9
3     3 0     6.54e-13
4     4 0     5.65e- 3
```

Topic Modeling

Let us look at word-topic probabilities

```
> ga_top4 = ga_topics4 %>%
+   group_by(topic) %>%
+   slice_max(beta, n=10) %>%
+   ungroup() %>%
+   arrange(topic, -beta)
> View(ga_top4)
```

Also by means of a graph

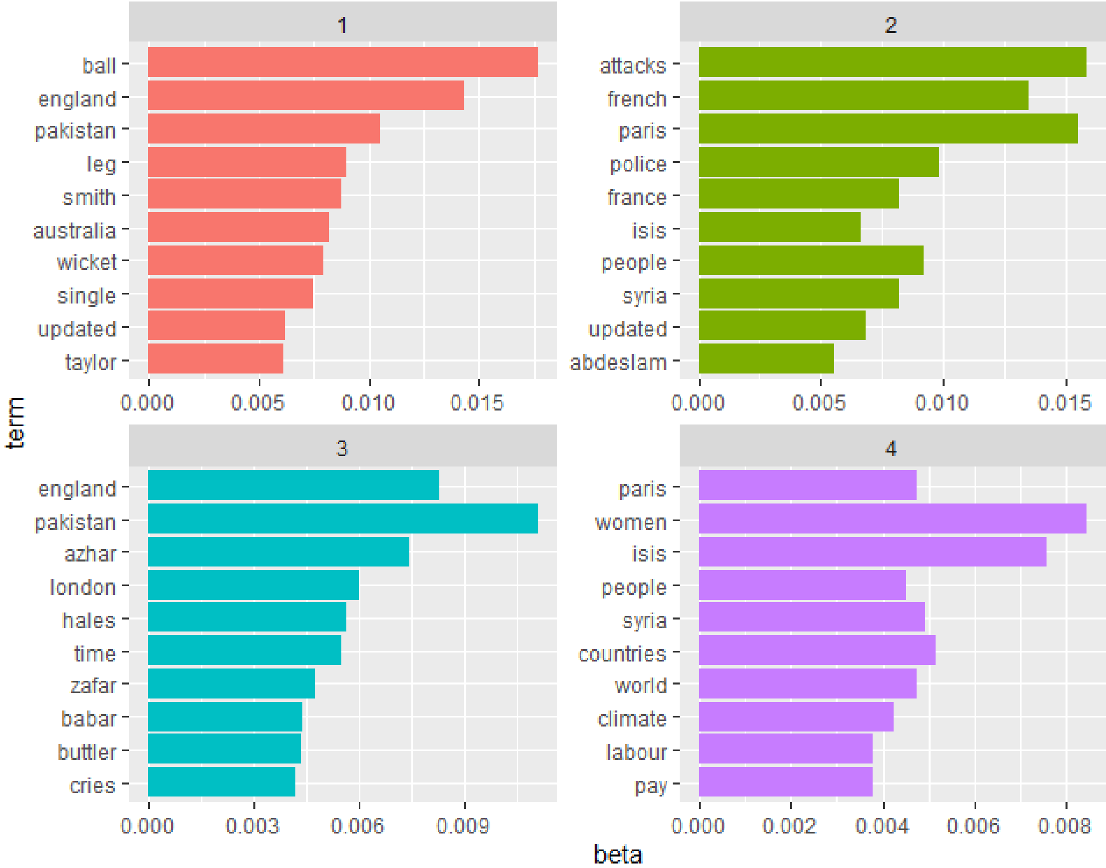
```
> ga_top4 %>%
+   mutate(term = reorder(term, beta)) %>%
+   ggplot(aes(term, beta, fill= factor(topic))) +
+   geom_col(show.legend = F)+
+   facet_wrap(~ topic, scales = "free")+
+   coord_flip()
```

Topic Modeling

Topic Modeling

Exercise. Rerun the analysis by first filtering out «gmt» and numbers and terms starting with numbers.

Topic Modeling



Topic Modeling

Let us look at document-topic probabilities

```
> ga_doc4 = tidy(ga_lda4, matrix = "gamma")
> ga_doc4
> summary_doc4 = ga_doc4 %>%
 + mutate(document = as.numeric(document)) %>%
 + arrange(document) %>%
 + pivot_wider(names_from = topic, values_from = gamma) %>%
 + cbind(text$sectionName)
> view(summary_doc4)
```

	document	1	2	3	4	text$sectionName
1	1	9.999661e-01	1.130985e-05	1.130985e-05	1.130985e-05	Sport
2	2	8.420768e-01	1.371716e-04	1.371716e-04	1.576488e-01	Sport
3	3	2.789969e-02	1.478088e-01	1.550524e-04	8.241365e-01	Technology
4	4	9.999698e-01	1.007276e-05	1.007276e-05	1.007276e-05	Sport
5	5	1.639576e-04	3.375886e-02	9.659132e-01	1.639576e-04	World news

Topic Modeling

```
> ga.td %>%
+   filter(IDn == 5) %>%
+   arrange(desc(n)) %>%
+   View()
```

	IDn	word	n
1	5	government	6
2	5	pakistan	6
3	5	execution	5
4	5	rights	5
5	5	human	4
6	5	basit	3
7	5	basits	3

Topic Modeling

```
> assignments4 = augment(ga_lda4, ga.dtm)
> assignments4 %>% filter(document == 5) %>%
+   arrange(desc(count)) %>%
+   View()
```

	document	term	count	.topic
1	5	pakistan	6	3
2	5	government	6	3
3	5	execution	5	3
4	5	rights	5	3
5	5	human	4	3
6	5	basit	3	3
7	5	basits	3	3

How many topics?

Topic models such as LDA allow you to specify the number of topics in the model. On the one hand, this is a nice thing, because it allows you to adjust the granularity of the topics: between a few broad topics and many more specific topics. On the other hand, it begets the question what *the best number* of topics is.

The short and perhaps disappointing answer is that *the best number* of topics does not exist.

Two general approaches. The first approach is to look at how well our model fits the data. Second approach based on human interpretation.

Exercise for you

Exercise 1
With reference to the Associated Press case study, try to investigate whether considering 2 topics in the analysis was a correct choice or more topics are needed.

Exercise 2
With reference to the Guardian articles case study, answer the following questions:

1) Explain the regular expression used on slide 5, namely "\\<.*?\\>"
2) What is the number of topics you would suggest?
3) Compute the length of each articles and comment on the amount of effort the Guardian devoted to topics related to Pakistan. [Hint: we shall approximate the length of each article by counting the number of words in the tidy dataframe].