Using topic models as classifiers

TOPIC MODELING IN R

Pavel Oleinikov

Associate Director, Quantitative Analysis Center, Wesleyan University

datacamp

Topic models as soft classifiers

- Classification task - find probability of belonging to class
  - Named entity recognition - is an entity a geographic name or a person?
  - "***Washington*** crossed the Delaware river" vs. "They did a road trip across ***Washington***"
- Topic modeling of context - a set of words occurring next to the entity

Effect of control parameter alpha

- k = 2, alpha = 1

```
  document   `1`    `2`
  <chr>     <dbl>  <dbl>
1 d_1       0.154 0.846
2 d_2       0.278 0.722
3 d_3       0.875 0.125
4 d_4       0.923 0.0769
5 d_5       0.5   0.5
```

- k = 2, alpha = $\frac{50}{k}$ = 25

```
  document   `1`   `2`
  <chr>     <dbl> <dbl>
1 d_1       0.475 0.525
2 d_2       0.530 0.470
3 d_3       0.482 0.518
4 d_4       0.508 0.492
5 d_5       0.5   0.5
```

How LDA fits a model

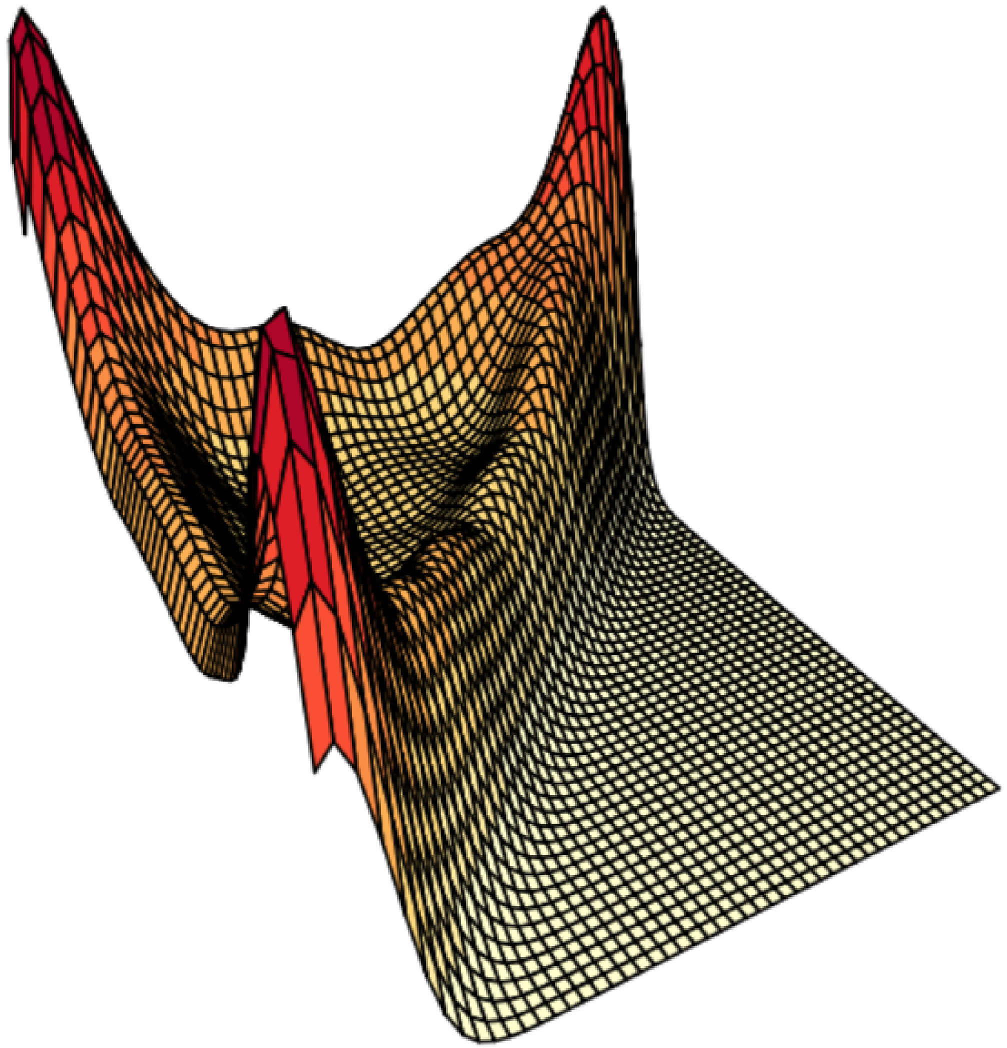
- A bag of M&M candy - multinomial distribution
  - Several outcomes (colors of candy) repeated n times
  - Each outcome has its own probability - fixed by the factory that filled the bag
  - Probabilities sum up to 1
  - What's the probability of getting 5 yellow, 2 brown, 2 blue, and 1 black when we take out 10 pieces of candy?
- In LDA model, topics are color, and there are two "bags of candy": one for documents and one for words

The Dirichlet in LDA

- Randomized search for the best values of probabilities
  - E.g., try 0.2, 0.5 and 0.3 for proportions of three topics
- Hard to do for large number of topics and words
- Instead, use values from Dirichlet distribution as guesses
- Dirichlet distribution returns a set of numbers that add up to 1 - serve as probabilities of colors for M&M candy bags

```
      [,1]  [,2]  [,3]
[1,] 0.604 0.100 0.295
[2,] 0.133 0.609 0.259
[3,] 0.514 0.221 0.265
[4,] 0.113 0.112 0.775
[5,] 0.258 0.502 0.240
```

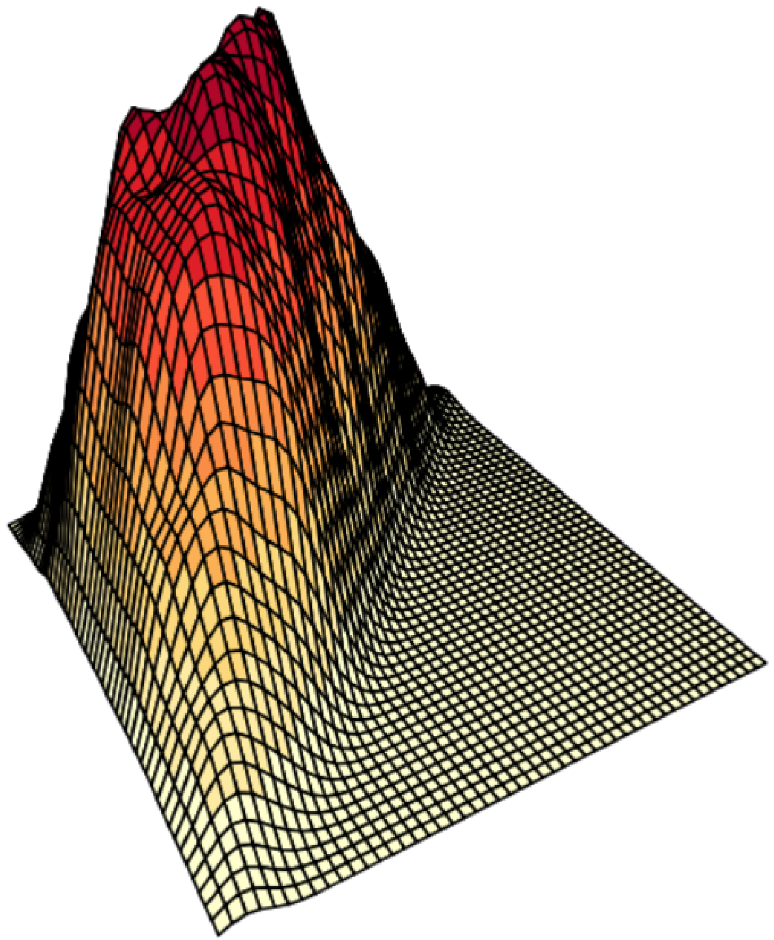
Dirichlet distribution

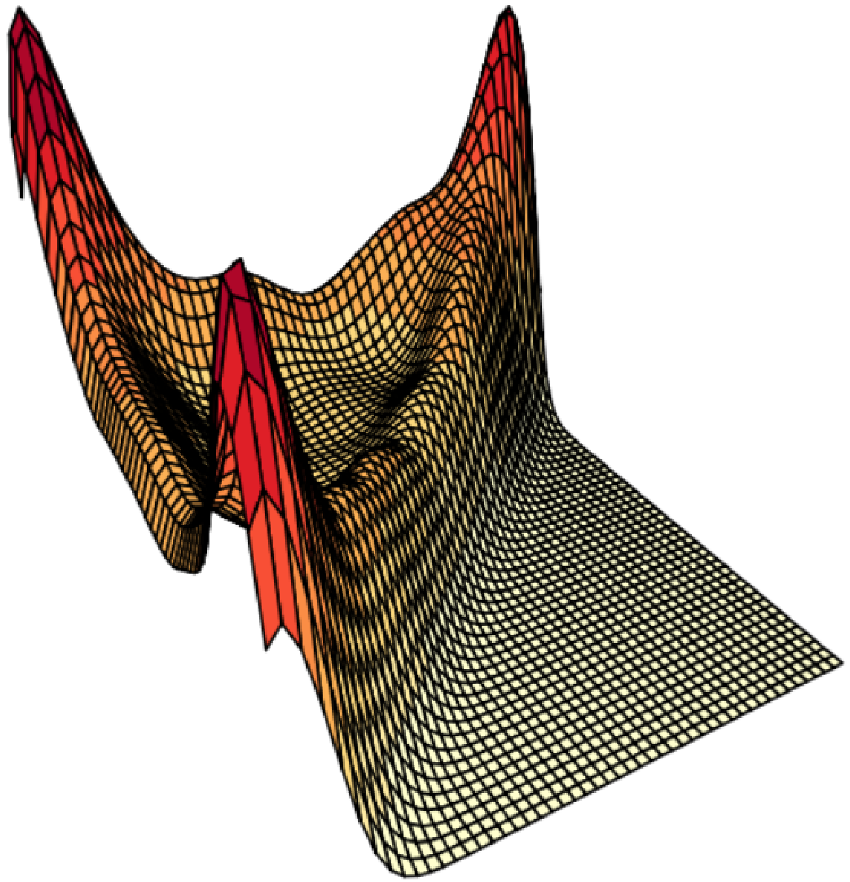


- Example: density profile of a 3-dimensional symmetric Dirichlet distribution
- Corners correspond to (1,0,0), (0,1,0), and (0,0,1) combinations
- Favors assignment of a document to a single topic

alpha and the shape of Dirichlet distribution

- Left: alpha > 1, right: alpha < 1





Let's practice!

TOPIC MODELING IN R

From word windows to dtm

TOPIC MODELING IN R

Pavel Oleinikov

Associate Director, Quantitative Analysis Center, Wesleyan University

datacamp

Word window

- Entity is a personal noun (capitalized word)
- If we take n words on the left and n words on the right of an entity, we get a word window
- E.g., *attention of* **Megara** *was turned*
  - entity is Megara
  - context is *attention of was turned*
- Also possible to tag words to differentiate the side
  - attention_L1 of_L2 was_R1 turned_R2

A document for every entity

- Combine (using `paste`) contexts of the same entity to make a document

```
docs <- df %>% group_by(entity) %>%
summarise(doc_id = first(entity),
          text = paste(text, collapse=" "))
```

- Example
  - entity - Anastasius
  - document (made of 2 word windows):
    `treasure_L1 which_L2 had_R1 bequeathed_R2 two_L1 years_L2 was_R1 overthrown_R2`

Finding entities with regular expressions & stringr

- An entity is a capitalized word
- Find it using regular expressions pattern matching

```
pattern <- "[A-Z][a-z]+"
m <- gregexpr(text, pattern)
entities <- unlist(regmatches(text, m))
```

- Regular expressions:
  - character class `[A-Z]` and `[a-z]`
  - quantifiers: {n,m} - character occurs at least `n` times and at most `m` times
  - Quantifier shortcuts: `?` is `{0,1}`, `*` is `{0,}`, `+` is `{1,}`
  - `[A-Z][a-z]+` - one uppercase letter followed by one or more lowercase

Regular expressions with groups

- Parentheses serve to group some patterns together
  - Possible to "capture" the groups
- Some entities include `St.` in them, e.g. `St. Sophia`
- New pattern:

```
p <- "(St[.] )?[A-Z][a-z]+"
```

- `(St[.] )` is a group. The `?` quantifier means the group is optional

Using capture groups to add a suffix

- Contents of a group can be back-referenced in substitution operations

```
t <- "the great Darius threw across"
gsub("^([a-z]+) ([a-z]+)", "\\1_L1 \\2_L2", t)
```

- Two groups, each matches a lowercase word `[a-z]+`
- The `^` is an anchor - specifies position in the string. `^` - the start, `$` - at the end
- The `\\1` is back-reference to contents of group 1. Its contents are substituted.
- Result

```
"the_L1 great_L2 Darius threw across"
```

Let's practice

TOPIC MODELING IN R

datacamp

Corpus alignment and classification

TOPIC MODELING IN R

Pavel Oleinikov

Associate Director, Quantitative Analysis Center, Wesleyan University

Unsupervised classification

- Topic model for `k=3` useful for telling the meaning of a named entity

```
topics = tidy(mod, matrix="gamma") %>%
spread(topic, gamma)
topics %>%
   filter(document %in% c(" Alboin", " Alexander", " Asia Minor",
   " Amorium", " Cappadocian"))
```

```
  document           `1`    `2`   `3`
  <chr>            <dbl> <dbl> <dbl>
1 " Alboin"       0.143  0.143 0.714
2 " Alexander"    0.143  0.143 0.714
3 " Amorium"      0.364  0.364 0.273
4 " Asia Minor"   0.0213 0.723 0.255
5 " Cappadocian"  0.571  0.143 0.286
```

Using pre-trained model

- Apply a pre-trained topic model to new data
- Function `posterior` in package `topicmodels`

```
model = LDA(...)
result = posterior(model, new_data)
result$topics
```

- `new_data` must be aligned with the vocabulary used in the model
- LDA algorithm iterates over items and their counts, does not "know" that it's dealing with words

Corpus alignment

- Drop words from dtm that are not part of model's vocabulary
- Function `tidy` with `matrix="beta"` extracts the terms and their probabilities

```
model_vocab <- tidy(mod, matrix="beta") %>%
      select(term) %>% distinct()
```

- Do right-join with the model's vocabulary to keep only the words the model was trained on

```
new_table <- new_doc %>%
   unnest_tokens(input=text, output=word) %>%
   count(doc_id, word) %>%
   right_join(model_vocab, by=c("word"="term"))
```

- Side effect - NA values

Handling NA values

- Right join assigns NA values in columns of rows for which there was no match

```
  doc_id word            n
  <chr>  <chr>       <int>
1 NA      emerged_r1    NA
2 NA      from_r2       NA
3 NA      horde_l1      NA
```

- We will end up with a new "document" - its name will be "NA"
- Solution: set NA counts to 0, set NA doc_id to the first "good" doc id

```
new_dtm <- new_table %>%
  arrange(desc(doc_id)) %>%
  mutate(doc_id = ifelse(is.na(doc_id), first(doc_id), doc_id),
  n = ifelse(is.na(n), 0, n)) %>%
  cast_dtm(document=doc_id, term=word, value=n)
```

Held-out data

- Machine learning: training vs. test data
- Held-out data for testing
  - Hold out a percentage of full records (same as with test datasets in ML)
  - Hold out a percentage of terms inside a document (unique to topic modeling)
- Estimate quality of fit by looking at the log-likelihood
  - "held-out log-likelihood"
- Our case: withhold full documents, no cross-validation

Let's practice!

TOPIC MODELING IN R