The great library heist

LDA vs Clustering

M Loecher

Validation

When examining a statistical method, it can be useful to try it on a very simple case where you know the "right answer". For example, we could collect a set of documents that definitely relate to four separate topics, then perform topic modeling to see whether the algorithm can correctly distinguish the four groups. This lets us double-check that the method is useful, and gain a sense of how and when it can go wrong. We'll try this with some data from classic literature.

Suppose a vandal has broken into your study and torn apart four of your books:

- *Great Expectations* by Charles Dickens
- *The War of the Worlds* by H.G. Wells
- *Twenty Thousand Leagues Under the Sea* by Jules Verne
- *Pride and Prejudice* by Jane Austen

Unsupervised Modeling

This vandal has torn the books into individual chapters, and left them in one large pile. How can we restore these disorganized chapters to their original books? This is a challenging problem since the individual chapters are **unlabeled**: we don't know what words might distinguish them into groups. We'll thus use topic modeling to discover how chapters cluster into distinct topics, each of them (presumably) representing one of the books.

We'll retrieve the text of these four books from gutenberg.

As pre-processing, we divide these into chapters and remove `stop_words`.

LDA on chapters

We can then use the `LDA()` function to create a four-topic model. In this case we know we're looking for four topics because there are four books; in other problems we may need to try a few different values of `k`.

Much as we did on the Associated Press data, we can examine per-topic-per-word probabilities.

```
##   topic  term         beta
## 1     1   joe 5.830326e-17
## 2     2   joe 3.194447e-57
## 3     3   joe 4.162676e-24
## 4     4   joe 1.445030e-02
## 5     1 biddy 7.846976e-27
## 6     2 biddy 4.672244e-69
```

For each combination topic-term, the model computes the probability of that term being generated from that topic. For example, the term "joe" has an almost zero probability of being

Results

Find the top 5 terms within each topic.

```
##   topic      term         beta
## 1     1 elizabeth 0.014107538
## 2     1     darcy 0.008814258
## 3     1      miss 0.008706741
## 4     1    bennet 0.006947431
## 5     1      jane 0.006497512
## 6     2   captain 0.015507696
```

visualization

Figure 1: The terms that are most common within each topic

Impressive

These topics are pretty clearly associated with the four books! There's no question that the topic of "captain", "nautilus", "sea", and "nemo" belongs to *Twenty Thousand Leagues Under the Sea*, and that "jane", "darcy", and "elizabeth" belongs to *Pride and Prejudice*. We see "pip" and "joe" from *Great Expectations* and "martians", "black", and "night" from *The War of the Worlds*. We also notice that, in line with LDA being a "fuzzy clustering" method, there can be words in common between multiple topics, such as "miss" in topics 1 and 4, and "time" in topics 3 and 4.

Per-document classification

Can we put the chapters back together in the correct books? We can find this by examining the per-document-per-topic probabilities, γ ("gamma").

```
##                   document topic        gamma
## 1 Great Expectations_57     1 1.351886e-05
## 2  Great Expectations_7     1 1.470726e-05
## 3 Great Expectations_17     1 2.117127e-05
```

Each of these values is an estimated proportion of words from that document that are generated from that topic. For example, the model estimates that each word in the Great Expectations_57 document has only a 0.00135% probability of coming from topic 1 (Pride and Prejudice).

Validation I

Now that we have these topic probabilities, we can see how well our unsupervised learning did at distinguishing the four books. We'd expect that chapters within a book would be found to be mostly (or entirely), generated from the corresponding topic.

First we re-separate the document name into title and chapter, after which we can visualize the per-document-per-topic probability for each.

```
##                  title chapter topic        gamma
## 1 Great Expectations      57     1 1.351886e-05
## 2 Great Expectations       7     1 1.470726e-05
## 3 Great Expectations      17     1 2.117127e-05
## 4 Great Expectations      27     1 1.919746e-05
## 5 Great Expectations      38     1 3.544403e-01
## 6 Great Expectations       2     1 1.723723e-05
```

Validation II

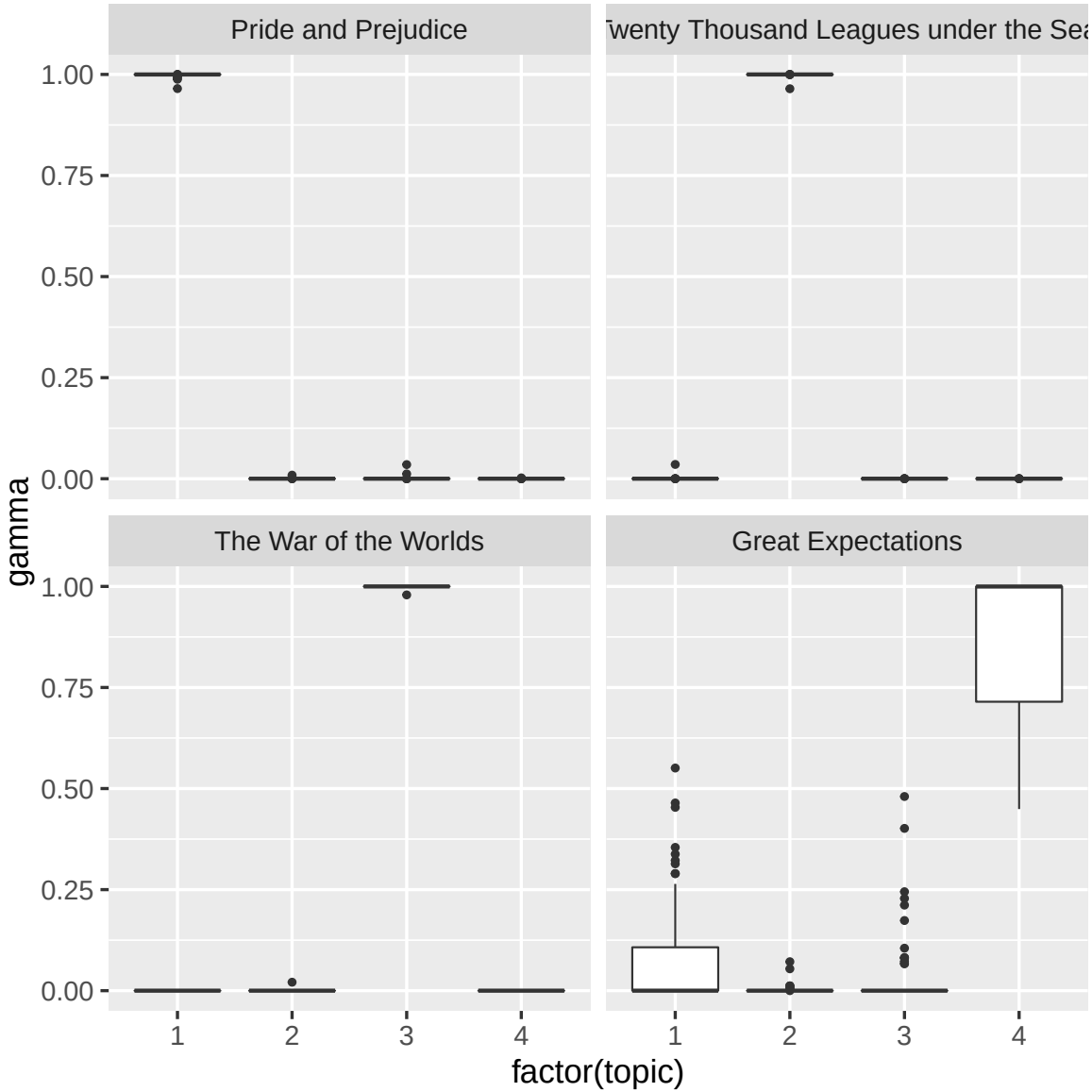

Figure 2: The gamma probabilities for each chapter within each book

Validation III

We notice that almost all of the chapters from *Pride and Prejudice*, *War of the Worlds*, and *Twenty Thousand Leagues Under the Sea* were uniquely identified as a single topic each.

It does look like some chapters from Great Expectations (which should be topic 4) were somewhat associated with other topics. Are there any cases where the topic most associated with a chapter belonged to another book? First we'd find the topic that was most associated with each chapter using `top_n()`, which is effectively the "classification" of that chapter.

```
##                  title chapter topic     gamma
## 1  Great Expectations      23     1 0.5507241
## 2 Pride and Prejudice      43     1 0.9999610
## 3 Pride and Prejudice      18     1 0.9999654
## 4 Pride and Prejudice      45     1 0.9999038
## 5 Pride and Prejudice      16     1 0.9999466
## 6 Pride and Prejudice      29     1 0.9999300
```

Validation IV

We can then compare each to the "consensus" topic for each book (the most common topic among its chapters), and see which were most often misidentified.

```
##                 title chapter topic     gamma
## 1 Great Expectations      23     1 0.5507241   Pride and
## 2 Great Expectations      54     3 0.4803234 The War of
```

We see that only two chapters from *Great Expectations* were misclassified, as LDA described one as coming from the "Pride and Prejudice" topic (topic 1) and one from The War of the Worlds (topic 3). That's not bad for unsupervised clustering!

By word assignments:

One step of the LDA algorithm is assigning each word in each document to a topic. The more words in a document are assigned to that topic, generally, the more weight (`gamma`) will go on that document-topic classification.

We may want to take the original document-word pairs and find which words in each document were assigned to which topic.

```
##                  document term count .topic
## 1 Great Expectations_57  joe    88      4
## 2  Great Expectations_7  joe    70      4
## 3 Great Expectations_17  joe     5      4
## 4 Great Expectations_27  joe    58      4
## 5  Great Expectations_2  joe    56      4
```

Classification I

We can remove chapter information to find which words were incorrectly classified.

```
##                 title chapter term count .topic          consensus
## 1 Great Expectations      57  joe    88      4 Great Expectations
## 2 Great Expectations       7  joe    70      4 Great Expectations
## 3 Great Expectations      17  joe     5      4 Great Expectations
## 4 Great Expectations      27  joe    58      4 Great Expectations
## 5 Great Expectations       2  joe    56      4 Great Expectations
```

This combination of the true book (`title`) and the book assigned to it (`consensus`) is useful for further exploration. We can, for example, visualize a **confusion matrix**, showing how often words from one book were assigned to another.

confusion matrix

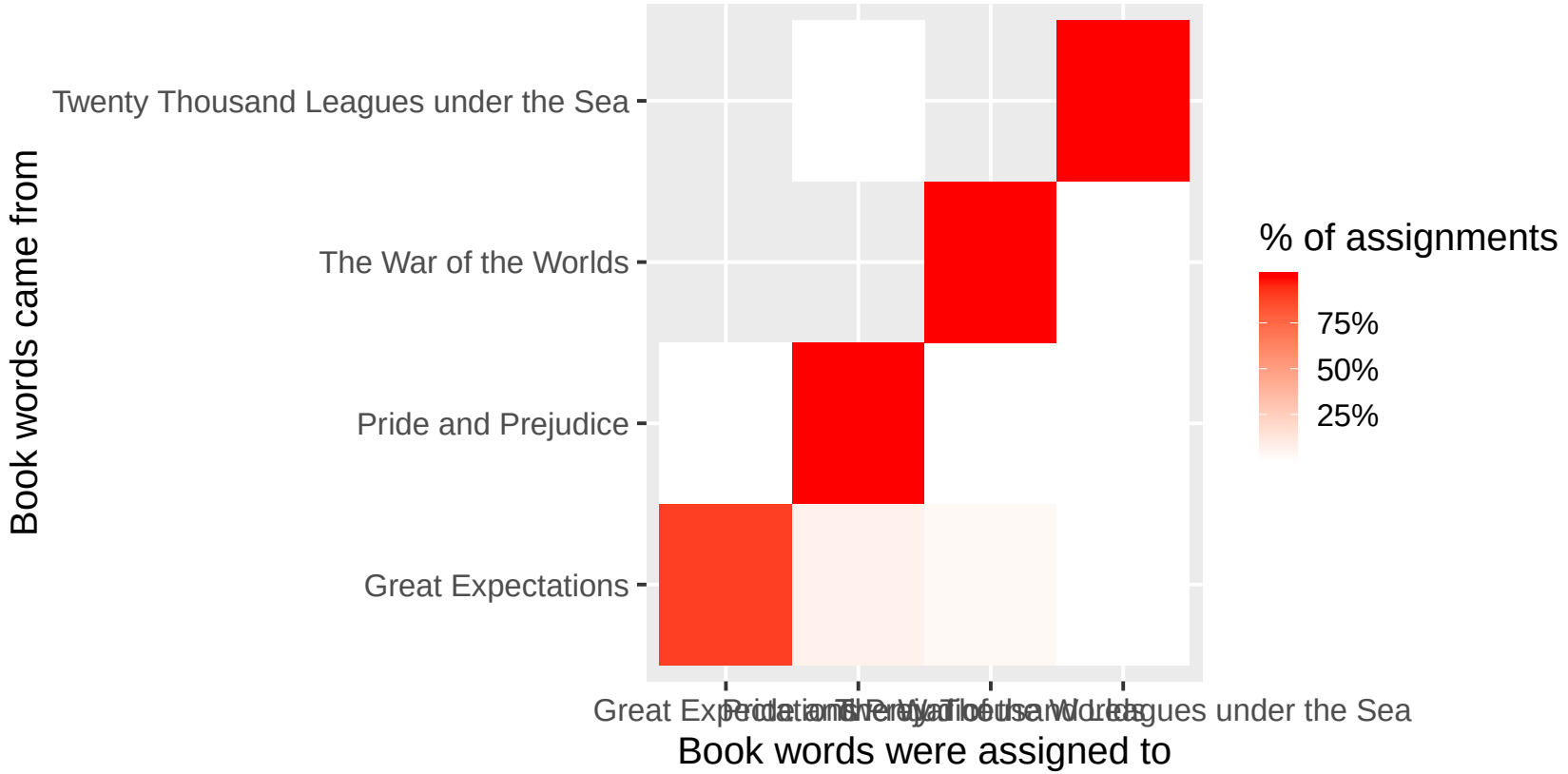

Figure 3: Confusion matrix showing where LDA assigned the words from each book. Each row of this table represents the true book each word came from, and each column represents what book it was assigned to.

Classification Errors I

We notice that almost all the words for *Pride and Prejudice*, *Twenty Thousand Leagues Under the Sea*, and *War of the Worlds* were correctly assigned, while *Great Expectations* had a fair number of misassigned words (which, as we saw above, led to two chapters getting misclassified).

What were the most commonly mistaken words?

```
##                  title                 consensus     term  n
## 1 Great Expectations   Pride and Prejudice     love 44
## 2 Great Expectations   Pride and Prejudice sergeant 37
## 3 Great Expectations   Pride and Prejudice     lady 32
## 4 Great Expectations   Pride and Prejudice     miss 26
## 5 Great Expectations The War of the Worlds     boat 25
## 6 Great Expectations   Pride and Prejudice   father 19
```

Classification Errors II

We can see that a number of words were often assigned to the Pride and Prejudice or War of the Worlds cluster even when they appeared in Great Expectations. For some of these words, such as "love" and "lady", that's because they're more common in Pride and Prejudice (we could confirm that by examining the counts).

On the other hand, there are a few wrongly classified words that never appeared in the novel they were misassigned to. For example, we can confirm "flopson" appears only in *Great Expectations*, even though it's assigned to the "Pride and Prejudice" cluster.

```
##                 document    word  n
## 1 Great Expectations_22 flopson 10
## 2 Great Expectations_23 flopson  7
## 3 Great Expectations_33 flopson  1
```

The LDA algorithm is stochastic, and it can accidentally land on a topic that spans multiple books.