Applied Scaling & Classification Techniques in Political Science

Lecture 5

Unsupervised classification methods: the structural topic model

Reference

- Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, Christopher Luca, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, David G. Rand (2014). Structural Topic Models for Open-Ended Survey Response, *American Journal of Political Science*, 58(4), 1064-1082
- Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley(2014). STM: R Package for Structural Topic Models, *Journal of Statistical Software*, https://cran.r-project.org/web/packages/stm/vignettes/stmVignette.pdf

Remember

Classification vs. Scaling

Scaling methods (such a Wordscores, Wordfish) aim to estimate the location of actors in policy space, or produce a scaling

Classification methods organize texts into a set of categories

Classification methods

Unsupervised learning methods are a class of methods that “**learn**” underlying features of text without explicitly imposing categories of interest (as it happens with supervised methods)

They use modeling assumptions and properties of the texts to estimate a **set of categories** and simultaneously **assign documents (or parts of documents)** to those categories

Therefore such models *infer* rather than *assume* the content of the categories under study

Classification methods

Unsupervised classfication methods are *text mining* methods

- By *data mining* we mean the set of tools aimed at *discovering regularities in the data*
- By *text mining* we mean the set of techniques able to *detect patterns in the texts*

Technically speaking, they are the same techniques, just applied to different data

Still, in *data mining* the information is hidden in the dimensionality of the data, whereas in *text mining* the information is contained in the texts and is visible and transparent though difficult to extract

Classification methods

Among the unsupervised classification methods, we can have…

Single membership models: these technique aims to rearrange observations (i.e., documents in a corpus) into homogenous subgroups according to some notion of **distance** among them

That's the idea of a **clustering**!

C_i will represent each document's cluster assignment and **C** = (C_1 , C_2, …, C_N) will represent a partition (clustering) of documents

Classification methods

More in details: given a **dissimilarity measure** *d* among the data (for example, Euclidean distance), *clustering* algorithms proceed by grouping (*agglomerative* methods) or splitting (*dissociative* methods) subsequently the whole set of data according to *d*

If this procedure is sequential, the method is called *hierarchical*

For example, an **agglomerative hierarchical method** is as follows: a first group is formed by taking the closest units in the data (according to *d*). Then each new aggregation occurs either forming a new group of two units, or aggregating a unit to the closets group already formed or aggregating two distinct groups.

Classification methods

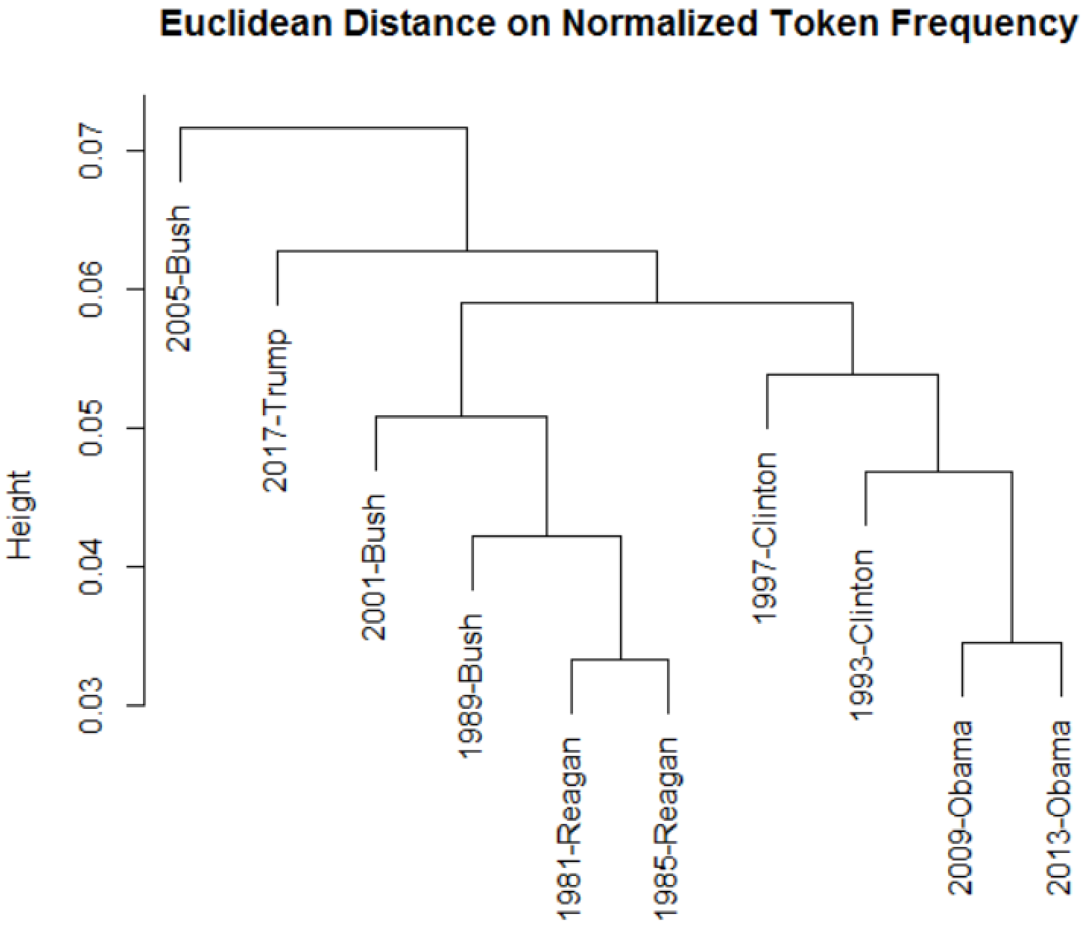
An example from the Inaugural Speeches by US Presidents corpus

The hierarchical agglomerative cluster algorithm works as follows:

1) Put each document in its own cluster

2) Identify the closest two clusters (by focusing on the maximum possible *d* between points belonging to two different clusters) and combine them into one cluster

3) Repeat the above step till all the documents are in a single cluster.

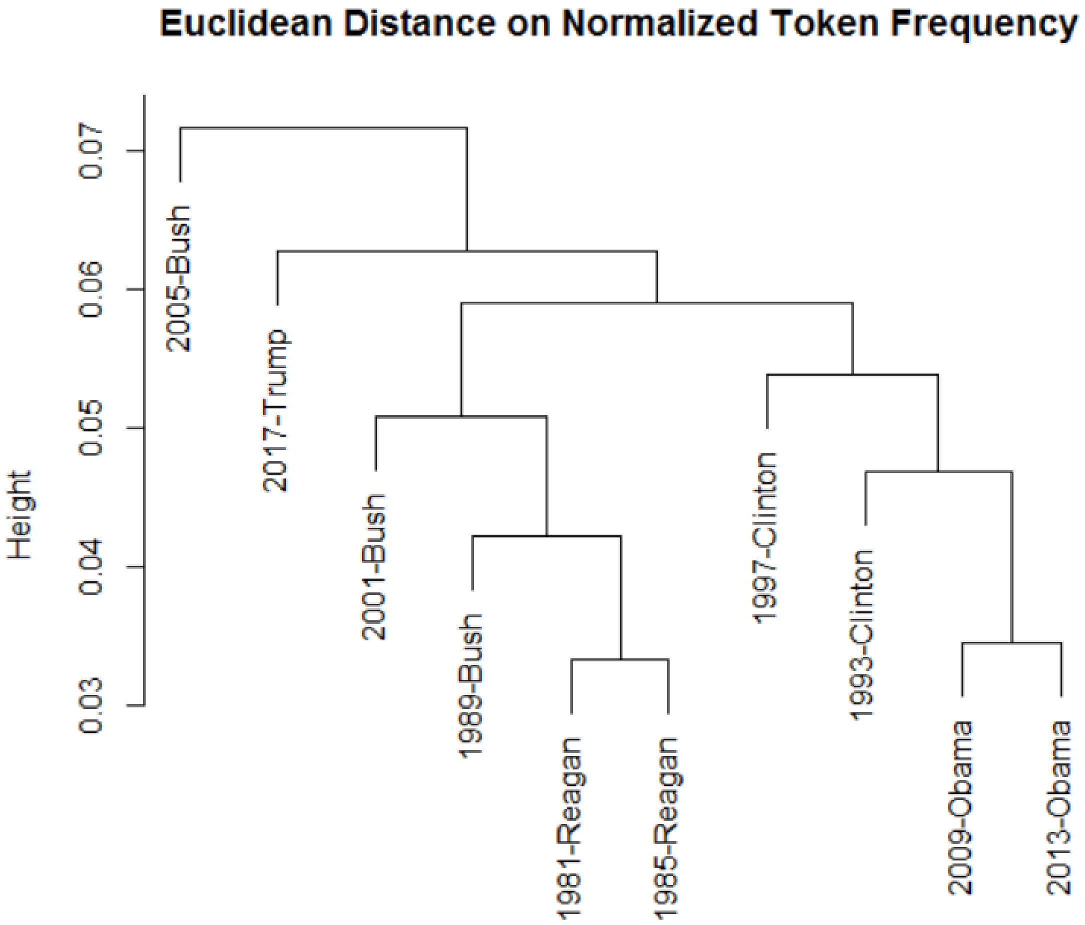


Euclidean Distance on Normalized Token Frequency

Classification methods

An example from the Inaugural Speeches by US Presidents corpus

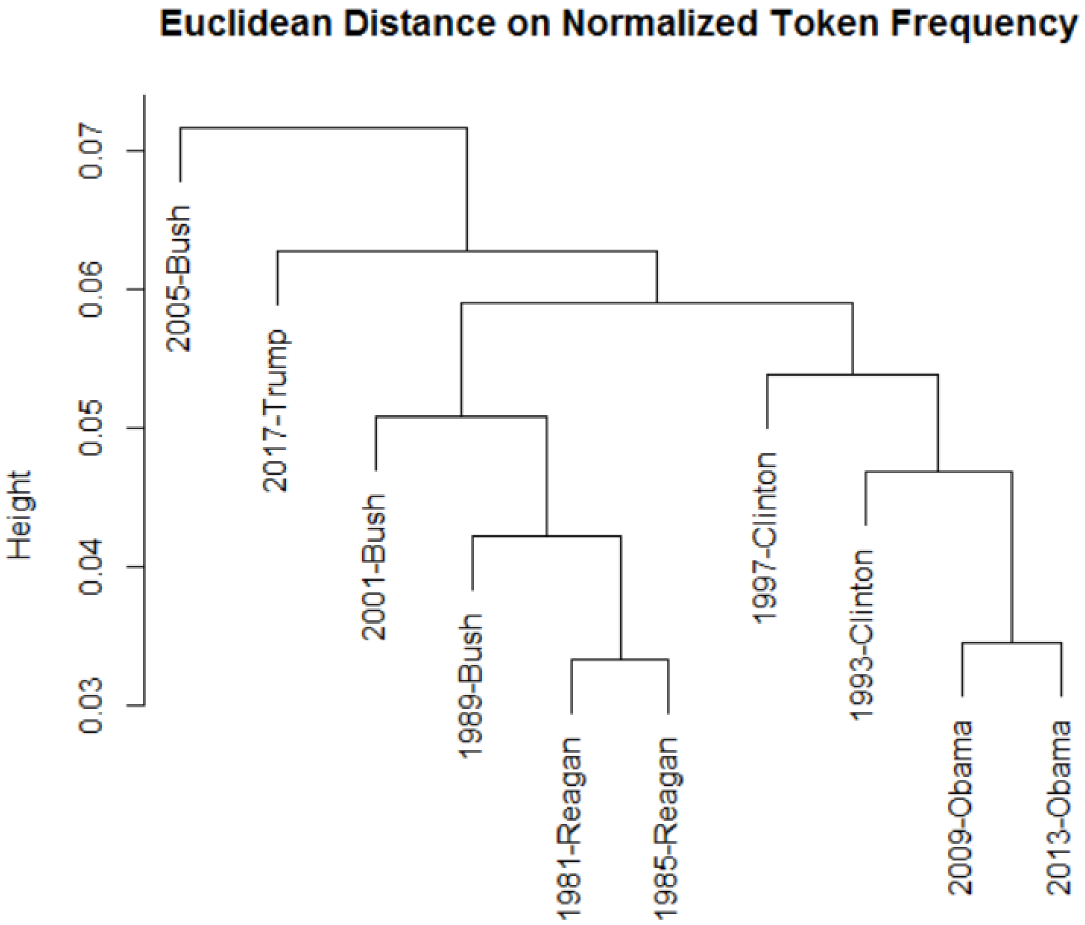
In the **dendrogram**, long vertical lines indicate more distinct separation between the groups, while short vertical bars show observations that are all close to each other



Classification methods

An example from the Inaugural Speeches by US Presidents corpus

The script to run this example is included in the “Lab 5 extra script” on the page of the course



Classification methods

Whichever method of clustering is used, in the end **one problem** remains: one has to look into the clusters to get some clue of what these clusters mean in term of semantic sense (they are unsupervised methods after all!)

Classification methods

Moreoveor, the main limit of the **single membership model** approach is that the categories are assumed to be **mutually exclusive and jointly exhaustive**

This setting could result as too restrictive when classifying more complex documents, such as political speeches. In this case, each politicians’ speech is likely to deal with a **variety of categories**

Mixed membership models (aka, **topic models**) assume precisely that each document is a mixture of categories (**topics**), meaning that a single document can be composed of multiple categories

Classification methods

To understand topic models, we need to start first of all with what we mean by “**topic**”

Statistically, a topic is a probability mass function over words, i.e., a topic is defined as a (multinomial) **distribution over the words in the vocabulary of the corpus**

Classification methods

Substantively, topics are **distinct concepts**

In congressional speech, one topic may convey attention to America’s involvement in Afghanistan, with a **high probability attached to words** like troop, war, taliban, and Afghanistan

A second topic may discuss the health-care debate, regularly using words like health, care, reform, and insurance.

Classification methods

How therefore to estimate a topic (which, remember, is **learned & discovered** rather than **assumed** by the researcher)?

We can observe **only documents and words**, **not topics** – the latter are part of the hidden (or latent) structure of documents

Still, our aim is to infer precisely the latent topic structure given the words and document

For solving this riddle, models use **the patterns of words co-occurrence within and across documents**

Classification methods

To this aim, we take advantage of the latent Dirichlet allocation (LDA) model

The basic assumption behind LDA is that each of the documents in a corpus consists of a **mixture of topics** (by “mixture” in this context we mean a set of positive values that sum to one) with **each word** within a given document belonging to **exactly one topic**

Moreover each word is assumed to be conditionally independent given its topic

Classification methods

As a result, each document can be represented as a **vector of proportions** that denote what **fraction of the words belongs to each topic**

In single membership models, on the contrary, **each document is restricted to only one topic**, so all words within it are generated from the same distribution

Classification methods

LDA “recreates” the documents in the corpus by adjusting the relative importance of topics in documents and words in topics **iteratively**, that is…

…given a corpus, LDA **backtracks** and tries to figure out what topics would create the documents included in the corpus in the first place!

Classification methods

The assumed data generating process for each document:

Let’s suppose you have N documents in your corpus and the total number of words (features) in your document-term-matrix is M

1. You begin by telling to the algorithm how many topics (K) you think there are in your corpus. You can either use an informed estimate (e.g. results from a previous analysis), or simply trial-and-error (more on this later)

Classification methods

LDA then splits the original TDM of our corpus into two lower dimensions matrices (an example with K=2)

	w1	w2	w3	wm
d1	0	2	3	1
d2	2	0	2	4
dn	3	1	2	3

	k1	k2
d1	??	??
d2	??	??

This is a **document-topics matrix** with dimension (N, K)

	w1	w2	w3	wm
k1	??	??	??	??
k2	??	??	??	??

This is a **topic-terms matrix** with dimension (K, M)

N = total number of documents (d)
K = total number of topics (k)
M = the vocabulary size (words: w)

Classification methods

2. A **topic mixture** $\theta_{d,k}$ is then drawn for the document *d* according to a Dirichlet distribution over the fixed set of K topics (say K=3, $\theta_{d,k}$ = 0.3, i.e., 30% of the words in document *d* refers to topic 1; 0.4, 0.4)

Dirichlet distributions provide good approximations to word distributions in documents and are computationally convenient

Classification methods

3. The **probability** of observing a word in the vocabulary under a certain topic ($\beta_{k,w}$) is then given by a two-step process:

a) the first step is to draw the topic;

b) conditional on topic assignment, the actual word is drawn from a multinomial distribution

Classification methods

Each *w* word in a document *d* is assigned only to one topic. However, if a word **appears twice** in a document, each word may be assigned to different topics

LDA considers that any given topic will have a high probability of generating certain words and a low probability of generating other words

Classification methods

So in summary, from the LDA view, documents are created by the following process:

1. Choose the number of topics from which each document is generated
2. Estimate the proportion of the document to come from each topic
3. Generate appropriate words from the topics chosen in the proportions specified

More in details…

Classification methods

After having defined the total number of topics K to discover, you start with some given values for $\theta_{d,k}$ and $\beta_{k,w}$

This first assignment already gives you both topic representations of all the documents and word distributions of all the topics (albeit not necessarily a very good ones)

So to improve on them, **both values are updated** throughout the LDA process in the following way:

for each document d…

….go through each word w in d…

Classification methods

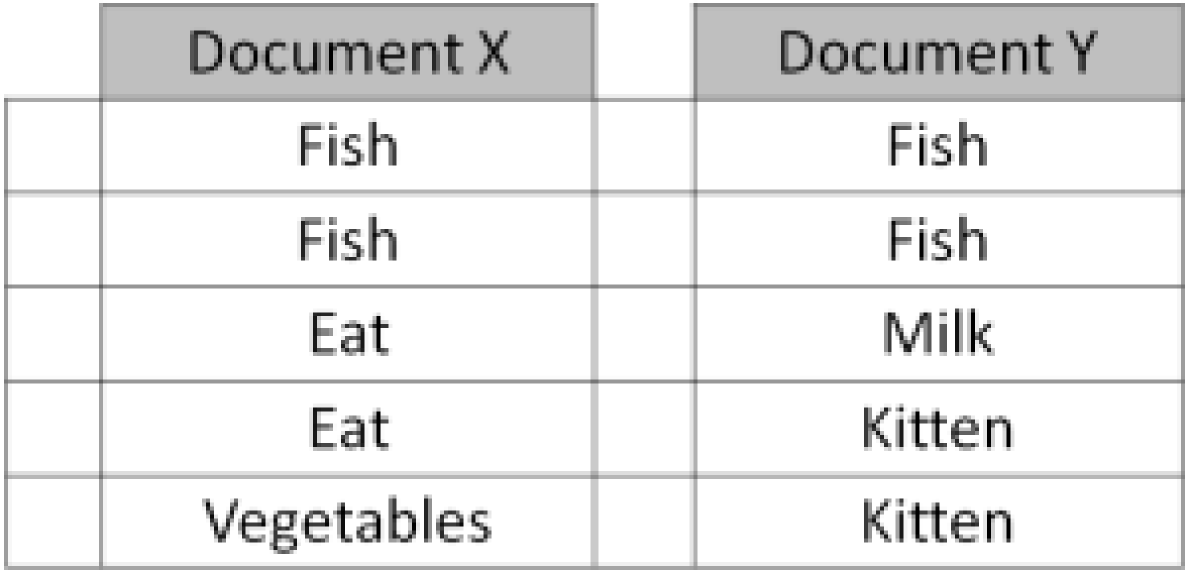
...And for each topic *k*, compute two things:

1) p(topic k | document d) = the proportion of words in document d that are currently assigned to topic k, i.e., **how prevalent are topics in the document?**
2) p(word w | topic k) = the proportion of assignments to topic k over all documents that come from this word w, i.e., **how prevalent is that word across topics?**

What we mean by that? An example

Classification methods

Imagine you are analyzing two documents about foods and animals with the following words:

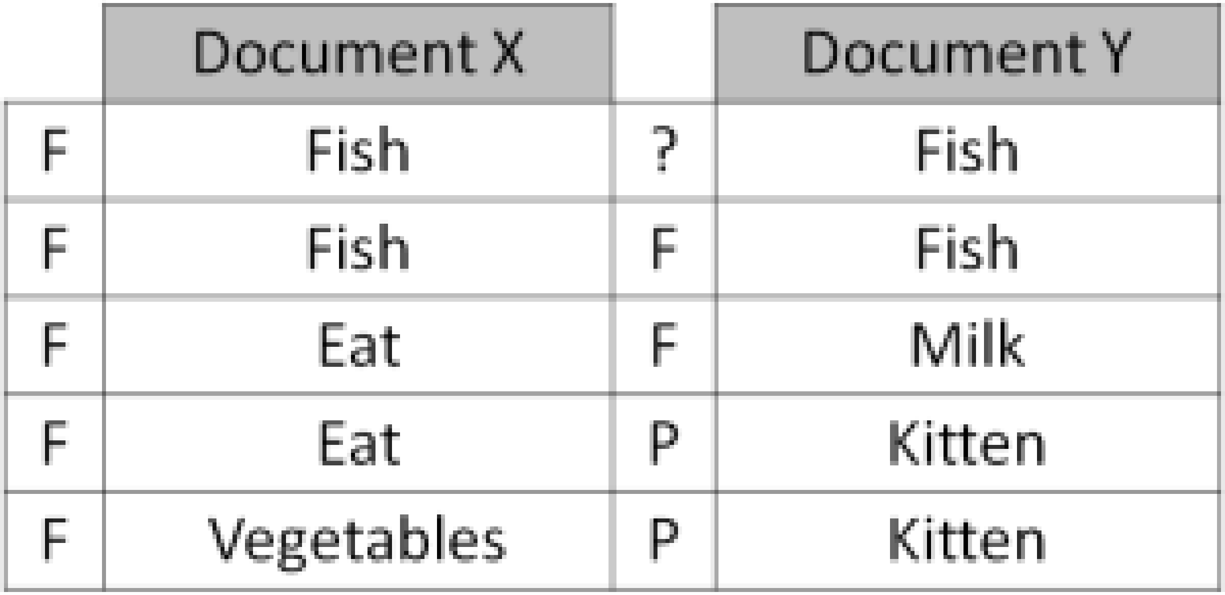


	Document X		Document Y
	Fish		Fish
	Fish		Fish
	Eat		Milk
	Eat		Kitten
	Vegetables		Kitten

You select at the beginning K=2

Classification methods

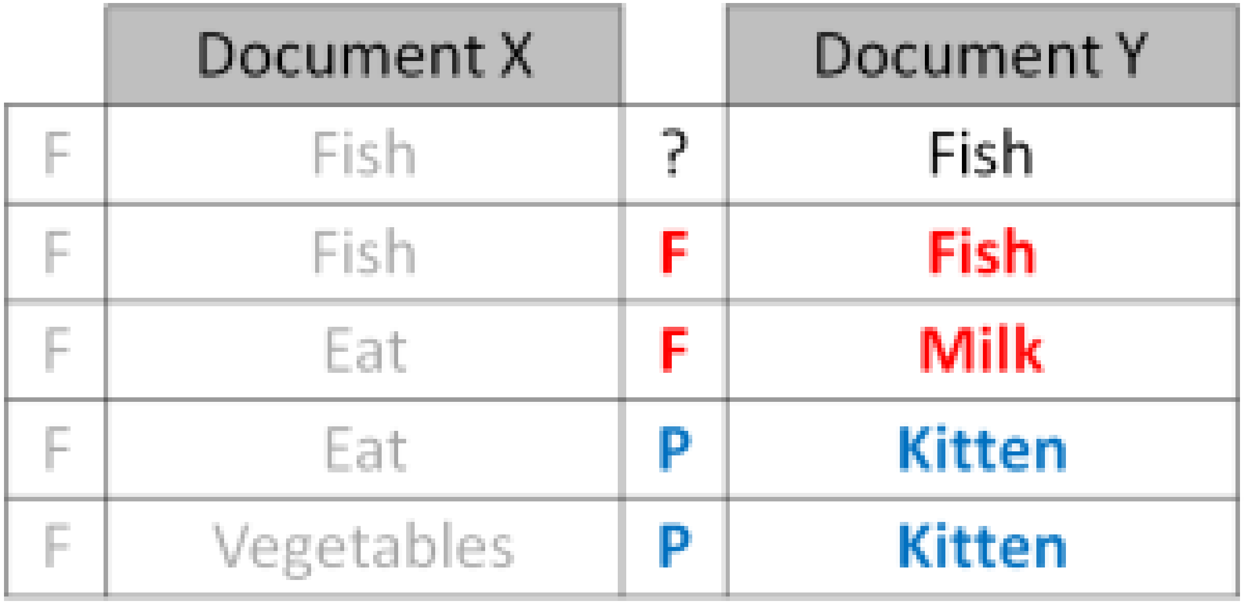
Imagine now that we are now checking the possible **new topic assignment** (across F and P, the two topics) for the word “fish” in Doc Y after the first assignment done by LDA:



	Document X		Document Y
F	Fish	?	Fish
F	Fish	F	Fish
F	Eat	F	Milk
F	Eat	P	Kitten
F	Vegetables	P	Kitten

Classification methods

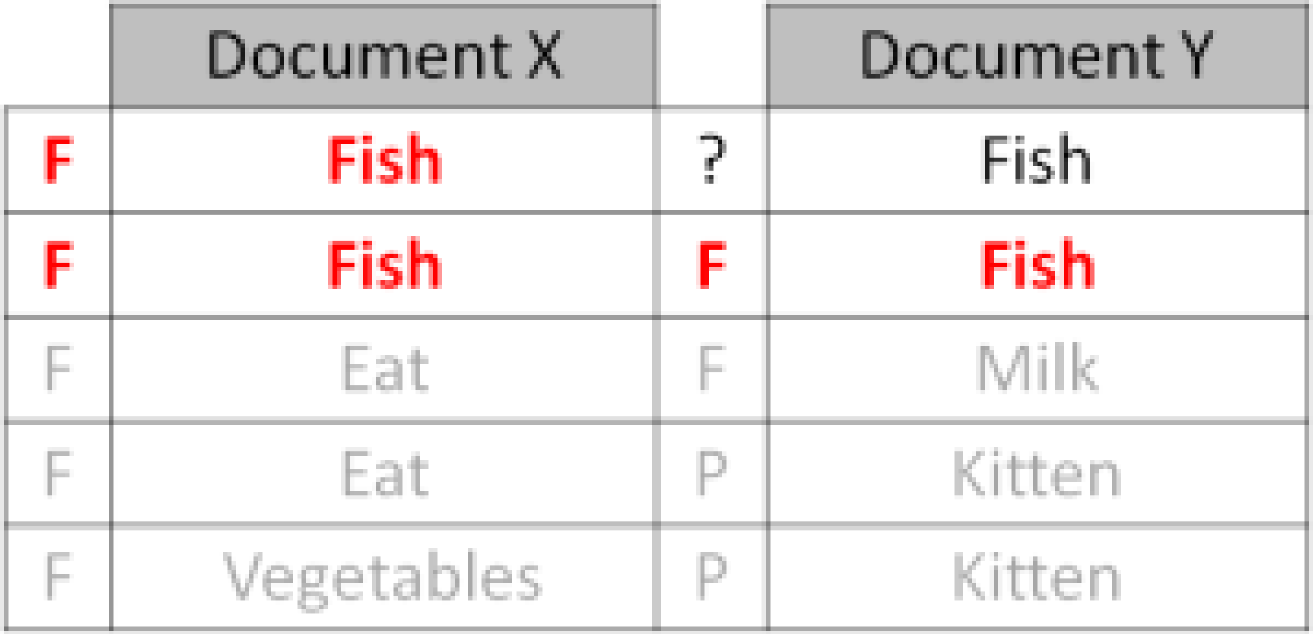
How prevalent are topics in the document? Since the words in Doc Y are assigned to Topic F and Topic P in a 50-50 ratio, the remaining “fish” word seems equally likely to be about either topic.



	Document X		Document Y
F	Fish	?	Fish
F	Fish	F	Fish
F	Eat	F	Milk
F	Eat	P	Kitten
F	Vegetables	P	Kitten

Classification methods

How prevalent is that word across topics? The “fish” words across both documents appears nearly half of the time in Topic F words (3/7), but 0% among Topic P words



	Document X		Document Y
F	Fish	?	Fish
F	Fish	F	Fish
F	Eat	F	Milk
F	Eat	P	Kitten
F	Vegetables	P	Kitten

Classification methods

Weighing conclusions from the two criteria (i.e., by multiplying the two previous probabilities), we would assign the “fish” word of Doc Y to Topic F (Doc Y might then be a document on what to feed kittens?)

Classification methods

By following this procedure, we (eventually) reassign w to a new topic, where topic k is chosen with probability p(topic k | document d) * p(word w | topic k)

According to our generative model, this is essentially the **probability that topic k generated word w**

When doing it, **we are assuming that all topic assignments** except for the current word in question, **are correct**, and then updating the assignment of the current word using our model of how documents are generated

Classification methods

After repeating the previous step a **large number of times**, you’ll eventually reach a roughly steady state where your assignments (the document topic and topic term distributions) are pretty good

This is the **convergence point** of LDA

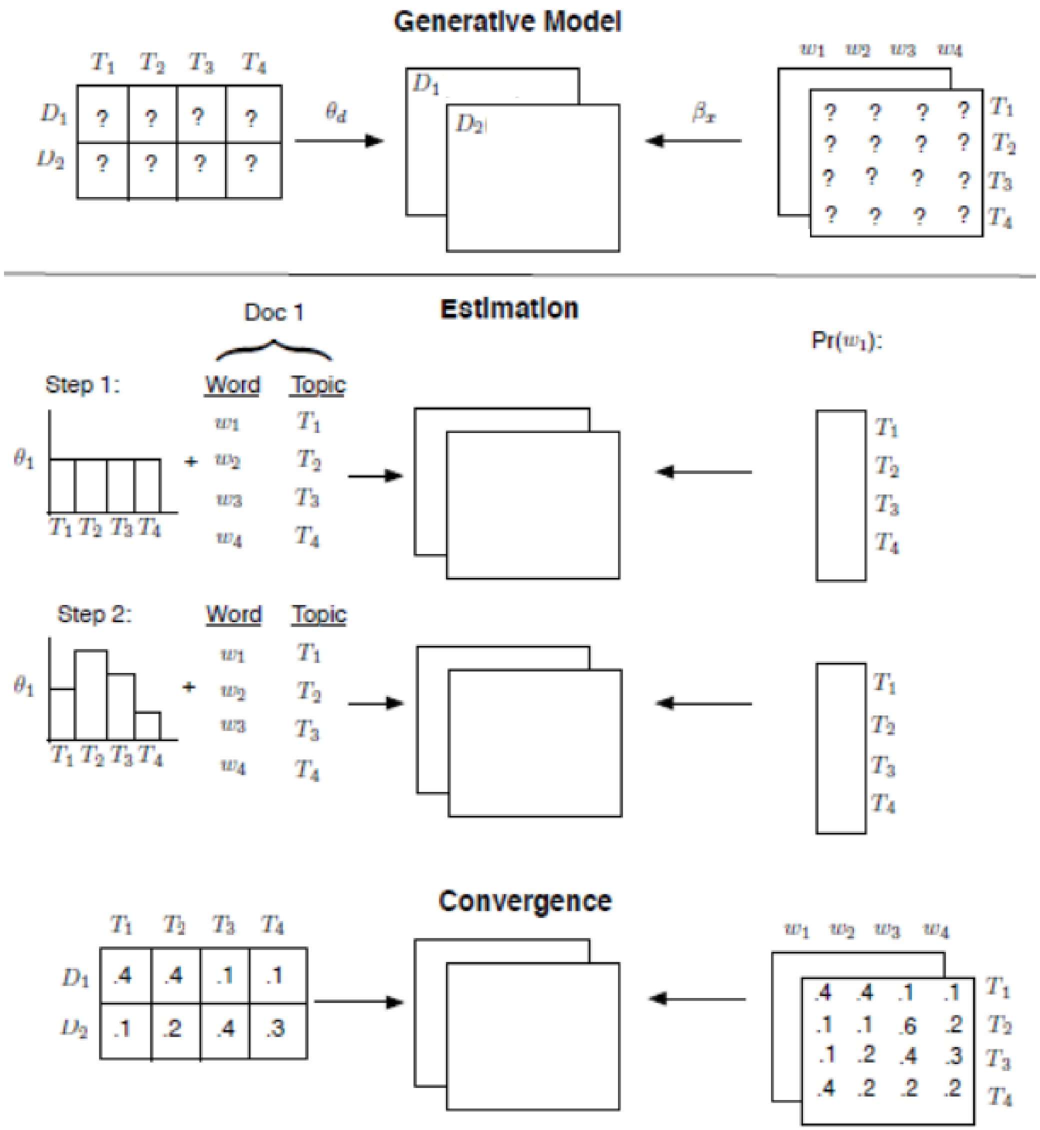
LDA uses a process known as *collapsed Gibbs sampling*: Gibbs sampling works by performing a random walk in such a way that reflects the characteristics of a desired distribution. The starting point of the walk is chosen at random

Classification methods

Once the convergent point is reached, use the obtained assignments to estimate the:

1. **Document-topic proportions** (by counting the proportion of words assigned to each topic *within* that document)
2. **Topic-word proportions** (by counting the proportion of words assigned to each topic overall, i.e., *across documents*)

Classification methods



	T_1	T_2	T_3	T_4
D_1	?	?	?	?
D_2	?	?	?	?

w_1	w_2	w_3	w_4	
?	?	?	?	T_1
?	?	?	?	T_2
?	?	?	?	T_3
?	?	?	?	T_4

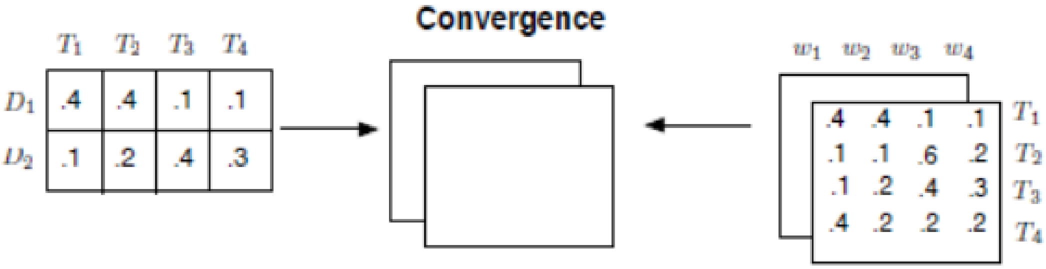
Word	Topic
w_1	T_1
w_2	T_2
w_3	T_3
w_4	T_4

Word	Topic
w_1	T_1
w_2	T_2
w_3	T_3
w_4	T_4

	T_1	T_2	T_3	T_4
D_1	.4	.4	.1	.1
D_2	.1	.2	.4	.3

w_1	w_2	w_3	w_4	
.4	.4	.1	.1	T_1
.1	.1	.6	.2	T_2
.1	.2	.4	.3	T_3
.4	.2	.2	.2	T_4

Classification methods



	T_1	T_2	T_3	T_4
D_1	.4	.4	.1	.1
D_2	.1	.2	.4	.3

Convergence

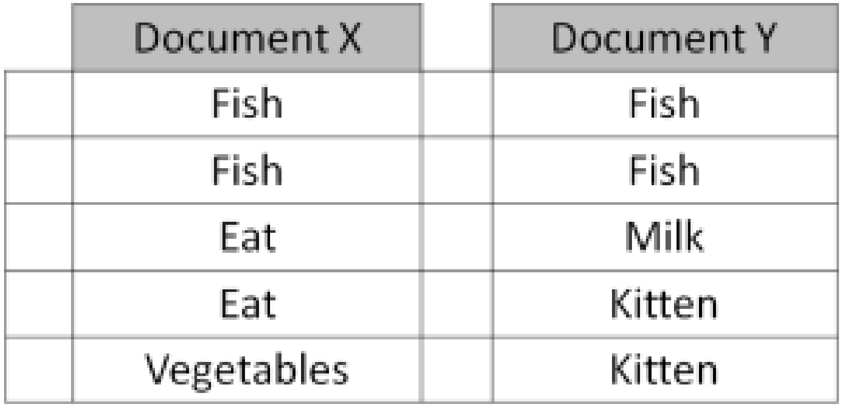
w_1	w_2	w_3	w_4	
.4	.4	.1	.1	T_1
.1	.1	.6	.2	T_2
.1	.2	.4	.3	T_3
.4	.2	.2	.2	T_4

Of course, the sum of the topic proportions across all topics for a document is 1

Of course, the sum of the topic probabilities for a word, across all topics, is 1

Classification methods

Going back to our example



	Document X	Document Y
	Fish	Fish
	Fish	Fish
	Eat	Milk
	Eat	Kitten
	Vegetables	Kitten

	fish	eat	vegetables	milk	kitten
D1	2	2	1	0	0
D2	2	0	0	1	2

	K1	K2
D1	?	?
D2	?	?

Document-topics matrix

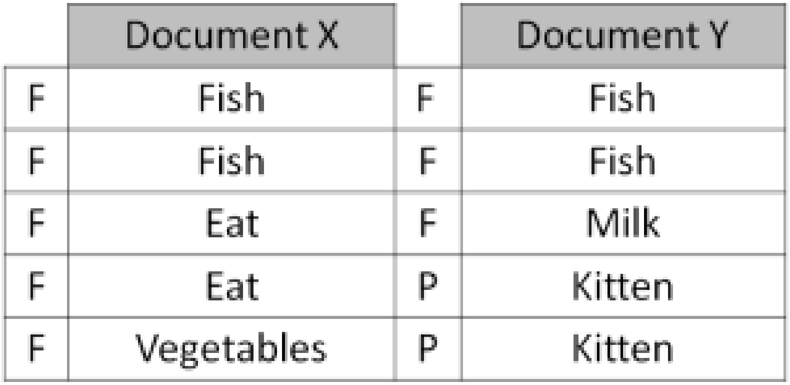
	fish	eat	vegetables	milk	kitten
K1	?	?	?	?	?
K2	?	?	?	?	?

Topic-terms matrix

Classification methods

Going back to our example (where K1=F; K2=P)

	fish	eat	vegetables	milk	kitten
D1	2	2	1	0	0
D2	2	0	0	1	2



	Document X		Document Y
F	Fish	F	Fish
F	Fish	F	Fish
F	Eat	F	Milk
F	Eat	P	Kitten
F	Vegetables	P	Kitten

	K1	K2
D1	1	0
D2	0.6	0.4

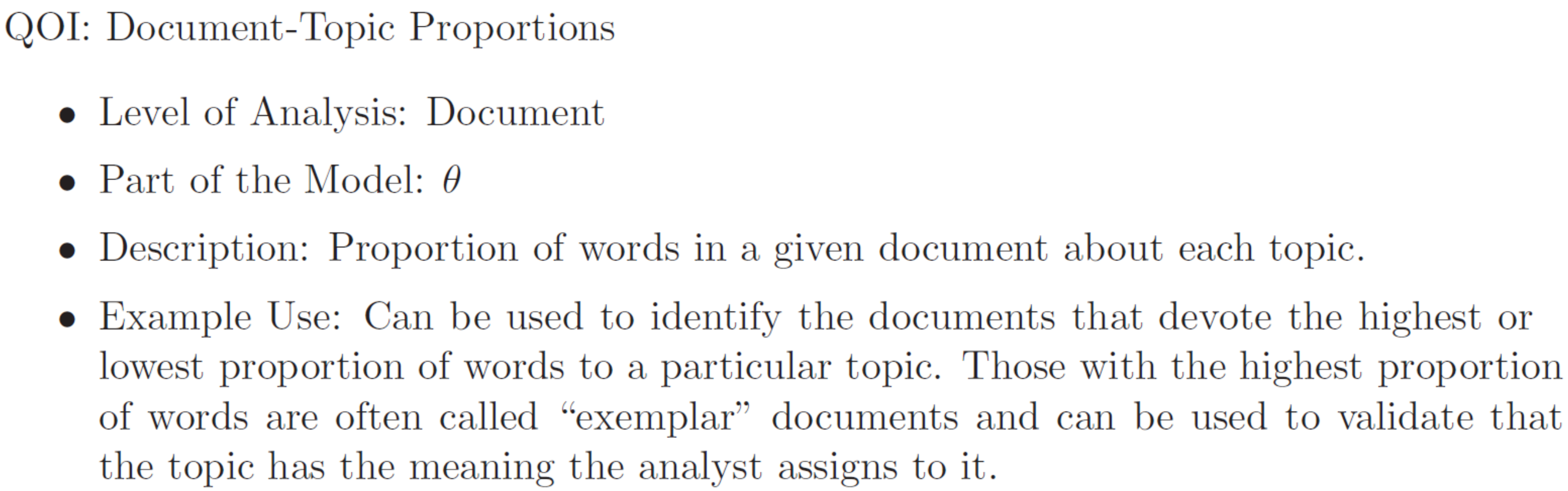
Document-topics matrix

	fish	eat	vegetables	milk	kitten
K1	0.5	0.25	0.125	0.125	0
K2	0	0	0	0	1

Topic-terms matrix

Classification methods

The quantities of interest from a Topic Model:

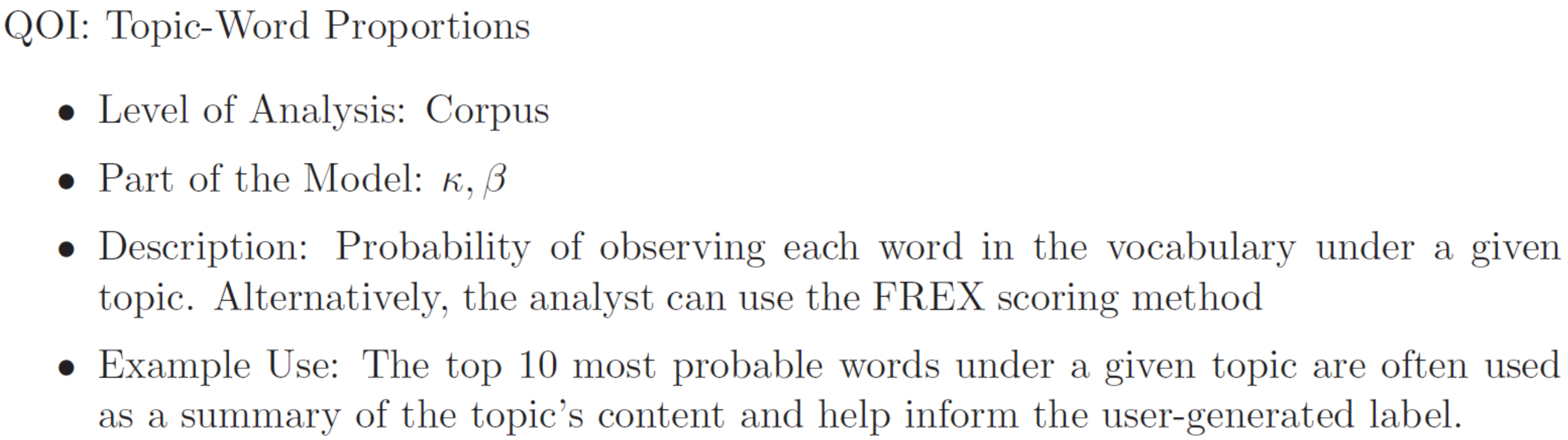


QOI: Document-Topic Proportions

- Level of Analysis: Document
- Part of the Model: θ
- Description: Proportion of words in a given document about each topic.
- Example Use: Can be used to identify the documents that devote the highest or lowest proportion of words to a particular topic. Those with the highest proportion of words are often called "exemplar" documents and can be used to validate that the topic has the meaning the analyst assigns to it.

Classification methods

The quantities of interest from a Topic Model:



QOI: Topic-Word Proportions

- Level of Analysis: Corpus
- Part of the Model: κ, β
- Description: Probability of observing each word in the vocabulary under a given topic. Alternatively, the analyst can use the FREX scoring method
- Example Use: The top 10 most probable words under a given topic are often used as a summary of the topic's content and help inform the user-generated label.

Classification methods

The challanges of any topic model:

1. *Understanding the semantic meaning of a topic*

A **semantically interpretable topic** has two qualities:

(a) it is *coherent/cohesive* in the sense that high-probability words for the topic tend to co-occur (i.e., *do top words of one topic tend to co-occur across documents*?)

Therefore semantic coherence is a property of the “within topics”

Classification methods

Semantic coherence **however** only addresses whether a topic is internally consistent (i.e., it checks if we are evaluating a well-defined concept)

It **does not penalize topics that are alike**

This could be a problem!

Classification methods

The challanges of any topic models:

1. *Understanding the semantic meaning of a topic*

A **semantically interpretable topic** has two qualities

(b) it is *exclusive* in the sense that the top words for that topic are unlikely to appear within top words of other topics (i.e., *are the top words of one topic different from the top words of other topics*?): if words with high probability under topic *k* have low probabilities under other topics, then we say that topic *k* is exclusive

Therefore semantic exclusivity is a property of the “between topics”

Classification methods

The challanges of any topic models:

1. *Understanding the semantic meaning of a topic*

A topic that is both *cohesive and exclusive* is more likely to be **semantically useful**

The frequency/exclusivity (**FREX**) scoring summarizes words according to their probability of appearance under a topic and the exclusivity to that topic

These words provide more semantically intuitive representations of each topic

Classification methods

The challanges of any topic models:

2. *How many topics?*

The analyst **must choose the number of topics**. There is no “right” answer to this choice. Varying the number of topics varies the level of granularity of the view into the data

Therefore, the choice will be dependent both on the nature of the documents under study and the goals of the analysis

The appropriateness of particular levels of aggregation will vary with your research questions

Classification methods

Largely, the answer will be also related to the semantic meaning of the topics extracted

The researcher is tasked with selecting any number of topics (K) and confirming that those recovered are substantively meaningful

Classification methods

Given that is practically impossible to guess the exact number of topics in the corpus (although new empirically tests have been introduced in the literature…), a good practice is beginning with a **wider number of topics** rather than a potentially too narrow one

Then a researcher should settled on a specification of K lower that the initial one when she found that at higher specifications, substantively-meaningful topics were being divided up in ways that were less amenable to testing her hypotheses

Examining the terms with highest probabilities of belonging to each topic and reading the documents with highest probabilities of belonging to it gives the researcher a sense of the substantive meaning of each topic

Classification methods

Structural Topic Model (STM) innovates on the models just described in two different ways:

First: topic proportions ($\theta_{d,k}$) can be **correlated**: this is a reasonable assumption given that in documents topics discussed are correlated!

For example, if a manifesto contains discussion of Topic X (e.g. administrative reform), the probabilities that it will also contain discussion of Topics Y (e.g. curbing public works) and Z (e.g. reducing the number of Lower House members), are not independent of each other, but correlated

In this sense, STM fits a Correlated Topic Model (rather than a LDA)

Classification methods

Graphical depictions of the correlation between topics provide insight into the organizational structure at the corpus level

In essence, the model identifies when two topics are likely to co-occur (by focusing on positive correlation) within a document

Classification methods

Structural Topic Model (STM) innovates on the models just described in two different ways:

Second: in all topic models, the analyst estimates for each document the proportion of words attributable to each topic, providing a measure of *topic prevalence*. The model also calculates the words most likely to be generated by each topic, which provides a measure of *topical content*

However, in standard LDA, the document collection is assumed to be **unstructured**; that is, each document is assumed to arise from the same data-generating process irrespective of additional information the analyst might possess

Classification methods

By contrast, a STM framework is designed to incorporate additional information about the document or its author into the estimation process

That is, rather than assuming that **topical prevalence** (i.e., the frequency with which a topic is discussed) and **topical content** (i.e., the words used to discuss a topic) **are constant** across all documents, the analyst can incorporate covariates over which we might expect to see variance

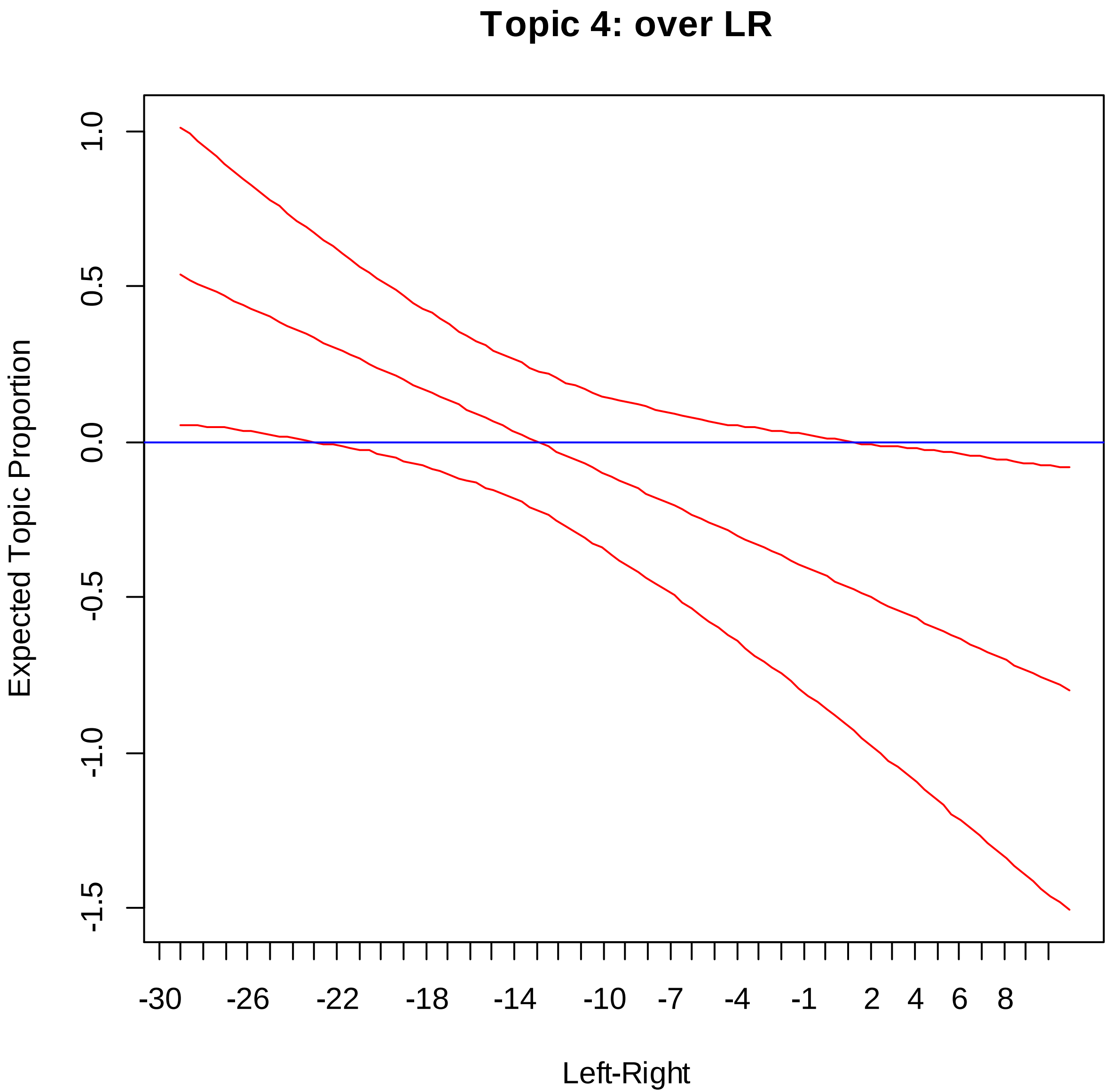
Classification methods

This allows to measure systematic changes in topical prevalence and topical content over the conditions in our experiment, as measured by the *X* covariates for prevalence and the *U* covariates for content

Thus, for example, we can easily obtain measures of how our treatment condition affects how often a topic is discussed (prevalence)!

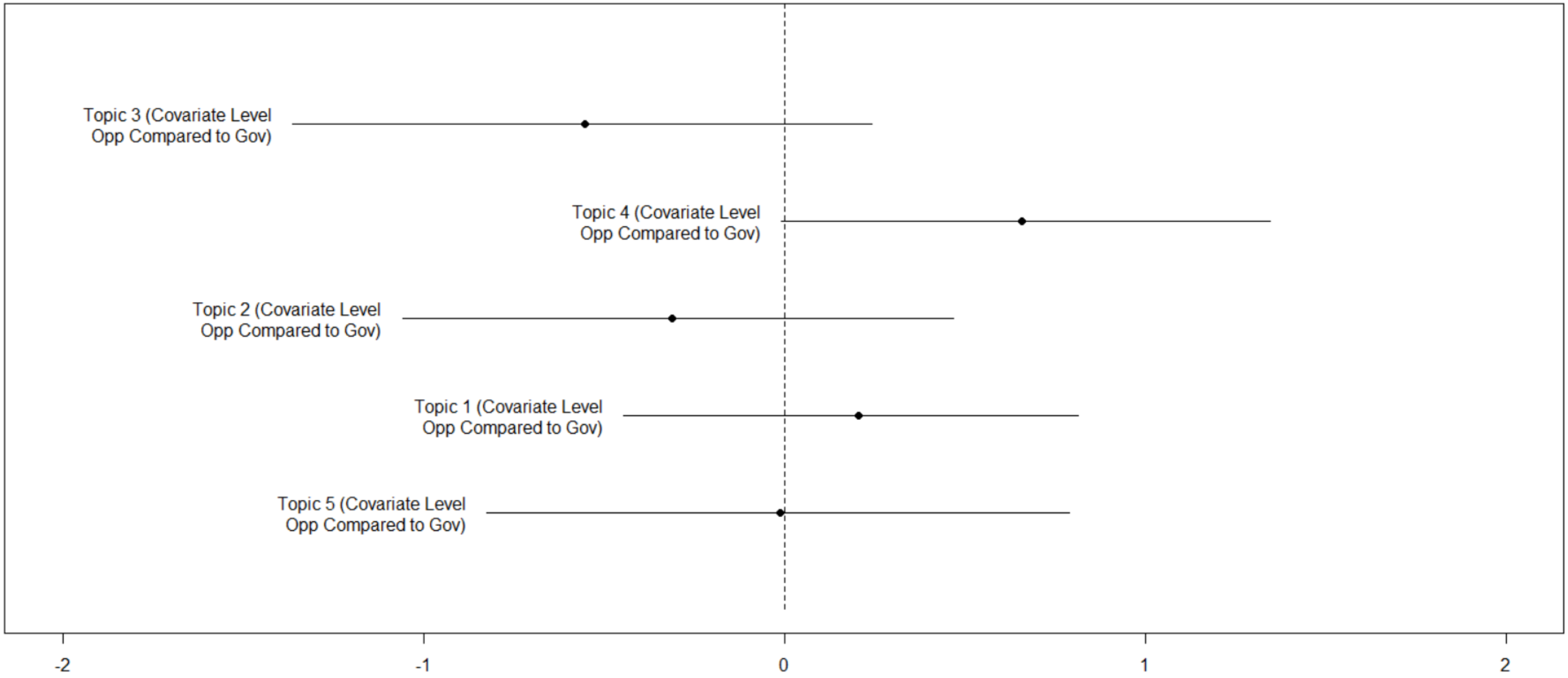
- ➢ for example, do documents of left parties discuss more about a given topic than documents of right parties?

Classification methods



Topic 4: over LR

Classification methods



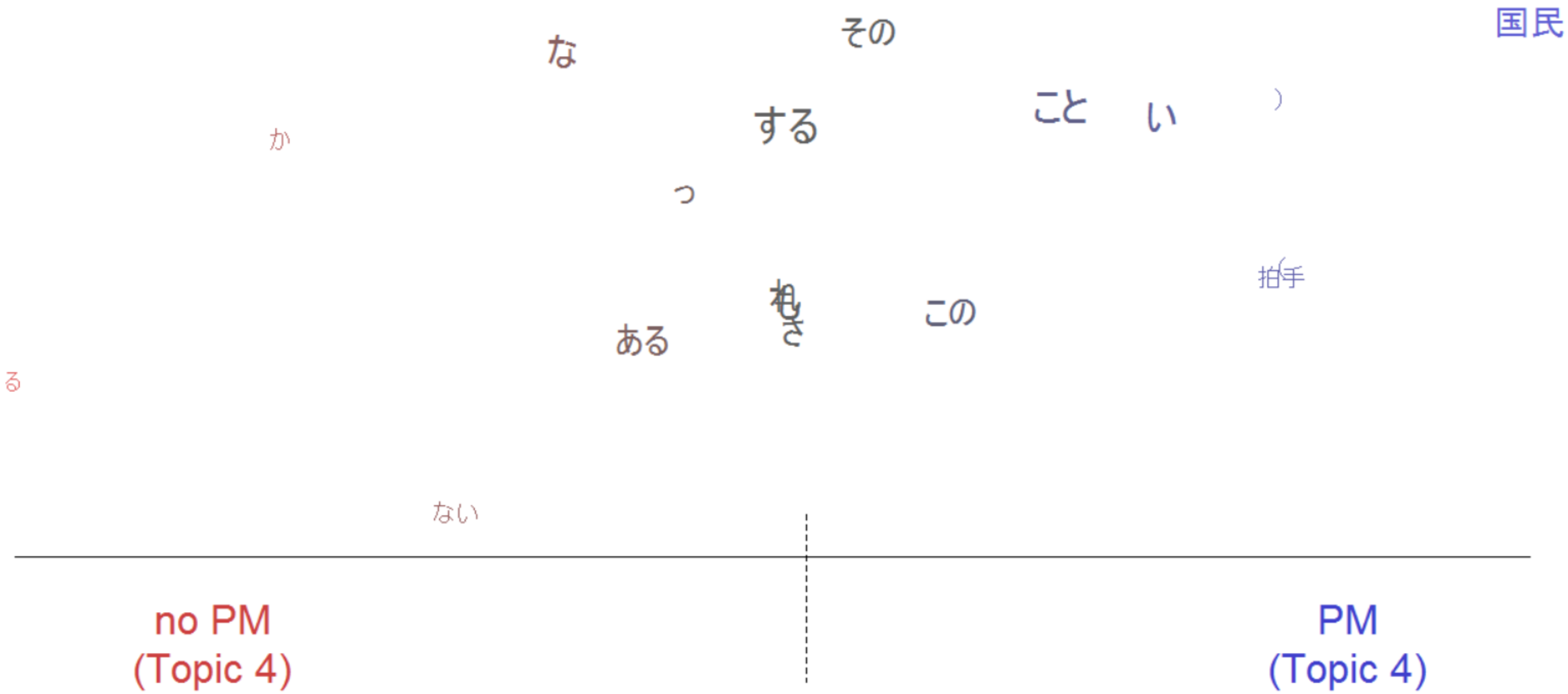
Reported coefficient:
«opposition – government»

Classification methods

Moreover, we can easily obtain measures of how the language used to discuss the same topic (content)

- ➢ for example, when men politicians discuss about a particular topic do they use the same words than female politicians?

Classification methods



Classification methods

STM conducts this type of analysis, while **simultaneously** estimating the topics

This is more efficient than doing the two processes in separated steps: aka, first the topic analysis, and then running an analysis on the topic extracted

Classification methods

In the STM framework, the researcher has therefore the option to choose covariates to incorporate in the model

These covariates inform either the **topic prevalence** or the **topical content** latent variables with observed information about the respondent

The analyst will want to include a covariate in the topical prevalence portion of the model (*X*) when she believes that the observed covariate will affect *how much* the respondent is to discuss a particular topic

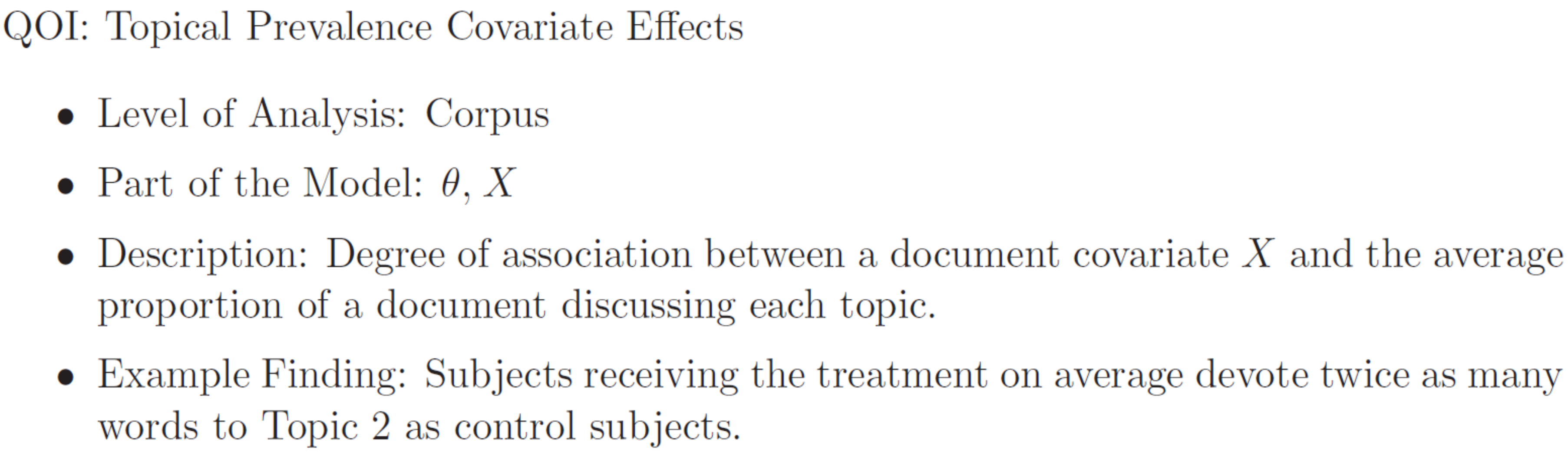
The analyst also has the option to include a covariate in the topical content portion of the model (*U*) when she believes that the observed covariate will affect *the words which a respondent uses* to discuss a particular topic.

Classification methods

These two sets of covariates can overlap, suggesting that the topic proportion and the way the topic is discussed change with particular covariate values

Classification methods

The quantities of interest from a **Structural** Topic Model (beyond the previous two…)

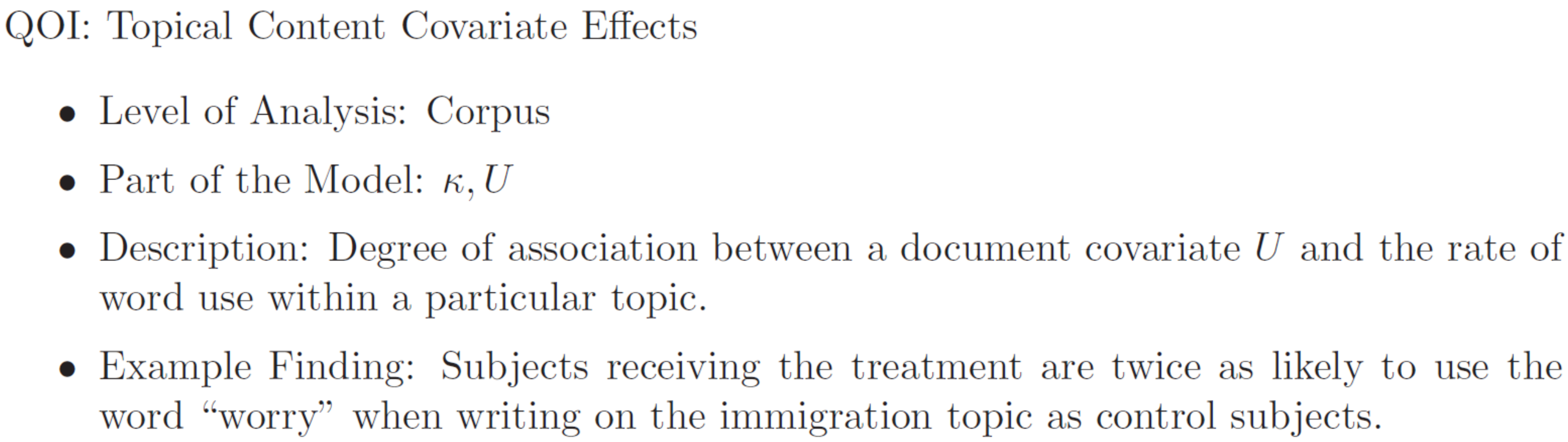


QOI: Topical Prevalence Covariate Effects

- Level of Analysis: Corpus
- Part of the Model: θ, X
- Description: Degree of association between a document covariate X and the average proportion of a document discussing each topic.
- Example Finding: Subjects receiving the treatment on average devote twice as many words to Topic 2 as control subjects.

Classification methods

The quantities of interest from a **Structural** Topic Model (beyond the previous two…)



QOI: Topical Content Covariate Effects

- Level of Analysis: Corpus
- Part of the Model: κ, U
- Description: Degree of association between a document covariate U and the rate of word use within a particular topic.
- Example Finding: Subjects receiving the treatment are twice as likely to use the word "worry" when writing on the immigration topic as control subjects.

STM and R

```
install.packages("stm", repos='http://cran.us.r-project.org')
install.packages("igraph", repos='http://cran.us.r-project.org')
install.packages("stmBrowser", repos='http://cran.us.r-project.org')
install.packages("stmCorrViz", repos='http://cran.us.r-project.org')
install.packages("LDAvis", repos='http://cran.us.r-project.org')
install.packages("servr", repos='http://cran.us.r-project.org')
install.packages("lubridate", repos='http://cran.us.r-project.org')
install.packages("topicmodels", repos='http://cran.us.r-project.org')
```