Text Mining, pt. II

Philip D. Waggoner

MACS 40800: Unsupervised Machine Learning

November 21, 2019

Lecture Outline

1. Text Mining Revisited
2. Mixed Membership Models
3. Latent Dirichlet Allocation
4. Structural Topic Modeling
5. Some Code for Text Mining

Lecture Outline

Text Mining

- Today we return to an unsupervised framework for mining text

Text Mining

- Today we return to an unsupervised framework for mining text
- Our goal today?

Text Mining

- Today we return to an unsupervised framework for mining text
- Our goal today? Uncover structure in text data, which is usually considered some mixture of topics in a single document

Text Mining

- Today we return to an unsupervised framework for mining text
- Our goal today? Uncover structure in text data, which is usually considered some mixture of topics in a single document $\rightsquigarrow$ topic models

Text Mining

- Today we return to an unsupervised framework for mining text
- Our goal today? Uncover structure in text data, which is usually considered some mixture of topics in a single document $\rightsquigarrow$ topic models
- We will briefly touch on structural topic models given several projects interested in these

Lecture Outline

The Basics of Topic Models

- Topic modeling is a methods for grouping terms in a corpus into substantively meaningful categories, or "topics," based on some statistical correlations between frequency of words used together ("co-occurrence")

The Basics of Topic Models

- Topic modeling is a methods for grouping terms in a corpus into substantively meaningful categories, or "topics," based on some statistical correlations between frequency of words used together ("co-occurrence")

- It is unsupervised because we don't tell the algorithm the topics beforehand

The Basics of Topic Models

- Topic modeling is a methods for grouping terms in a corpus into substantively meaningful categories, or "topics," based on some statistical correlations between frequency of words used together ("co-occurrence")

- It is unsupervised because we don't tell the algorithm the topics beforehand

- Rather, the algorithm "discovers" abstract topics that can be thought of as a constellation of words that tend to show up together

The Basics of Topic Models

- Topic modeling is a methods for grouping terms in a corpus into substantively meaningful categories, or "topics," based on some statistical correlations between frequency of words used together ("co-occurrence")

- It is unsupervised because we don't tell the algorithm the topics beforehand

- Rather, the algorithm "discovers" abstract topics that can be thought of as a constellation of words that tend to show up together

- Topic modeling is distinct from clustering given the assumed nature of the **membership** of topics in a document: *mixed* membership vs. *single* membership

The Basics of Topic Models

- Suppose we had some set of documents on policymaking in Congress:

The Basics of Topic Models

- Suppose we had some set of documents on policymaking in Congress:

  Together, Republicans and Democrats can work toward a better future.

  The problem of polarization flows from a refusal of Republicans and Democrats to work together.

  Policy formation requires input from multiple stakeholders.

  Congressional committees should be required to subpoena stakeholders in related hearings.

  Republicans and Democrats don't seem to want to work together to find a solution to the policy gridlock crisis in Congress.

The Basics of Topic Models

- To uncover the topics, recall we are interested in *co-occurrence* of terms across documents

The Basics of Topic Models

- To uncover the topics, recall we are interested in *co-occurrence* of terms across documents

Together, Republicans and Democrats can work toward a better future.

The problem of polarization flows from a refusal of Republicans and Democrats to work together.

Policy formation requires input from multiple stakeholders.

Congressional committees should be required to subpoena stakeholders in related hearings.

Republicans and Democrats don't seem to want to work together to find a solution to the policy gridlock crisis in Congress.

The Basics of Topic Models

- So what is the goal of a topic model?

The Basics of Topic Models

- So what is the goal of a topic model? $\rightsquigarrow$ **method to derive topics in text based on co-occurrence of terms**

The Basics of Topic Models

- So what is the goal of a topic model? $\rightsquigarrow$ **method to derive topics in text based on co-occurrence of terms**

- A class of techniques for for discovering the broad themes that pervade a large and otherwise unstructured collection of documents

The Basics of Topic Models

- So what is the goal of a topic model? $\rightsquigarrow$ **method to derive topics in text based on co-occurrence of terms**

- A class of techniques for for discovering the broad themes that pervade a large and otherwise unstructured collection of documents

- Topic models can organize the documents, then, according to the discovered themes

The Basics of Topic Models

- So what is the goal of a topic model? $\rightsquigarrow$ **method to derive topics in text based on co-occurrence of terms**

- A class of techniques for for discovering the broad themes that pervade a large and otherwise unstructured collection of documents

- Topic models can organize the documents, then, according to the discovered themes $\rightsquigarrow$ **reducing complexity** of the (document) feature space

The Basics of Topic Models

- So what is the goal of a topic model? $\rightsquigarrow$ **method to derive topics in text based on co-occurrence of terms**
- A class of techniques for for discovering the broad themes that pervade a large and otherwise unstructured collection of documents
- Topic models can organize the documents, then, according to the discovered themes $\rightsquigarrow$ **reducing complexity** of the (document) feature space
- Note that in social science we often use the outputs from topic models to inform some measurement strategy, e.g.,

The Basics of Topic Models

- So what is the goal of a topic model? $\rightsquigarrow$ **method to derive topics in text based on co-occurrence of terms**

- A class of techniques for for discovering the broad themes that pervade a large and otherwise unstructured collection of documents

- Topic models can organize the documents, then, according to the discovered themes $\rightsquigarrow$ **reducing complexity** of the (document) feature space

- Note that in social science we often use the outputs from topic models to inform some measurement strategy, e.g.,
  - "who pays more attention to education, conservatives or liberals?"

Clustering or Topics?

Clustering

Document $\rightsquigarrow$ One Cluster

Doc 1

Doc 2

Doc 3

$\vdots$

Doc N

Topic 1

Topic 2

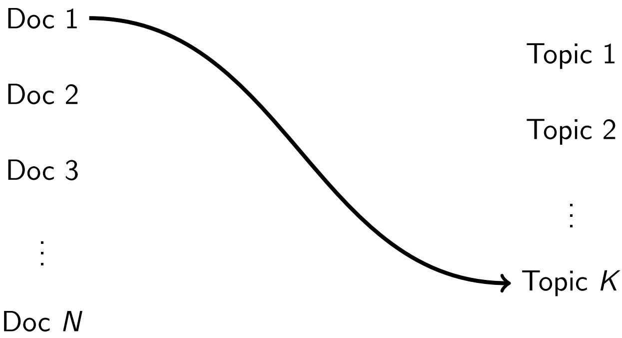
$\vdots$

Topic K

Clustering or Topics?

Clustering

Document $\rightsquigarrow$ One Cluster



Clustering or Topics?

Clustering

Document $\rightsquigarrow$ One Cluster

Doc 1	
	Topic 1
Doc 2	
	Topic 2
Doc 3	
	$\vdots$
$\vdots$	
	Topic K
Doc N	

Clustering or Topics?

Clustering
Document ⇝ One Cluster

Doc 1

Doc 2

Doc 3

⋮

Doc N

Topic 1

Topic 2

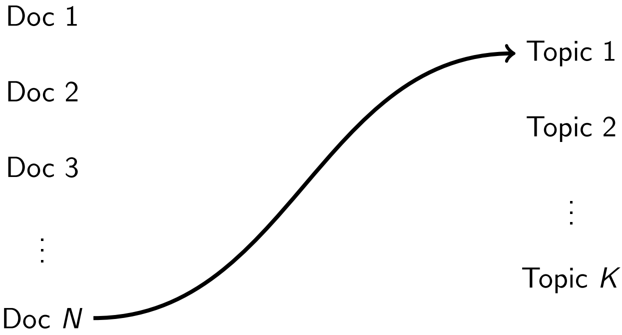
⋮

Topic K

Clustering or Topics?

Clustering

Document ⇝ One Cluster



Clustering or Topics?

Topic Models (Mixed Membership)

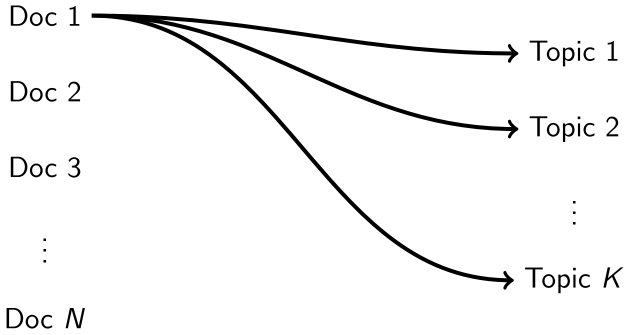
Document $\rightsquigarrow$ Many clusters

Doc 1	
	Topic 1
Doc 2	
	Topic 2
Doc 3	
	$\vdots$
$\vdots$	
	Topic K
Doc N	

Clustering or Topics?

Topic Models (Mixed Membership)

Document $\rightsquigarrow$ Many clusters



Data Generating Process (DGP)

- Importantly, in topic modeling, we assume there is some **unobserved** data generating process

Data Generating Process (DGP)

- Importantly, in topic modeling, we assume there is some **unobserved** data generating process
- Core assumption

Data Generating Process (DGP)

- Importantly, in topic modeling, we assume there is some **unobserved** data generating process
- Core assumption $\rightsquigarrow$ documents exhibit different topics, and in different proportions

Data Generating Process (DGP)

- Importantly, in topic modeling, we assume there is some **unobserved** data generating process
- Core assumption $\rightsquigarrow$ documents exhibit different topics, and in different proportions
  - e.g., A speech by Trump might be 50% drawn from the topic `IMMIGRATION`, 40% from the topic `AMERICA`, 9.9% from the topic `GREAT`, 0.1% from the topic `SECURITY`

Data Generating Process (DGP)

- A topic, then, is a distribution of terms over a fixed vocabulary, with some degree of probability

Data Generating Process (DGP)

- A topic, then, is a distribution of terms over a fixed vocabulary, with some degree of probability
  - The `IMMIGRATION` topic will have words like `wall` and `illegal` with high probabilities, and words like `Democrats` and `education` might have low probabilities

Data Generating Process (DGP)

- A topic, then, is a distribution of terms over a fixed vocabulary, with some degree of probability
  - The `IMMIGRATION` topic will have words like `wall` and `illegal` with high probabilities, and words like `Democrats` and `education` might have low probabilities
- Important: as we are trying to uncover **latent** structure, we are assuming the topics were actually generated (as a function of this DGP) **first**, and the documents then are generated from those topics

Data Generating Process (DGP)

- A topic, then, is a distribution of terms over a fixed vocabulary, with some degree of probability
  - The `IMMIGRATION` topic will have words like `wall` and `illegal` with high probabilities, and words like `Democrats` and `education` might have low probabilities
- Important: as we are trying to uncover **latent** structure, we are assuming the topics were actually generated (as a function of this DGP) **first**, and the documents then are generated from those topics
- So... where do the *words* in the documents come from?

Working Backwards to "Create" a Document: Generating Words

- Extending this imaginary world, we work backwards

Working Backwards to "Create" a Document: Generating Words

- Extending this imaginary world, we work backwards

- For each document:

Working Backwards to "Create" a Document: Generating Words

- Extending this imaginary world, we work backwards

- For each document:

    1. Randomly choose one of many multinomial distributions, each which mixes the topics in different proportions

Working Backwards to "Create" a Document: Generating Words

- Extending this imaginary world, we work backwards

- For each document:

  1. Randomly choose one of many multinomial distributions, each which mixes the topics in different proportions

  2. Then, for every word in the document:

Working Backwards to “Create” a Document: Generating Words

- Extending this imaginary world, we work backwards
- For each document:
  1. Randomly choose one of many multinomial distributions, each which mixes the topics in different proportions
  2. Then, for every word in the document:
     1. Randomly choose a topic from the distribution over topics from step 1

Working Backwards to “Create” a Document: Generating Words

- Extending this imaginary world, we work backwards
- For each document:
    1. Randomly choose one of many multinomial distributions, each which mixes the topics in different proportions
    2. Then, for every word in the document:
        1. Randomly choose a topic from the distribution over topics from step 1
        2. Randomly choose a word from the distribution over the vocabulary that the topic implies

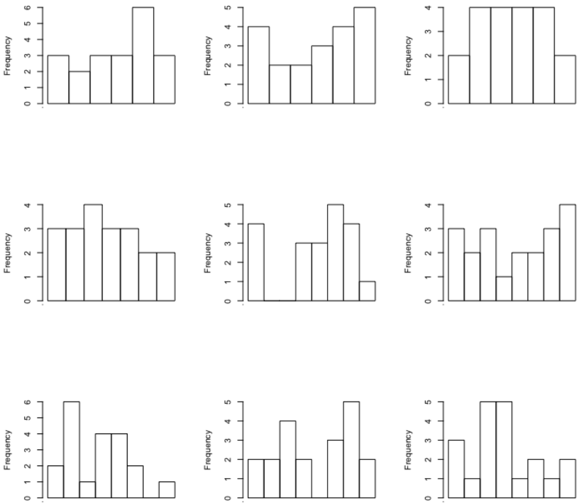
Working Backwards to "Create" a Document: Generating Words

- Extending this imaginary world, we work backwards
- For each document:
  1. Randomly choose one of many multinomial distributions, each which mixes the topics in different proportions
  2. Then, for every word in the document:
     1. Randomly choose a topic from the distribution over topics from step 1
     2. Randomly choose a word from the distribution over the vocabulary that the topic implies
- Aggregating across these steps for all words and all topics $\rightsquigarrow$ in the documents, which are the only things we actually observe

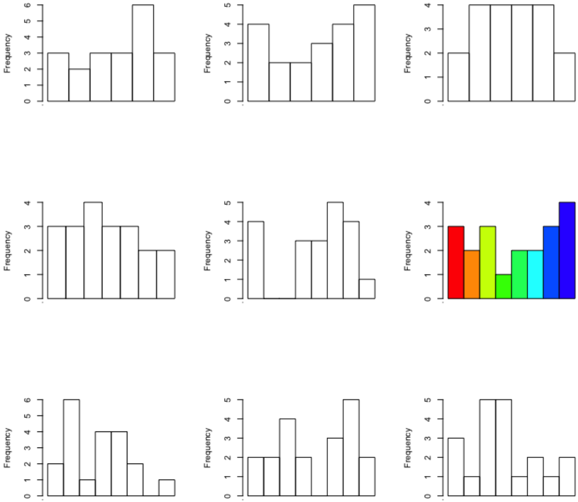
Generating Words: Step 1

- Randomly choose a distribution over topics
- That is, choose one of many multinomial distributions, each which mixes the topics in different proportions

Generating Words: Step 1



Generating Words: Step 1



Generating Words: Step 2

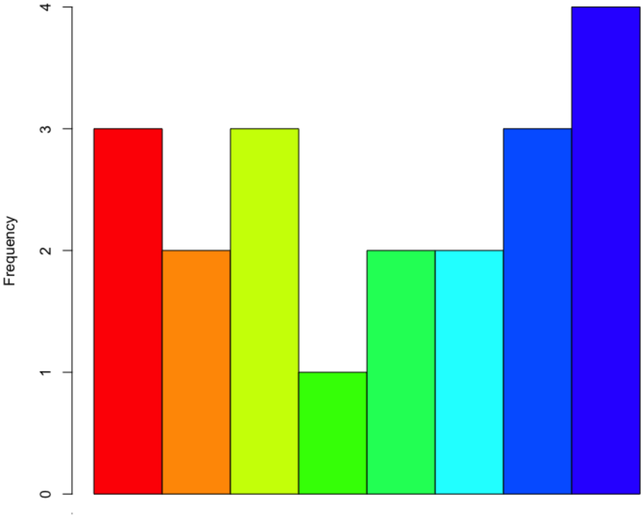
- Then, for every word in the document

Generating Words: Step 2

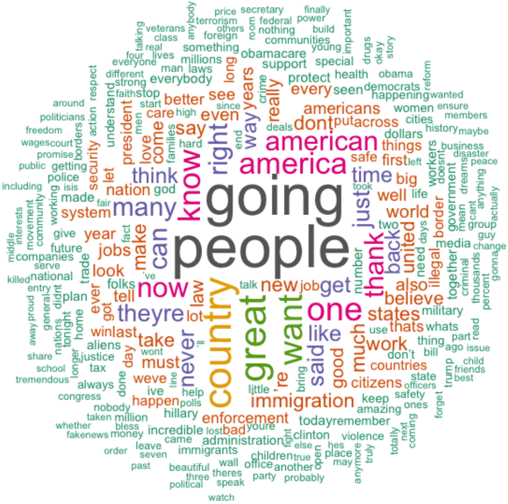
- Then, for every word in the document

  1. Randomly choose a **topic** from the distribution over **topics** from step 1
  2. Randomly choose a **word** from the distribution over the **vocabulary** that the topic implies

Generating Words: Step 2



Generating Words: Step 2



Generating Words: Step 2

`great`

Topic Definitions & Word Distributions

- Some of our variables – the documents which contain the words – are observable

Topic Definitions & Word Distributions

- Some of our variables – the documents which contain the words – are observable
- But, topic structure – topics, per-document topic distributions, per-document per-word topic assignments – are **latent**

Topic Definitions & Word Distributions

- Some of our variables – the documents which contain the words – are observable
- But, topic structure – topics, per-document topic distributions, per-document per-word topic assignments – are **latent**
- We need a distribution from which to draw the per-document topic distribution

Topic Definitions & Word Distributions

- Some of our variables – the documents which contain the words – are observable
- But, topic structure – topics, per-document topic distributions, per-document per-word topic assignments – are **latent**
- We need a distribution from which to draw the per-document topic distribution
- Most commonly, we use a **dirichlet** distribution: multiple categorical variables (mixture of multinomials), with shifting membership

Topic Definitions & Word Distributions

- Some of our variables – the documents which contain the words – are observable
- But, topic structure – topics, per-document topic distributions, per-document per-word topic assignments – are **latent**
- We need a distribution from which to draw the per-document topic distribution
- Most commonly, we use a **dirichlet** distribution: multiple categorical variables (mixture of multinomials), with shifting membership
- The Dirichlet process controls **allocation** of the words in the documents to different topics $\rightsquigarrow$ it is used as a prior over the distribution of words, which define the topics

Topic Definitions & Word Distributions

- Some of our variables – the documents which contain the words – are observable
- But, topic structure – topics, per-document topic distributions, per-document per-word topic assignments – are **latent**
- We need a distribution from which to draw the per-document topic distribution
- Most commonly, we use a **dirichlet** distribution: multiple categorical variables (mixture of multinomials), with shifting membership
- The Dirichlet process controls **allocation** of the words in the documents to different topics $\rightsquigarrow$ it is used as a prior over the distribution of words, which define the topics
- So what do we get...?

Topic Definitions & Word Distributions

- Some of our variables – the documents which contain the words – are observable
- But, topic structure – topics, per-document topic distributions, per-document per-word topic assignments – are **latent**
- We need a distribution from which to draw the per-document topic distribution
- Most commonly, we use a **dirichlet** distribution: multiple categorical variables (mixture of multinomials), with shifting membership
- The Dirichlet process controls **allocation** of the words in the documents to different topics $\rightsquigarrow$ it is used as a prior over the distribution of words, which define the topics
- So what do we get...?
- **latent Dirichlet allocation** $\rightsquigarrow$ specific type of probabilistic topic model controlling the assignment of words to topics

Topic Definitions & Word Distributions

- Some of our variables – the documents which contain the words – are observable
- But, topic structure – topics, per-document topic distributions, per-document per-word topic assignments – are **latent**
- We need a distribution from which to draw the per-document topic distribution
- Most commonly, we use a **dirichlet** distribution: multiple categorical variables (mixture of multinomials), with shifting membership
- The Dirichlet process controls **allocation** of the words in the documents to different topics $\rightsquigarrow$ it is used as a prior over the distribution of words, which define the topics
- So what do we get...?
- **latent Dirichlet allocation** $\rightsquigarrow$ specific type of probabilistic topic model controlling the assignment of words to topics
- In sum, we want to model the most likely-to-exist combined membership of words across all topics, in a probabilistic way

Lecture Outline

A General Process

1. Preprocess

A General Process

1. Preprocess
2. Select k topics to initialize

A General Process

1. Preprocess
2. Select k topics to initialize
3. Evaluate, rinse and repeat at different values of k until a “robust” set of topics is uncovered

A General Process

1. Preprocess
2. Select k topics to initialize
3. Evaluate, rinse and repeat at different values of k until a "robust" set of topics is uncovered
   - In most social science applications, the number of topics, k, is not picked automatically; a general approach to fit multiple models at multiple values of k and compare

A General Process

1. Preprocess
2. Select k topics to initialize
3. Evaluate, rinse and repeat at different values of k until a "robust" set of topics is uncovered
   - In most social science applications, the number of topics, k, is not picked automatically; a general approach to fit multiple models at multiple values of k and compare
   - As with all unsupervised learning, interpretation is non-trivial, and requires a lot of **thinking** and **validation**

Fitting an LDA Topic Model

- LDA is a generative model

Fitting an LDA Topic Model

- LDA is a generative model $\rightsquigarrow$ defines a DGP for each document and then uses the data to find the most likely values for the parameters within the model

Fitting an LDA Topic Model

- LDA is a generative model $\rightsquigarrow$ defines a DGP for each document and then uses the data to find the most likely values for the parameters within the model
- The result is a set of topics made up of words that frequently (conditionally) appear together, and most likely to belong to a similar topic

Fitting an LDA Topic Model

- LDA is a generative model $\rightsquigarrow$ defines a DGP for each document and then uses the data to find the most likely values for the parameters within the model
- The result is a set of topics made up of words that frequently (conditionally) appear together, and most likely to belong to a similar topic
- Note all words have *some probability of belonging to each topic*

Fitting an LDA Topic Model

- LDA is a generative model $\rightsquigarrow$ defines a DGP for each document and then uses the data to find the most likely values for the parameters within the model
- The result is a set of topics made up of words that frequently (conditionally) appear together, and most likely to belong to a similar topic
- Note all words have *some probability of belonging to each topic*
- We use the observed data (all tokens in our corpus) to make some inference about the latent parameters (β's and θ's)

Fitting an LDA Topic Model

- LDA is a generative model $\rightsquigarrow$ defines a DGP for each document and then uses the data to find the most likely values for the parameters within the model
- The result is a set of topics made up of words that frequently (conditionally) appear together, and most likely to belong to a similar topic
- Note all words have *some probability of belonging to each topic*
- We use the observed data (all tokens in our corpus) to make some inference about the latent parameters (β's and θ's)
- The parameters captures the conditional probabilities that some sequence of words belong to a given topic based on co-occurrence throughout the document

Fitting an LDA Topic Model

- LDA is a generative model $\rightsquigarrow$ defines a DGP for each document and then uses the data to find the most likely values for the parameters within the model
- The result is a set of topics made up of words that frequently (conditionally) appear together, and most likely to belong to a similar topic
- Note all words have *some probability of belonging to each topic*
- We use the observed data (all tokens in our corpus) to make some inference about the latent parameters (β's and θ's)
- The parameters captures the conditional probabilities that some sequence of words belong to a given topic based on co-occurrence throughout the document
- The sum of the topic proportions across all topics for each document is one, and the sum of the word probabilities for each topic is one

Output

- For user-selected k topics, a typical implementation of LDA will return

Output

- For user-selected k topics, a typical implementation of LDA will return
  - The word distribution for each topic, β (e.g., the proportion of each word in each topic)

Output

- For user-selected k topics, a typical implementation of LDA will return
  - The word distribution for each topic, β (e.g., the proportion of each word in each topic)
  - The topic distribution for each document, θ (e.g., the proportion of all topics, k in each document)

Selecting k?

1. In social science, researchers fit topic models until they see what they think they should
   - e.g., a certain topic like `IMMIGRATION` consistently (or at least *suddenly*) appears, so stop there

Selecting k?

1. In social science, researchers fit topic models until they see what they think they should
   - e.g., a certain topic like `IMMIGRATION` consistently (or at least *suddenly*) appears, so stop there
2. Its best practice, at a minimum, to check that findings are robust in some neighborhood
   - e.g., if best model likely has $k = 15$, check whether $k = 10 - 20$ yield similar patterns, and thus inferences

A More Principled Approach to Selecting k?

1. Split texts randomly into training and testing sets (typically 80/20)

A More Principled Approach to Selecting k?

1. Split texts randomly into training and testing sets (typically 80/20)
2. For the training set, pick some value of k and fit a topic model

A More Principled Approach to Selecting k?

1. Split texts randomly into training and testing sets (typically 80/20)
2. For the training set, pick some value of k and fit a topic model
3. Record parameter values on a document for a specific topic distribution (θ), and the word distributions for the topics (β)
4. Find the highest log-likelihood across many specifications of a similar distribution of words over topics, and topics over documents,

$$\mathcal{L}(\mathbf{w}) = \log pr(\mathbf{w}|\beta, \theta) = \sum_{d} \log pr(w_d|\beta, \theta),$$

where $\mathbf{w}$ are the words in the test set

A More Principled Approach to Selecting k?

1. Split texts randomly into training and testing sets (typically 80/20)
2. For the training set, pick some value of k and fit a topic model
3. Record parameter values on a document for a specific topic distribution (θ), and the word distributions for the topics (β)
4. Find the highest log-likelihood across many specifications of a similar distribution of words over topics, and topics over documents,

$$\mathcal{L}(\mathbf{w}) = \log pr(\mathbf{w}|\beta, \theta) = \sum_{d} \log pr(w_d|\beta, \theta),$$

where $\mathbf{w}$ are the words in the test set

- Highest $\mathcal{L}(\mathbf{w})$ means the best model

A More Principled Approach to Selecting k?

1. Split texts randomly into training and testing sets (typically 80/20)
2. For the training set, pick some value of k and fit a topic model
3. Record parameter values on a document for a specific topic distribution (θ), and the word distributions for the topics (β)
4. Find the highest log-likelihood across many specifications of a similar distribution of words over topics, and topics over documents,

$$\mathcal{L}(\mathbf{w}) = \log pr(\mathbf{w}|\beta, \theta) = \sum_{d} \log pr(w_d|\beta, \theta),$$

   where **w** are the words in the test set

- Highest $\mathcal{L}(\mathbf{w})$ means the best model
- The intuition is to calculate the likelihood of seeing the test words, given what we know produced the training set

A More Principled Approach to Selecting k?

- Several different values for k may be *substantively* plausible, but by increasing k, we sacrifice clarity

A More Principled Approach to Selecting k?

- Several different values for k may be *substantively* plausible, but by increasing k, we sacrifice clarity
- Thus, $\mathcal{L}(\mathbf{w})$ is also used to calculate **perplexity**,

$$\textbf{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{N_{tokens}}\right)$$

A More Principled Approach to Selecting k?

- Several different values for k may be *substantively* plausible, but by increasing k, we sacrifice clarity
- Thus, $\mathcal{L}(\mathbf{w})$ is also used to calculate **perplexity**,

$$\textbf{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{N_{tokens}}\right)$$

- Perplexity is a measure of how well a model predicts a sample

A More Principled Approach to Selecting k?

- Several different values for k may be *substantively* plausible, but by increasing k, we sacrifice clarity
- Thus, $\mathcal{L}(\mathbf{w})$ is also used to calculate **perplexity**,

$$\textbf{perplexity} = \exp(-\frac{\mathcal{L}(\mathbf{w})}{N_{tokens}})$$

- Perplexity is a measure of how well a model predicts a sample
- So here, we are calculating how likely the test set is given the model on which we trained

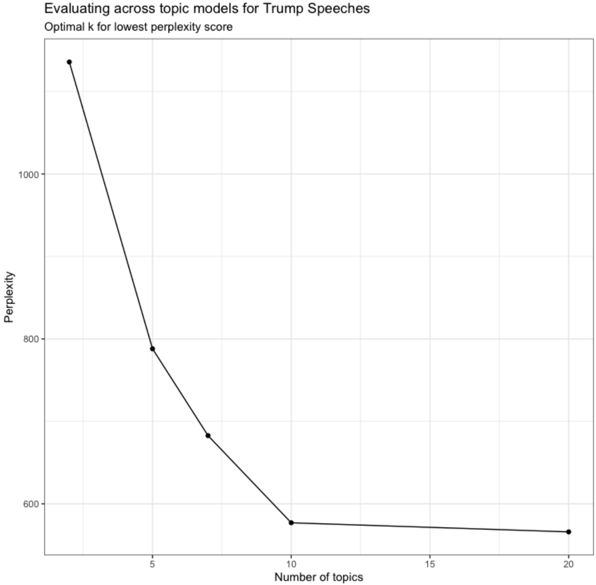
A More Principled Approach to Selecting k?

- Several different values for k may be *substantively* plausible, but by increasing k, we sacrifice clarity
- Thus, $\mathcal{L}(\mathbf{w})$ is also used to calculate **perplexity**,

$$\textbf{perplexity} = \exp(-\frac{\mathcal{L}(\mathbf{w})}{N_{tokens}})$$

- Perplexity is a measure of how well a model predicts a sample
- So here, we are calculating how likely the test set is given the model on which we trained
- (*hint*: `topicmodels` includes a function `perplexity()` which calculates this value for a model

Selecting k?


Evaluating across topic models for Trump Speeches
Optimal k for lowest perplexity score
Perplexity
1000
800
600
5
10
15
20
Number of topics

Lecture Outline

1. Text Mining Revisited
2. Mixed Membership Models
3. Latent Dirichlet Allocation
4. Structural Topic Modeling
5. Some Code for Text Mining

Structural Topic Modeling

- Usually, we have lots of metadata: e.g. author information, publication source, etc.

Structural Topic Modeling

- Usually, we have lots of metadata: e.g. author information, publication source, etc.
- But this may be non-trivial to include: STM $=$ LDA $+$ contextual information

Structural Topic Modeling

- Usually, we have lots of metadata: e.g. author information, publication source, etc.
- But this may be non-trivial to include: STM = LDA + contextual information
- STM, then, allows more precise estimation and usually more interpretable results (and essentially allows for a NHST framework)

Structural Topic Modeling

- Usually, we have lots of metadata: e.g. author information, publication source, etc.
- But this may be non-trivial to include: STM $=$ LDA $+$ contextual information
- STM, then, allows more precise estimation and usually more interpretable results (and essentially allows for a NHST framework)
- In brief, STMs model the topic distribution as a function of the document metadata

Structural Topic Modeling

- STMs too are generative models with document-topic and topic-word distributions assumed to be generating documents

Structural Topic Modeling

- STMs too are generative models with document-topic and topic-word distributions assumed to be generating documents
- But in the STM world, the documents have metadata associated with them (usually denoted as X_d, where d indexes the documents)

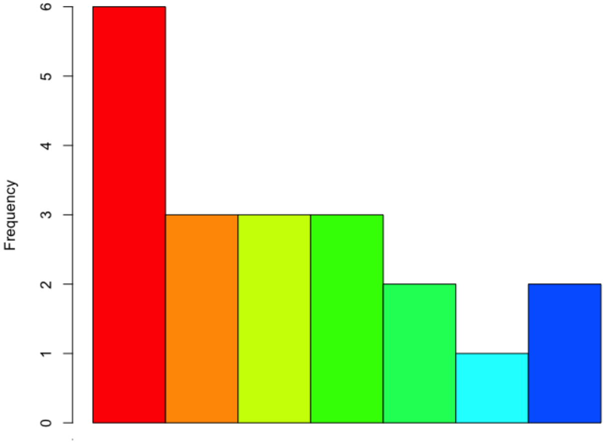
Structural Topic Modeling

- STMs too are generative models with document-topic and topic-word distributions assumed to be generating documents
- But in the STM world, the documents have metadata associated with them (usually denoted as X_d, where d indexes the documents)
- Then, just like LDA, a topic is defined as a *mixture* over words where each word has a probability of belonging to a topic

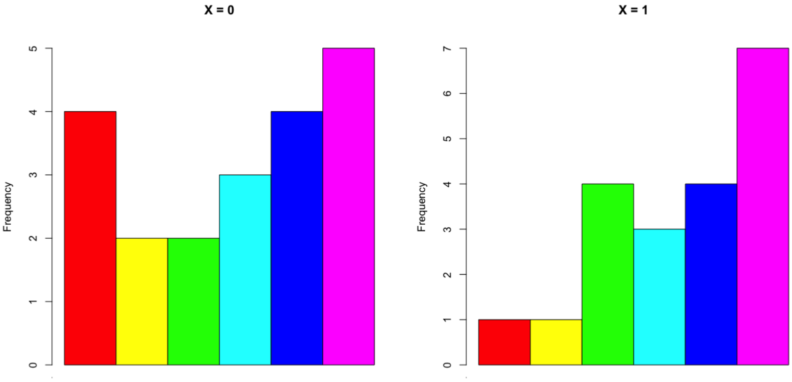
Structural Topic Modeling

- STMs too are generative models with document-topic and topic-word distributions assumed to be generating documents
- But in the STM world, the documents have metadata associated with them (usually denoted as X_d, where d indexes the documents)
- Then, just like LDA, a topic is defined as a *mixture* over words where each word has a probability of belonging to a topic
- And a document is a mixture over topics, where a single document can be composed of multiple topics

Topic Distribution over Documents (θ): LDA



Topic Distribution over Documents (θ): STM



Word Distribution per Topic (β): LDA

Word Distribution per Topic (β): STM

Figure: $X = 0$

Figure: $X = 1$

Lecture Outline