

University of Tokyo: Text-as-Data

Day 2, Part II

Arthur Spirling

June 4, 2017

Where Are We?

Where Are We?



Where Are We?



We've covered **supervised learning**: the situation in which we have labeled data.

Where Are We?



We've covered **supervised learning**: the situation in which we have labeled data.

Now begin to think about documents that are **not labelled** but within which we expect some important, latent **structure**.

Where Are We?



We've covered **supervised learning**: the situation in which we have labeled data.

Now begin to think about documents that are **not labelled** but within which we expect some important, latent **structure**.

cover some fundamental techniques for accessing that structure

Where Are We?



We've covered **supervised learning**: the situation in which we have labeled data.

Now begin to think about documents that are **not labelled** but within which we expect some important, latent **structure**.

cover some fundamental techniques for accessing that structure

and demonstrate challenges that emerge in interpreting the results.

Terminology

Terminology

Unsupervised techniques:

Terminology

Unsupervised techniques: learning
(hidden or latent) structure in
unlabeled data.

e.g. PCA of legislators's votes:

Terminology

Unsupervised techniques: learning
(hidden or latent) structure in
unlabeled data.

e.g. PCA of legislators's votes: want to see
how they are organized—

Terminology

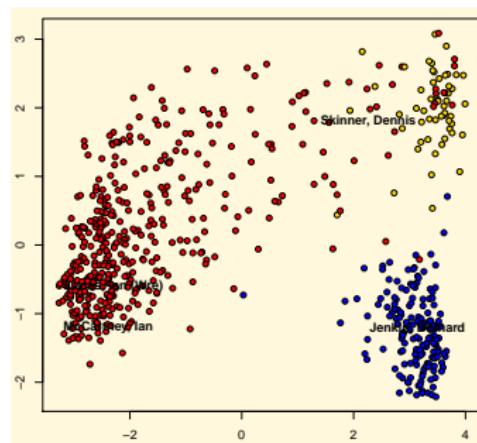
Unsupervised techniques: learning
(hidden or latent) structure in
unlabeled data.

e.g. PCA of legislators's votes: want to see
how they are organized—by party? by
ideology? by race?

Terminology

Unsupervised techniques: learning
(hidden or latent) structure in
unlabeled data.

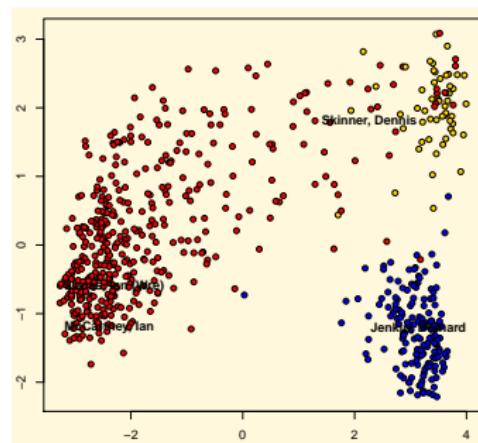
e.g. PCA of legislators's votes: want to see
how they are organized—by party? by
ideology? by race?



Terminology

Unsupervised techniques: learning (hidden or latent) structure in unlabeled data.

e.g. PCA of legislators's votes: want to see how they are organized—by party? by ideology? by race?

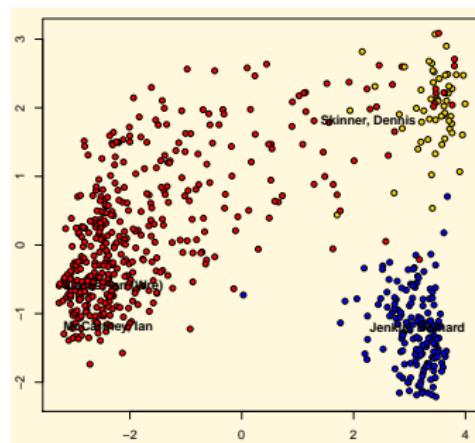


Supervised techniques:

Terminology

Unsupervised techniques: learning (hidden or latent) structure in unlabeled data.

e.g. PCA of legislators's votes: want to see how they are organized—by party? by ideology? by race?

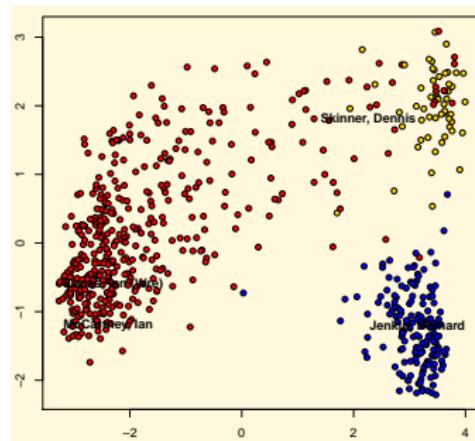


Supervised techniques: learning relationship between inputs and a labeled set of outputs.

Terminology

Unsupervised techniques: learning (hidden or latent) structure in unlabeled data.

e.g. PCA of legislators's votes: want to see how they are organized—by party? by ideology? by race?



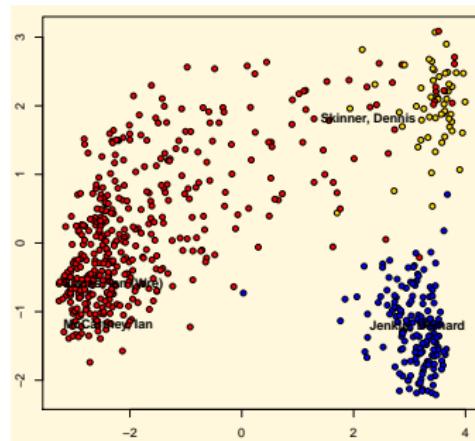
Supervised techniques: learning relationship between inputs and a labeled set of outputs.

e.g. opinion mining:

Terminology

Unsupervised techniques: learning (hidden or latent) structure in unlabeled data.

e.g. PCA of legislators's votes: want to see how they are organized—by party? by ideology? by race?



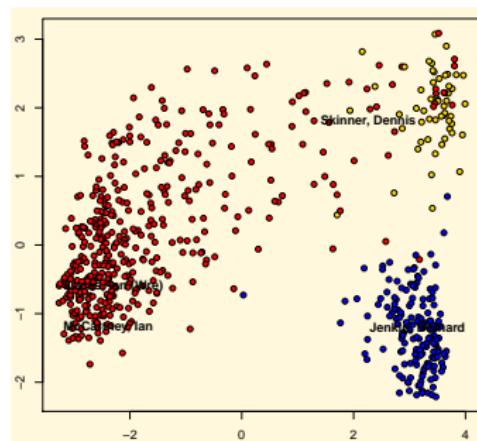
Supervised techniques: learning relationship between inputs and a labeled set of outputs.

e.g. opinion mining: what makes a critic like or dislike a movie ($y \in \{0, 1\}$)?

Terminology

Unsupervised techniques: learning (hidden or latent) structure in unlabeled data.

e.g. PCA of legislators's votes: want to see how they are organized—by party? by ideology? by race?



Supervised techniques: learning relationship between inputs and a labeled set of outputs.

e.g. opinion mining: what makes a critic like or dislike a movie ($y \in \{0, 1\}$)?

CRITIC REVIEWS FOR STAR WARS: EPISODE VII - THE FORCE AWAKENS

All Critics (313) | Top Critics (48) | My Critics | Fresh (293) | Rotten (20)

The new movie, as an act of pure storytelling, streams by with fluency and zip.
[Full Review...](#) | December 21, 2015

Anthony Lane
New Yorker
★ Top Critic

While Star Wars: The Force Awakens gets temporarily bogged down taking us back to the world that we left in 1983, it introduces us to the new and exciting torch-bearers of the franchise.
[Full Review...](#) | December 30, 2015

Blake Howard
Graffiti With Punctuation

At the end The Force Awakens looks more like a nostalgic film that will work as a transition to the new Star Wars' age. [Full Review in Spanish]
[Full Review...](#) | December 29, 2015

Salvador Franco Reyes

This film is a well-planned product that balances nostalgia with the capacity to attract new generations into the Star Wars universe. [Full Review in Spanish]
[Full Review...](#) | December 29, 2015

Overview: Unsupervised Learning

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

e.g. while we know who gave a particular speech, we don't yet know what that speech 'represents' in terms of its latent properties,

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

e.g. while we know who gave a particular speech, we don't yet know what that speech 'represents' in terms of its latent properties, what 'kind' of speech it is,

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

e.g. while we know who gave a particular speech, we don't yet know what that speech 'represents' in terms of its latent properties, what 'kind' of speech it is, what 'topics' it covers,

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

e.g. while we know who gave a particular speech, we don't yet know what that speech 'represents' in terms of its latent properties, what 'kind' of speech it is, what 'topics' it covers, what speeches it is similar to conceptually, etc.

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

e.g. while we know who gave a particular speech, we don't yet know what that speech 'represents' in terms of its latent properties, what 'kind' of speech it is, what 'topics' it covers, what speeches it is similar to conceptually, etc.

Goal is to take the observations and find hidden structure and meaning in them.

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

e.g. while we know who gave a particular speech, we don't yet know what that speech 'represents' in terms of its latent properties, what 'kind' of speech it is, what 'topics' it covers, what speeches it is similar to conceptually, etc.

Goal is to take the observations and find hidden structure and meaning in them.

→ look for (dis)similarities between documents (or observations):

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

e.g. while we know who gave a particular speech, we don't yet know what that speech 'represents' in terms of its latent properties, what 'kind' of speech it is, what 'topics' it covers, what speeches it is similar to conceptually, etc.

Goal is to take the observations and find hidden structure and meaning in them.

→ look for (dis)similarities between documents (or observations): it will be up to us to interpret what the groups/dimensions/concepts represent after the technique has been used.

Overview: Unsupervised Learning

Now our data—humans, documents, observations—are not pre-labeled in terms of some underlying concept.

e.g. while we know who gave a particular speech, we don't yet know what that speech 'represents' in terms of its latent properties, what 'kind' of speech it is, what 'topics' it covers, what speeches it is similar to conceptually, etc.

Goal is to take the observations and find hidden structure and meaning in them.

→ look for (dis)similarities between documents (or observations): it will be up to us to interpret what the groups/dimensions/concepts represent after the technique has been used.

So...

So...

in contrast to **supervised** approaches,

So...

in contrast to **supervised** approaches, we won't know 'how correct' the output is in a simple statistical sense

So...

in contrast to **supervised** approaches, we won't know 'how correct' the output is in a simple statistical sense

While there *are* ways to compare unsupervised models, we are generally judging utility/fit in a different way

So...

in contrast to **supervised** approaches, we won't know 'how correct' the output is in a simple statistical sense

While there *are* ways to compare unsupervised models, we are generally judging utility/fit in a different way

So, does it tell us something interesting/plausible?

So...

in contrast to **supervised** approaches, we won't know 'how correct' the output is in a simple statistical sense

While there *are* ways to compare unsupervised models, we are generally judging utility/fit in a different way

So, does it tell us something interesting/plausible? do somewhat similar (e.g. 2, 3, 4 clusters) specifications imply the same thing?

So...

in contrast to **supervised** approaches, we won't know 'how correct' the output is in a simple statistical sense

While there *are* ways to compare unsupervised models, we are generally judging utility/fit in a different way

So, does it tell us something interesting/plausible? do somewhat similar (e.g. 2, 3, 4 clusters) specifications imply the same thing?

(**not** "what is the recall/precision/accuracy?")

Basic Techniques: Principal Components and Clustering

Motivating Problem

Motivating Problem

Have an $n \times p$ matrix,

Motivating Problem

Have an $n \times p$ matrix, want to summarize/analyze:

Motivating Problem

Have an $n \times p$ matrix, want to summarize/analyze: could be DTM.

Motivating Problem

Have an $n \times p$ matrix, want to summarize/analyze: could be DTM.

e.g. Political science: n legislators, p roll calls of interest, $n > p$

Motivating Problem

Have an $n \times p$ matrix, want to summarize/analyze: could be DTM.

e.g. Political science: n legislators, p roll calls of interest, $n > p$

Motivating Problem

Have an $n \times p$ matrix, want to summarize/analyze: could be DTM.

e.g. Political science: n legislators, p roll calls of interest, $n > p$

Name	Party	Vote 1	Vote 2	Vote 3	
Ainsworth, Peter (E S)	Con	NA	1	NA	...
Alexander, Douglas	Lab	NA	0	0	...
Allan, Richard	LD	1	0	1	...
Allen, Graham	Lab	0	0	0	...
Amess, David	Con	1	1	NA	...
	

Motivating Problem

Have an $n \times p$ matrix, want to summarize/analyze: could be DTM.

Motivating Problem

Have an $n \times p$ matrix, want to summarize/analyze: could be DTM.

- e.g. Text: n speakers, p features in the speeches (often $p > n$ for text problems)

Motivating Problem

Have an $n \times p$ matrix, want to summarize/analyze: could be DTM.

e.g. Text: n speakers, p features in the speeches (often $p > n$ for text problems)

Name	Party	'cost'	'spend'	'tax'	...
Ainsworth, Peter (E S)	Con	0.00	0.01	0.30	...
Alexander, Douglas	Lab	0.32	0.20	0.86	...
Allan, Richard	LD	0.99	0.82	0.61	...
Allen, Graham	Lab	0.52	0.86	0.34	...
Amess, David	Con	0.07	0.34	0.33	...
	

PCA: Introduction

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

Very popular for data summary, exploration (and analysis?)

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

Very popular for data summary, exploration (and analysis?)

Aims:

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

Very popular for data summary, exploration (and analysis?)

Aims:

- extract core/important information from data

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

Very popular for data summary, exploration (and analysis?)

Aims:

- extract core/important information from data
- reduce the data/problem down to this information

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

Very popular for data summary, exploration (and analysis?)

Aims:

- extract core/important information from data
- reduce the data/problem down to this information
- simplify data

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

Very popular for data summary, exploration (and analysis?)

Aims:

- extract core/important information from data
- reduce the data/problem down to this information
- simplify data
- analyze data in terms of its patterns/groups

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

Very popular for data summary, exploration (and analysis?)

Aims:

- extract core/important information from data
- reduce the data/problem down to this information
- simplify data
- analyze data in terms of its patterns/groups

Generally: represent this information as new (and smaller number of) variables known as *principal components*

PCA: Introduction

Possibly oldest multivariate technique (Pearson, 1901?)

Very popular for data summary, exploration (and analysis?)

Aims:

- extract core/important information from data
- reduce the data/problem down to this information
- simplify data
- analyze data in terms of its patterns/groups

Generally: represent this information as new (and smaller number of) variables known as *principal components*

Overview

Overview

Features: these principal components will be uncorrelated (orthogonal) with (to) each other,

Overview

Features: these principal components will be uncorrelated (orthogonal) with (to) each other, will be **linear combinations** of original variables

Overview

Features: these principal components will be uncorrelated (orthogonal) with (to) each other, will be **linear combinations** of original variables

Result: lower dimensional ‘map’ of observations in new space:

Overview

Features: these principal components will be uncorrelated (orthogonal) with (to) each other, will be **linear combinations** of original variables

Result: lower dimensional ‘map’ of observations in new space:

- each **observation** now has a value on each principal component called its **(factor) score**,

Overview

Features: these principal components will be uncorrelated (orthogonal) with (to) each other, will be **linear combinations** of original variables

Result: lower dimensional ‘map’ of observations in new space:

- each **observation** now has a value on each principal component called its **(factor) score**, which are **projections** of (original) observations onto the PCs

Overview

Features: these principal components will be uncorrelated (orthogonal) with (to) each other, will be **linear combinations** of original variables

Result: lower dimensional ‘map’ of observations in new space:

- each **observation** now has a value on each principal component called its **(factor) score**, which are **projections** of (original) observations onto the PCs

Interpretation of given PC: depends on correlation between component and (original) variable—known as **loading**

Overview

Features: these principal components will be uncorrelated (orthogonal) with (to) each other, will be **linear combinations** of original variables

Result: lower dimensional ‘map’ of observations in new space:

- each **observation** now has a value on each principal component called its **(factor) score**, which are **projections** of (original) observations onto the PCs

Interpretation of given PC: depends on correlation between component and (original) variable—known as **loading**

Method: (eigen-) **decomposition** of cov matrix or **singular value decomposition** of data matrix

Overview

Features: these principal components will be uncorrelated (orthogonal) with (to) each other, will be **linear combinations** of original variables

Result: lower dimensional ‘map’ of observations in new space:

- each **observation** now has a value on each principal component called its **(factor) score**, which are **projections** of (original) observations onto the PCs

Interpretation of given PC: depends on correlation between component and (original) variable—known as **loading**

Method: (eigen-) **decomposition** of cov matrix or **singular value decomposition** of data matrix

Method

Method

PCA performs a **linear transformation**

Method

PCA performs a **linear transformation** on the original variables into new coordinate system,

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points:

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points: but,

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points: but, this is in terms of distance **perpendicular** (orthogonal) to line,

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points: but, this is in terms of distance **perpendicular** (orthogonal) to line, not in terms of y -distance

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points: but, this is in terms of distance **perpendicular** (orthogonal) to line, not in terms of y -distance (cf **OLS**)

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points: but, this is in terms of distance **perpendicular** (orthogonal) to line, not in terms of y -distance (cf **OLS**)

All subsequent components captures (sequentially) less variability

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points: but, this is in terms of distance **perpendicular** (orthogonal) to line, not in terms of y -distance (cf **OLS**)

All subsequent components captures (sequentially) less variability

Assumptions: observations are independent

Method

PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points: but, this is in terms of distance **perpendicular** (orthogonal) to line, not in terms of y -distance (cf **OLS**)

All subsequent components captures (sequentially) less variability

Assumptions: observations are independent and X is p -variate normal (may not find highest variance projection if not)

Method

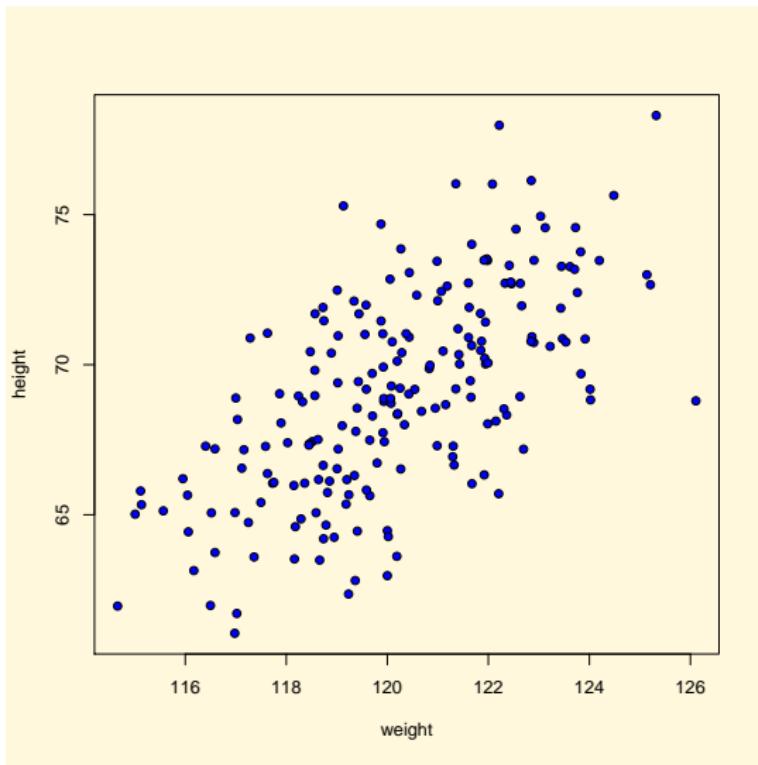
PCA performs a **linear transformation** on the original variables into new coordinate system, such that the first coordinate (first principal component) is the projection of the original data that contains the **most information** about that data

Can think of the first PC as being a line which **most closely fits** the data points: but, this is in terms of distance **perpendicular** (orthogonal) to line, not in terms of y -distance (cf **OLS**)

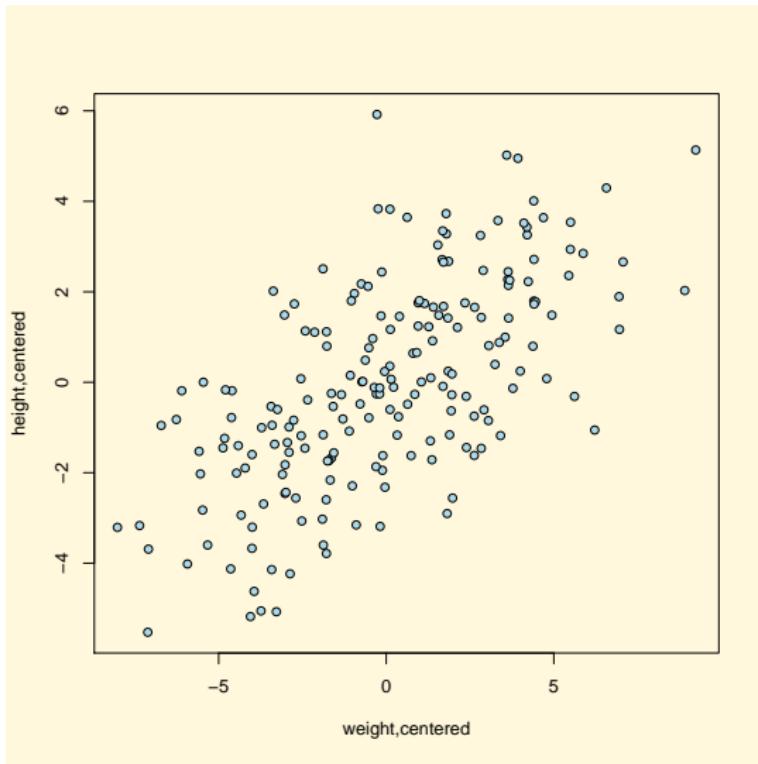
All subsequent components captures (sequentially) less variability

Assumptions: observations are independent and X is p -variate normal (may not find highest variance projection if not)

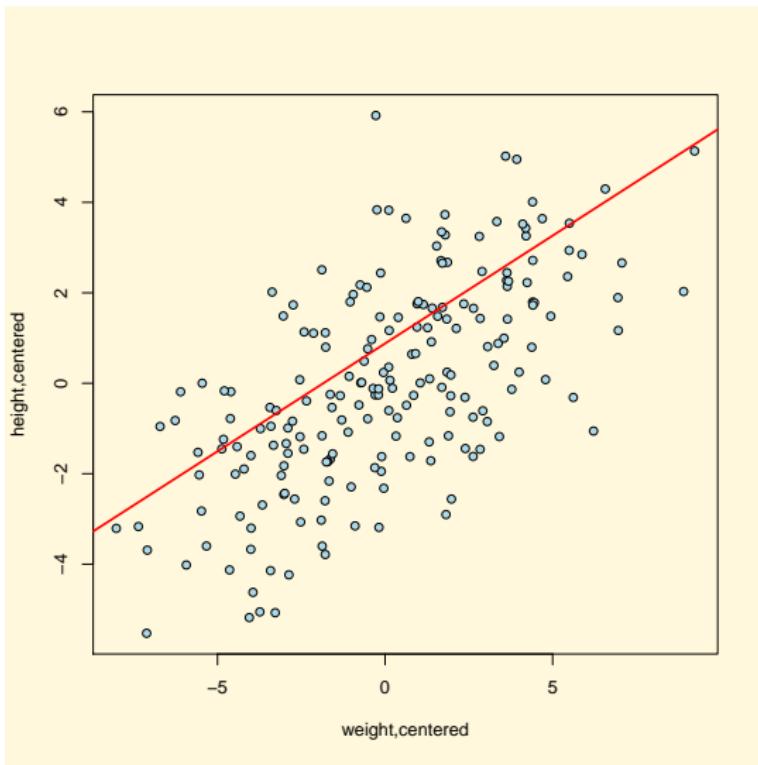
Heights, Weights



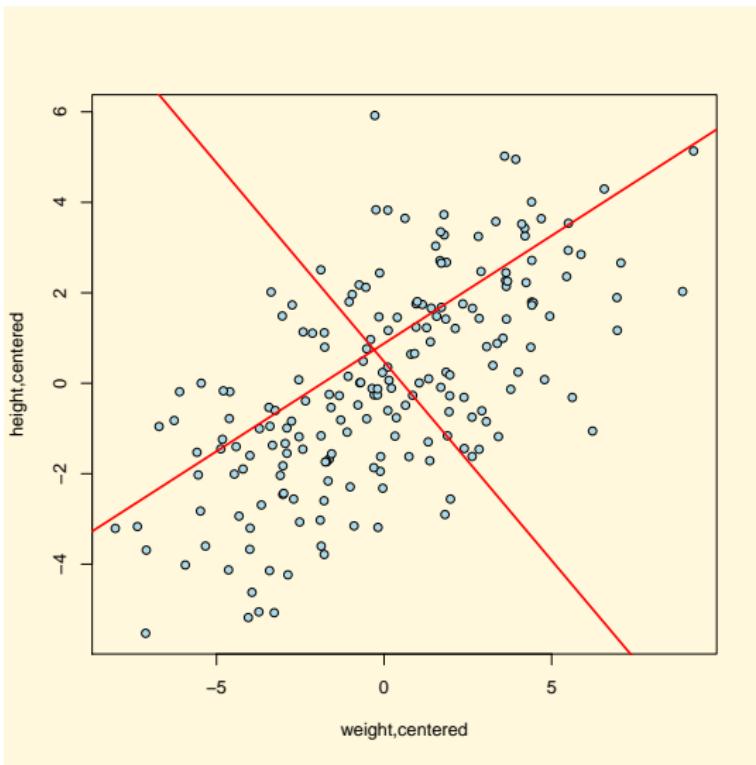
Heights, Weights



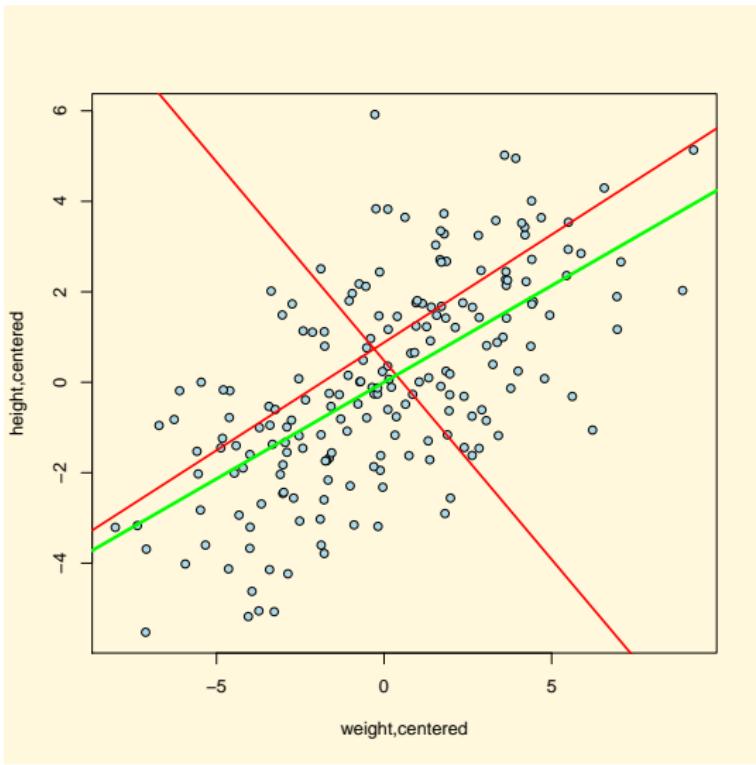
Heights, Weights



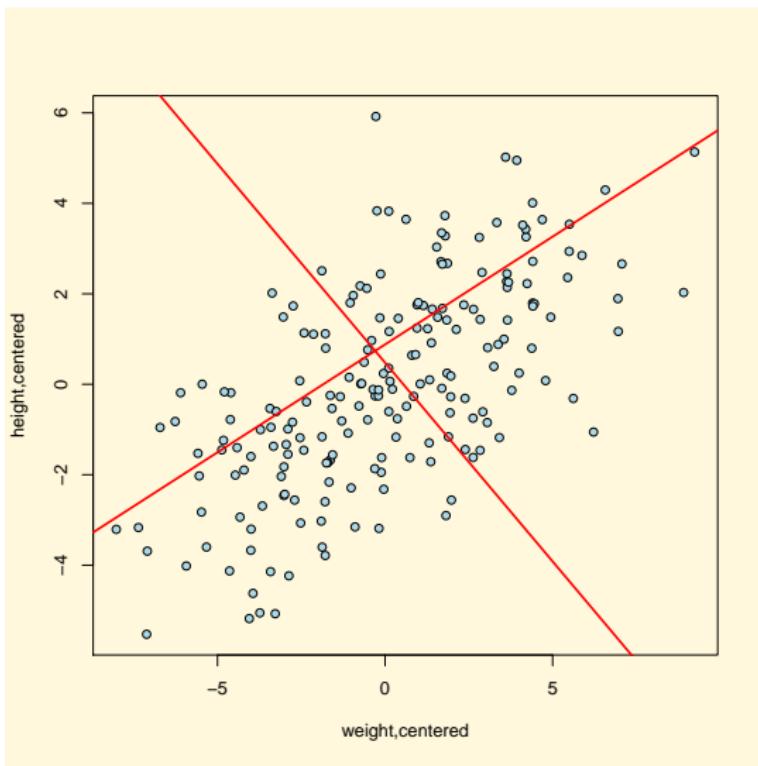
Heights, Weights



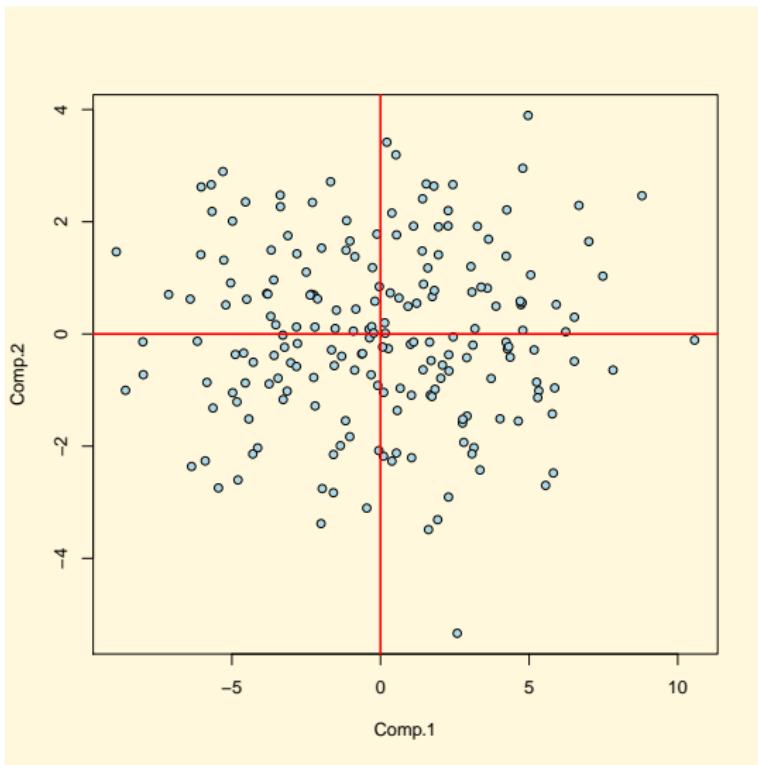
Heights, Weights



In new space



In new space



Political Speech: US Senate

Political Speech: US Senate

Beauchamp, 2010 (Text-Based
Scaling of Legislatures: A
Comparison of Methods with
Applications to the US Senate and
UK House of Commons)

Political Speech: US Senate

Beauchamp, 2010 (Text-Based
Scaling of Legislatures: A
Comparison of Methods with
Applications to the US Senate and
UK House of Commons)

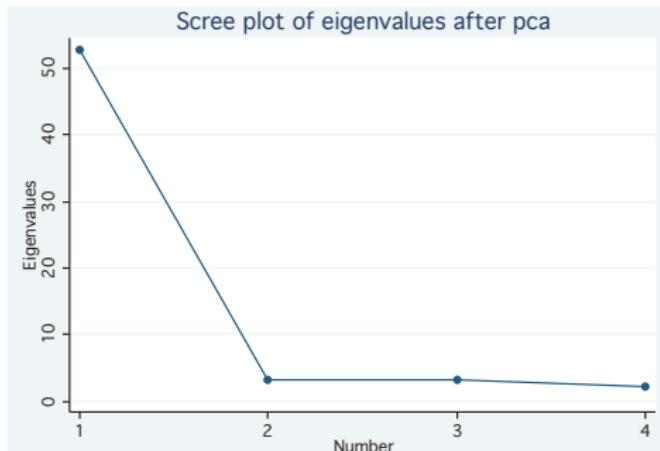
Considers PCA of (pre-processed)
1000-top-vectors for US Senators.

Political Speech: US Senate

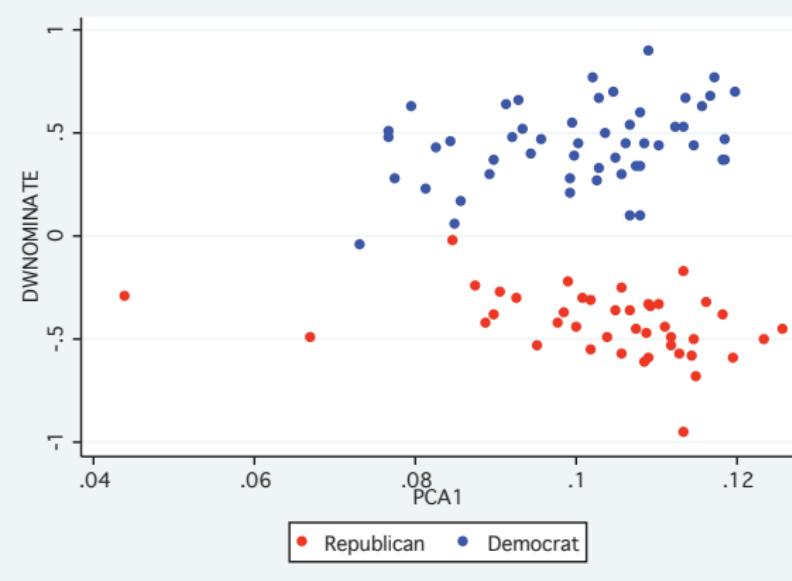
Beauchamp, 2010 (Text-Based
Scaling of Legislatures: A
Comparison of Methods with
Applications to the US Senate and
UK House of Commons)

Considers PCA of (pre-processed)
1000-top-vectors for US Senators.

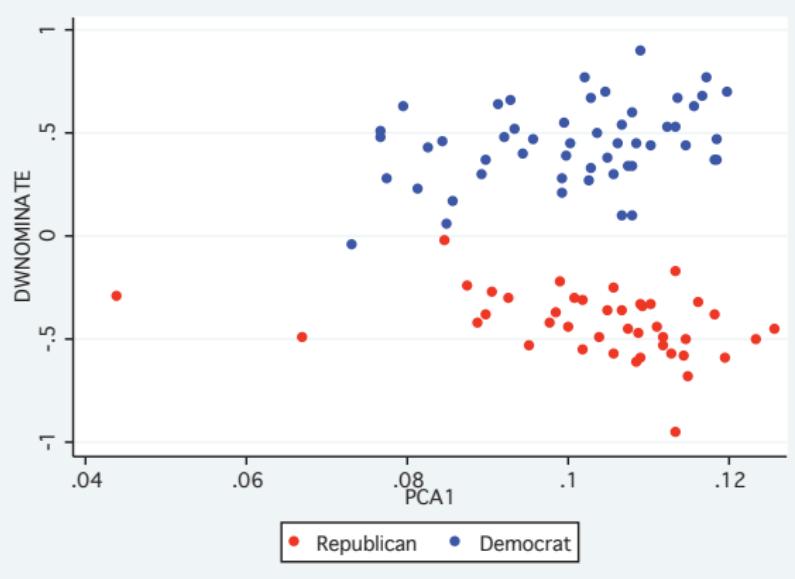
Fits several components, of which
1PC model looks very good...



Partner Exercise

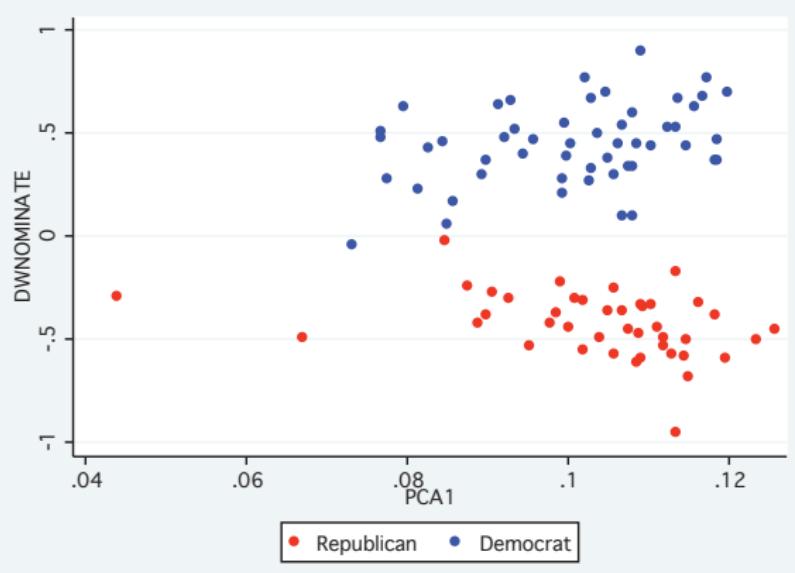


Partner Exercise



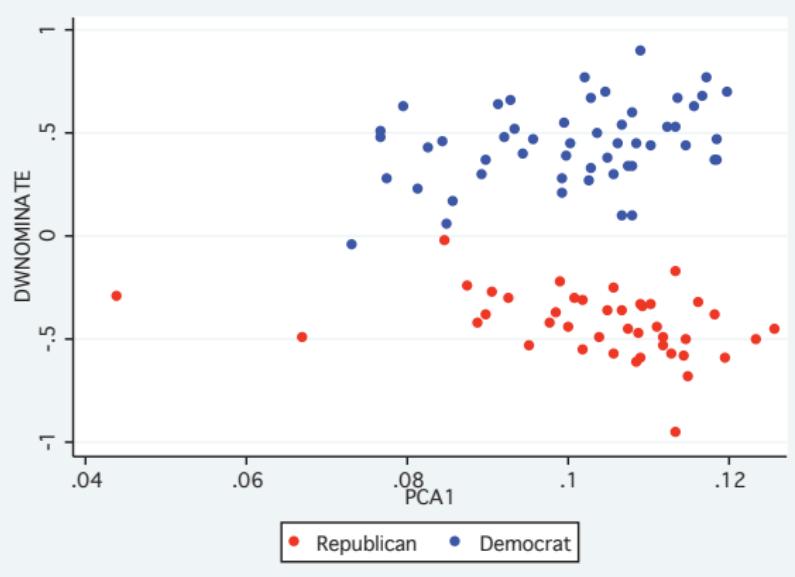
Strangely, in Beauchamp's work, PC1 **uncorrelated** with first dimension of roll calls scores.

Partner Exercise



Strangely, in Beauchamp's work, PC1 **uncorrelated** with first dimension of roll calls scores.
why?

Partner Exercise



Strangely, in Beauchamp's work, PC1 **uncorrelated** with first dimension of roll calls scores.
why?

Clustering

Clustering

Clustering:

Clustering

Clustering: look for 'groups' in data explicitly.

Clustering

Clustering: look for 'groups' in data explicitly.

Partition methods are most common:

Clustering

Clustering: look for 'groups' in data explicitly.

Partition methods are most common:

→ Include *K-means*, for which one pre-specifies cluster number,

Clustering

Clustering: look for 'groups' in data explicitly.

Partition methods are most common:

- Include *K-means*, for which one pre-specifies cluster number, and algorithm puts observations into clusters which minimize within cluster sum of squares

Clustering

Clustering: look for 'groups' in data explicitly.

Partition methods are most common:

- Include *K-means*, for which one pre-specifies cluster number, and algorithm puts observations into clusters which minimize within cluster sum of squares

Clustering

Clustering: look for 'groups' in data explicitly.

Partition methods are most common:

- Include *K-means*, for which one pre-specifies cluster number, and algorithm puts observations into clusters which minimize within cluster sum of squares
- so pick s such that $\sum_{i=1}^k \sum_{x_j \in S_i} ||x_j - \mu_i||^2$ is minimized where μ_i is (vector) mean of points in cluster S_i .

Clustering

Clustering: look for 'groups' in data explicitly.

Partition methods are most common:

- Include *K-means*, for which one pre-specifies cluster number, and algorithm puts observations into clusters which minimize within cluster sum of squares
- so pick s such that $\sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$ is minimized where μ_i is (vector) mean of points in cluster S_i .
- observations (documents) within clusters should be as similar as possible,

Clustering

Clustering: look for 'groups' in data explicitly.

Partition methods are most common:

- Include *K-means*, for which one pre-specifies cluster number, and algorithm puts observations into clusters which minimize within cluster sum of squares
- so pick s such that $\sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$ is minimized where μ_i is (vector) mean of points in cluster S_i .
- observations (documents) within clusters should be as similar as possible, observations (documents) in different clusters should be as different as possible.

“General purpose computer-assisted clustering and conceptualization”, Grimmer and King (2010)

“General purpose computer-assisted clustering and conceptualization”, Grimmer and King (2010)

Motivation: there are an **infinite** number of possible algorithms,

“General purpose computer-assisted clustering and conceptualization”, Grimmer and King (2010)

Motivation: there are an **infinite** number of possible algorithms, and hard to choose which one makes most ‘sense’ for given substantive problem.

“General purpose computer-assisted clustering and conceptualization”, Grimmer and King (2010)

Motivation: there are an **infinite** number of possible algorithms, and hard to choose which one makes most ‘sense’ for given substantive problem.

G&K Use ‘all’ of them,

“General purpose computer-assisted clustering and conceptualization”, Grimmer and King (2010)

Motivation: there are an **infinite** number of possible algorithms, and hard to choose which one makes most ‘sense’ for given substantive problem.

G&K Use ‘all’ of them, and allow **users** to choose one (or more) that maximizes some ‘insightful-ness’ criteria.

“General purpose computer-assisted clustering and conceptualization”, Grimmer and King (2010)

Motivation: there are an **infinite** number of possible algorithms, and hard to choose which one makes most ‘sense’ for given substantive problem.

G&K Use ‘all’ of them, and allow **users** to choose one (or more) that maximizes some ‘insightful-ness’ criteria.

This requires thoughtful **visualization**,

“General purpose computer-assisted clustering and conceptualization”, Grimmer and King (2010)

Motivation: there are an **infinite** number of possible algorithms, and hard to choose which one makes most ‘sense’ for given substantive problem.

G&K Use ‘all’ of them, and allow **users** to choose one (or more) that maximizes some ‘insightful-ness’ criteria.

This requires thoughtful **visualization**, to help humans select particular partition.

"General purpose computer-assisted clustering and conceptualization", Grimmer and King (2010)

Motivation: there are an **infinite** number of possible algorithms, and hard to choose which one makes most 'sense' for given substantive problem.

G&K Use 'all' of them, and allow **users** to choose one (or more) that maximizes some 'insightful-ness' criteria.

This requires thoughtful **visualization**, to help humans select particular partition.

Plus simultaneously allow users to select **combinations** of clusterings that look 'useful'.

Evaluating Clusterings

Evaluating Clusterings

A clustering is good if “the user, or the user's intended audience, finds the chosen clustering useful or insightful.”

Evaluating Clusterings

A clustering is good if “the user, or the user’s intended audience, finds the chosen clustering useful or insightful.”

But this is possibly unfalsifiable,

Evaluating Clusterings

A clustering is good if “the user, or the user’s intended audience, finds the chosen clustering useful or insightful.”

But this is possibly unfalsifiable, and not necessarily scientific...

Evaluating Clusterings

A clustering is good if “the user, or the user’s intended audience, finds the chosen clustering useful or insightful.”

But this is possibly unfalsifiable, and not necessarily scientific...

So suggest some more measurable/formal evaluation mechanisms:

Evaluating Clusterings

A clustering is good if “the user, or the user’s intended audience, finds the chosen clustering useful or insightful.”

But this is possibly unfalsifiable, and not necessarily scientific...

So suggest some more measurable/formal evaluation mechanisms:

- 1 **Cluster Quality**: randomly draw pairs of documents from **same** cluster and **different** clusters,

Evaluating Clusterings

A clustering is good if “the user, or the user’s intended audience, finds the chosen clustering useful or insightful.”

But this is possibly unfalsifiable, and not necessarily scientific...

So suggest some more measurable/formal evaluation mechanisms:

- 1 **Cluster Quality**: randomly draw pairs of documents from **same** cluster and **different** clusters, and ask **human coders** how closely related they are.

Evaluating Clusterings

A clustering is good if “the user, or the user’s intended audience, finds the chosen clustering useful or insightful.”

But this is possibly unfalsifiable, and not necessarily scientific...

So suggest some more measurable/formal evaluation mechanisms:

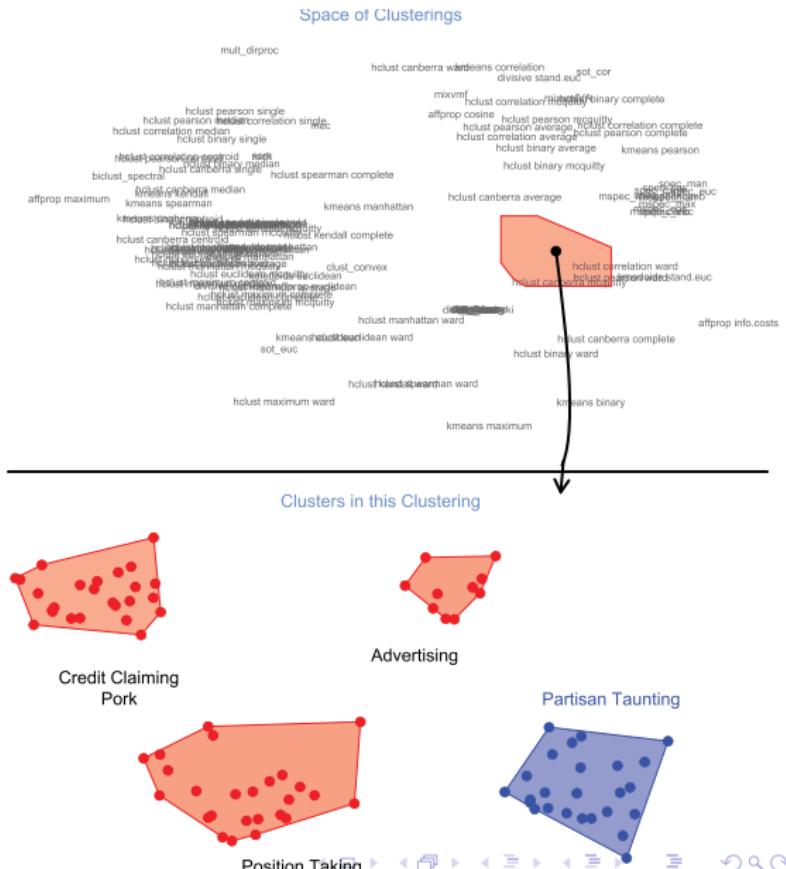
- 1 **Cluster Quality**: randomly draw pairs of documents from **same** cluster and **different** clusters, and ask **human coders** how closely related they are.
- 2 **Discovery Quality**: show scholars the cluster space and see if it improves their understanding of own data

Discovery of Partisan Taunting in Press Releases

Discovery of Partisan Taunting in Press Releases



Discovery of Partisan Taunting in Press Releases



Unsupervised Scaling: Wordfish

Time Series Problems

Time Series Problems



Time Series Problems



Time Series Problems



We suspect that the German Greens and Social Democrats have moved steadily rightwards, post-reunification.

Time Series Problems



We suspect that the German Greens and Social Democrats have moved steadily rightwards, post-reunification.

→ This is a **time series** problem,



Time Series Problems



We suspect that the German Greens and Social Democrats have moved steadily rightwards, post-reunification.

- This is a **time series** problem, but extant techniques struggle...

Time Series Problems



We suspect that the German Greens and Social Democrats have moved steadily rightwards, post-reunification.

- This is a **time series** problem, but extant techniques struggle...
- i.e. hand-coding is expensive,

Time Series Problems



We suspect that the German Greens and Social Democrats have moved steadily rightwards, post-reunification.

- This is a **time series** problem, but extant techniques struggle...
 - i.e. hand-coding is expensive,
 - and hard to find reference texts for **Wordscores** over time

Time Series Problems



10



We suspect that the German Greens and Social Democrats have moved steadily rightwards, post-reunification.

- This is a **time series** problem, but extant techniques struggle...
 - i.e. hand-coding is expensive,
 - and hard to find reference texts for **Wordscores** over time
- need to assume lexicon is pretty **stable**,

Time Series Problems



We suspect that the German Greens and Social Democrats have moved steadily rightwards, post-reunification.

- This is a **time series** problem, but extant techniques struggle...
 - i.e. hand-coding is expensive,
 - and hard to find reference texts for **Wordscores** over time
- need to assume lexicon is pretty **stable**, and that you can identify texts that contain **all** relevant terms.



Slapin & Proksch (2008)

Would be helpful to have an **unsupervised** approach,

Slapin & Proksch (2008)

Would be helpful to have an [unsupervised](#) approach, which is not dependent on [reference texts](#)

Slapin & Proksch (2008)

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique ("A Scaling Model for Estimating Time-Series Party Positions from Text")

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique ("A Scaling Model for Estimating Time-Series Party Positions from Text")

- 1 Begin with **naive Bayes assumption**:

Slapin & Proksch (2008)

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique ("A Scaling Model for Estimating Time-Series Party Positions from Text")

- 1 Begin with **naive Bayes assumption**: idea that each word's occurrence is **independent** of all other words in the text.

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique ("A Scaling Model for Estimating Time-Series Party Positions from Text")

- 1 Begin with **naive Bayes assumption**: idea that each word's occurrence is **independent** of all other words in the text.
→ surely false,

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique ("A Scaling Model for Estimating Time-Series Party Positions from Text")

- 1 Begin with **naive Bayes assumption**: idea that each word's occurrence is **independent** of all other words in the text.
→ surely false, but convenient starting point.

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique (“A Scaling Model for Estimating Time-Series Party Positions from Text”)

- 1 Begin with **naive Bayes assumption**: idea that each word's occurrence is **independent** of all other words in the text.
→ surely false, but convenient starting point.
- 2 Need a (parametric) model for **frequencies** of words.

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique (“A Scaling Model for Estimating Time-Series Party Positions from Text”)

- 1 Begin with **naive Bayes assumption**: idea that each word's occurrence is **independent** of all other words in the text.
→ surely false, but convenient starting point.

- 2 Need a (parametric) model for **frequencies** of words.
→ Choose **Poisson**:

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique (“A Scaling Model for Estimating Time-Series Party Positions from Text”)

- 1 Begin with **naive Bayes assumption**: idea that each word's occurrence is **independent** of all other words in the text.
→ surely false, but convenient starting point.
- 2 Need a (parametric) model for **frequencies** of words.
→ Choose **Poisson**: extremely simple because it has only one parameter

Would be helpful to have an **unsupervised** approach, which is not dependent on **reference texts**

Suggest **WORDFISH** scaling technique (“A Scaling Model for Estimating Time-Series Party Positions from Text”)

- 1 Begin with **naive Bayes assumption**: idea that each word's occurrence is **independent** of all other words in the text.
→ surely false, but convenient starting point.
- 2 Need a (parametric) model for **frequencies** of words.
→ Choose **Poisson**: extremely simple because it has only one parameter— λ (which is mean and variance!).

Poisson set up

Poisson set up

Recall the **density function** for Poisson:

Poisson set up

Recall the **density function** for Poisson:

$$\Pr(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

Poisson set up

Recall the **density function** for Poisson:

$$\Pr(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

And in a 'typical' **GLM** context,

Poisson set up

Recall the **density function** for Poisson:

$$\Pr(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

And in a 'typical' **GLM** context, we would make

Poisson set up

Recall the **density function** for Poisson:

$$\Pr(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

And in a 'typical' **GLM** context, we would make

$$\log(\lambda) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots$$

Poisson set up

Recall the **density function** for Poisson:

$$\Pr(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

And in a 'typical' **GLM** context, we would make

$$\log(\lambda) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots$$

with log-likelihood (dropping constant part),

$$l(\lambda; y) = \sum_{i=1}^n y_i \log \lambda - n\lambda.$$

Poisson set up

Recall the **density function** for Poisson:

$$\Pr(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

And in a 'typical' **GLM** context, we would make

$$\log(\lambda) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots$$

with log-likelihood (dropping constant part),

$$l(\lambda; y) = \sum_{i=1}^n y_i \log \lambda - n\lambda.$$

→ the λ which maximizes this is the **MLE**.

Poisson set up

Recall the **density function** for Poisson:

$$\Pr(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

And in a 'typical' **GLM** context, we would make

$$\log(\lambda) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots$$

with log-likelihood (dropping constant part),

$$l(\lambda; y) = \sum_{i=1}^n y_i \log \lambda - n\lambda.$$

→ the λ which maximizes this is the **MLE**.

Here...

Here...

The count of word j from party i , in year t ,

Here...

The count of word j from party i , in year t ,

$$y_{ijt} \sim \mathcal{P}(\lambda_{ijt})$$

Here...

The count of word j from party i , in year t ,

$$y_{ijt} \sim \mathcal{P}(\lambda_{ijt})$$

and

$$\log(\lambda_{ijt}) = \alpha_{it} + \psi_j + \beta_j \times \omega_{it}$$

Here...

The count of word j from party i , in year t ,

$$y_{ijt} \sim \mathcal{P}(\lambda_{ijt})$$

and

$$\log(\lambda_{ijt}) = \alpha_{it} + \psi_j + \beta_j \times \omega_{it}$$

or

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

So...

So...

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

So...

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

α_{it} fixed effect(s) for party i in time t : some parties have longer manifestos in certain years (which boosts all counts)

So...

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

α_{it} fixed effect(s) for party i in time t : some parties have longer manifestos in certain years (which boosts all counts)

ψ_j word fixed effect: some parties just use certain words more (e.g. their own name)

So...

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

α_{it} fixed effect(s) for party i in time t : some parties have longer manifestos in certain years (which boosts all counts)

ψ_j word fixed effect: some parties just use certain words more (e.g. their own name)

β_j word specific weight: importance of this word in discriminating between party positions.

So...

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

α_{it} fixed effect(s) for party i in time t : some parties have longer manifestos in certain years (which boosts all counts)

ψ_j word fixed effect: some parties just use certain words more (e.g. their own name)

β_j word specific weight: importance of this word in discriminating between party positions.

ω_{it} estimate of party's position in a given year (so, this applies to specific manifesto)

Notes

0

Notes

One dimensional:

Notes

One dimensional: which is assumed to be **left-right**.

Notes

One dimensional: which is assumed to be **left-right**.

- can limit analysis to given issue area to obtain dimensional scaling in **that** space.

Notes

One dimensional: which is assumed to be left-right.

- can limit analysis to given issue area to obtain dimensional scaling in that space.

Parties 'move' to the extent that the words they use look more or less like the words that other parties use.

Notes

One dimensional: which is assumed to be **left-right**.

- can limit analysis to given issue area to obtain dimensional scaling in **that** space.

Parties 'move' to the extent that the words they use look more or less like the words that **other** parties use.

No over time smoothing/constraints:

Notes

One dimensional: which is assumed to be **left-right**.

- can limit analysis to given issue area to obtain dimensional scaling in **that** space.

Parties ‘move’ to the extent that the words they use look more or less like the words that **other** parties use.

No over time smoothing/constraints: party manifesto position in t is not modeled as function of party manifesto position in $t - 1$

Problem

Problem

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Problem

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Nothing on RHS is known:

Problem

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Nothing on RHS is known: everything needs to be estimated.

Problem

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Nothing on RHS is known: everything needs to be estimated.

→ unlike GLM arrangement, where X s are known.

Problem

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Nothing on RHS is known: everything needs to be estimated.

→ unlike GLM arrangement, where X s are known.

but similar to ideal point estimation wherein the legislators' ideal points are not known: $\Phi(\beta_j' \mathbf{x}_i - \alpha_j)$.

Solution I

Solution I

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Solution I

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Suppose we knew the word parameters , ψ_j and β_j .

Solution I

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Suppose we knew the word parameters , ψ_j and β_j .

→ then we could use a Poisson GLM to estimate α_{it} (a constant/fixed effect) and ω_{it} which is the position.

Solution I

NB

$$\lambda_{ijt} = \exp(\alpha_{it} + \psi_j + \beta_j \times \omega_{it})$$

Suppose we knew the word parameters , ψ_j and β_j .

→ then we could use a Poisson GLM to estimate α_{it} (a constant/fixed effect) and ω_{it} which is the position.

Or Suppose we knew the party parameters, ω_{it} and α_{it} . Then we could use a Poisson GLM to estimate ψ_j (a constant/fixed effect) and β_j which is a word specific 'effect'.

Solution II: Intuition

Solution II: Intuition

first start with good guesses (starting values) of both sets of parameters,

Solution II: Intuition

first start with good guesses (starting values) of both sets of parameters,

then run a Poisson regression holding word parameters fixed, and
estimating the party parameters,

Solution II: Intuition

first start with good guesses (starting values) of both sets of parameters,

then run a Poisson regression holding word parameters fixed, and
estimating the party parameters,

then run a Poisson regression holding party parameters fixed, and
estimating the word parameter,

Solution II: Intuition

first start with good guesses (starting values) of both sets of parameters,

then run a Poisson regression holding word parameters fixed, and
estimating the party parameters,

then run a Poisson regression holding party parameters fixed, and
estimating the word parameter,

and iterate across these steps until confident we have correct answers (EM algorithm).

Solution II: Intuition

first start with good guesses (starting values) of both sets of parameters,

then run a Poisson regression holding word parameters fixed, and
estimating the party parameters,

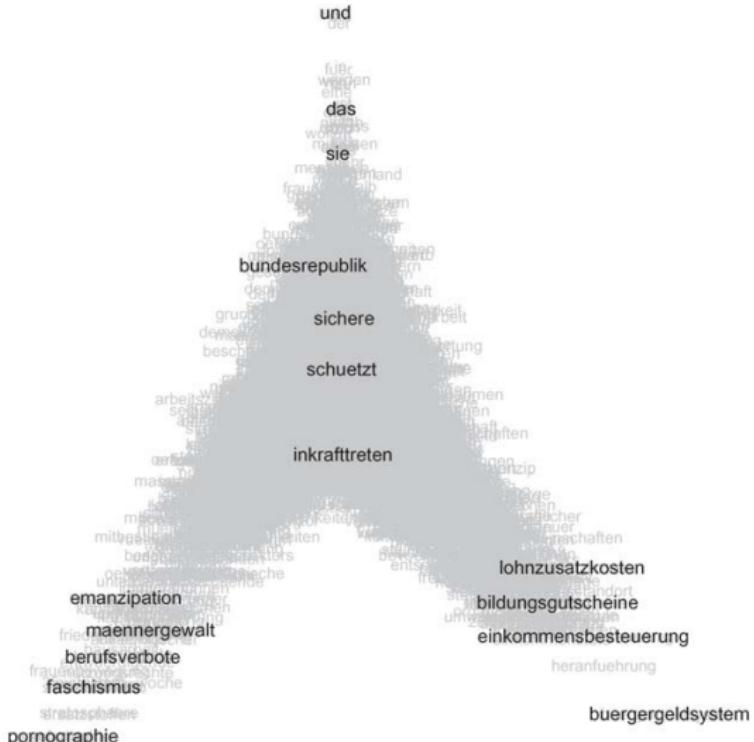
then run a Poisson regression holding party parameters fixed, and
estimating the word parameter,

and iterate across these steps until confident we have correct answers (EM algorithm).

btw can use parametric bootstrap for uncertainty estimates.

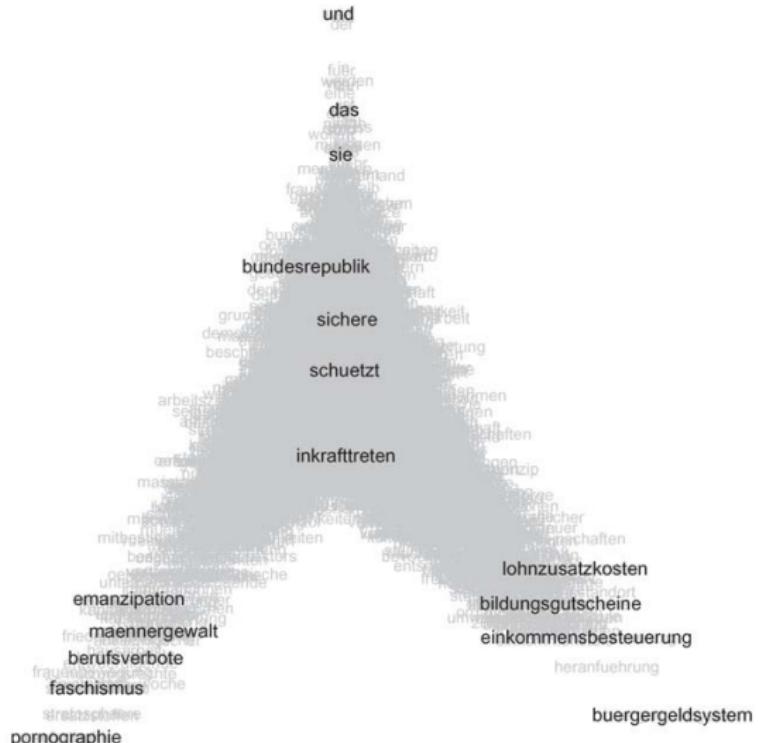
Results

Results



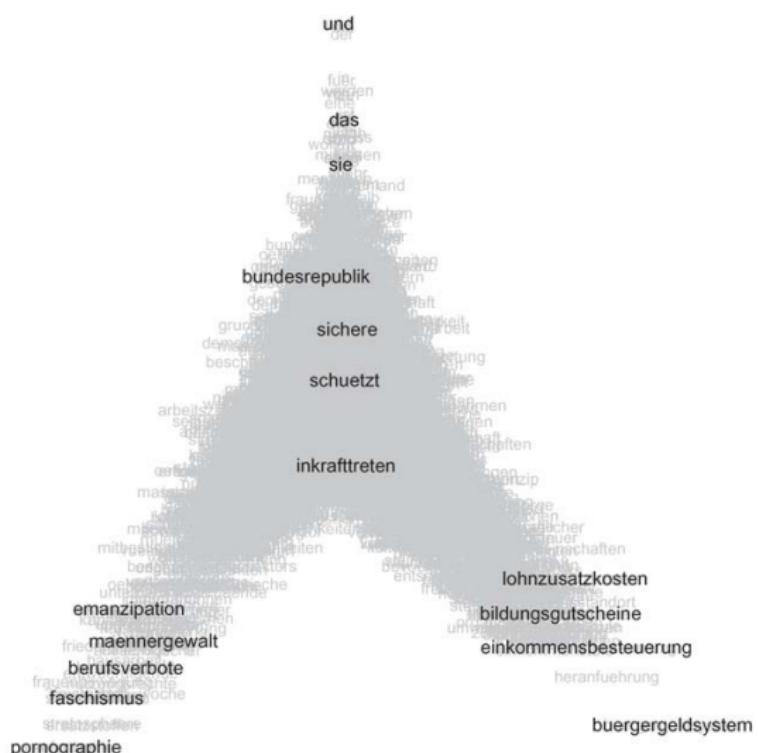
y is word fixed effects:

Results



y is word fixed effects: words with high fixed effects have zero weight (very common).

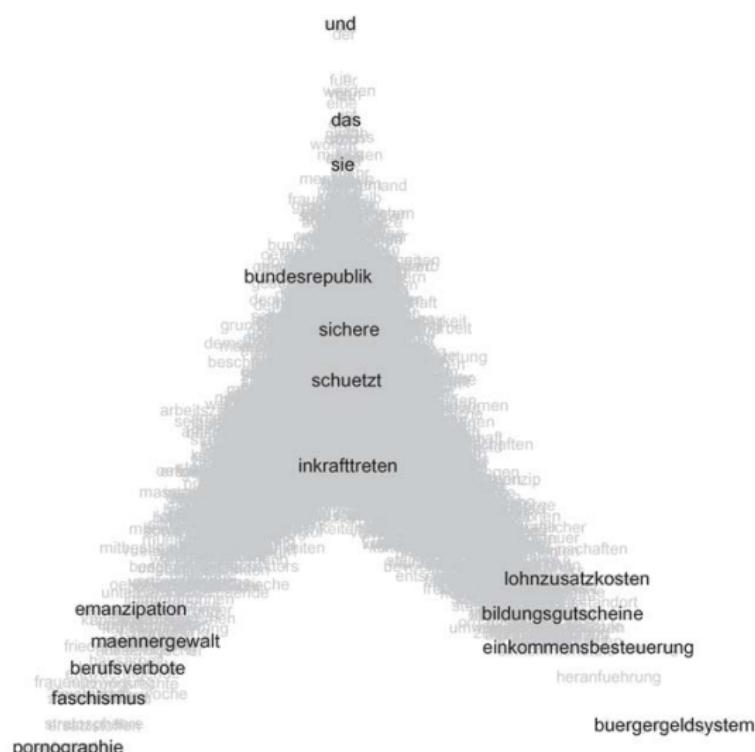
Results



y is word fixed effects: words with high fixed effects have zero weight (very common).

x is word weights:

Results



y is word fixed effects: words with high fixed effects have zero weight (very common).

x is word weights: those with high (absolute) weights discriminate well.

Results II

Results II

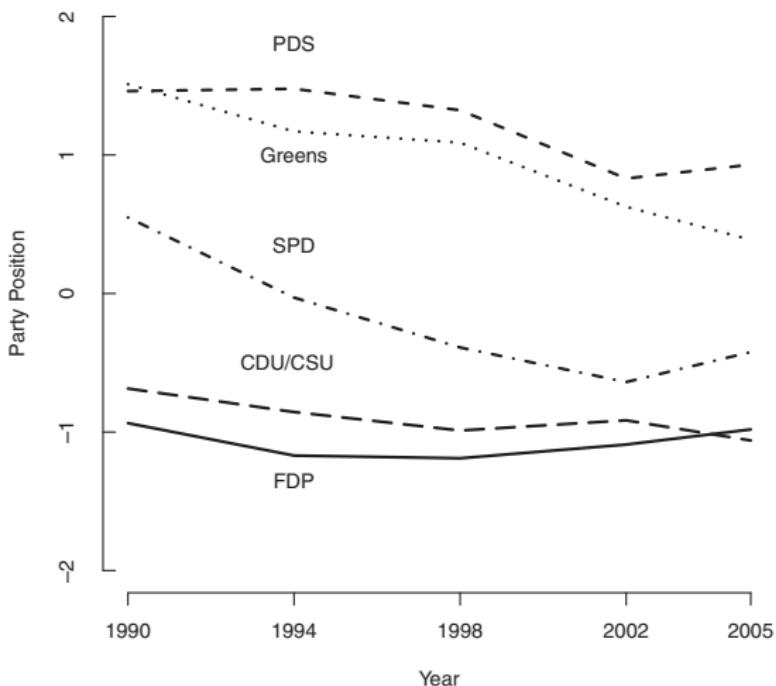
Top 10 Words Placing Parties on the . . .

Dimension	Left	Right
Left-Right	Federal Republic of Germany (BRD) immediate (sofortiger) pornography (Pornographie) sexuality (Sexualität) substitute materials (Ersatzstoffen) stratosphere (Stratosphäre) women's movement (Frauenbewegung) fascism (Faschismus) Two thirds world (Zweidrittewelt) established (etablierten)	general welfare payments (Bürgergeldsystem) introduction (Heranführung) income taxation (Einkommensbesteuerung) non-wage labor costs (Lohnzusatzkosten) business location (Wirtschaftsstandort) university of applied sciences (Fachhochschule) education vouchers (Bildungsgutscheine) mobility (Beweglichkeit) peace tasks (Friedensaufgaben) protection (Protektion)

Results III, the ω_{it} s

Results III, the ω_{its}

(A) Left–Right



Topic Models

Goal

0

Goal

*Topic models are algorithms for discovering the **main themes** that pervade a large and otherwise **unstructured** collection of documents.*

Goal

*Topic models are algorithms for discovering the **main themes** that pervade a large and otherwise **unstructured** collection of documents. Topic models can **organize** the collection according to the discovered themes.*

Blei, 2012

Goal

*Topic models are algorithms for discovering the **main themes** that pervade a large and otherwise **unstructured** collection of documents. Topic models can **organize** the collection according to the discovered themes.*

Blei, 2012

Note that in **social science** we often use the outputs from topic models as a **measurement** strategy:

Goal

*Topic models are algorithms for discovering the **main themes** that pervade a large and otherwise **unstructured** collection of documents. Topic models can **organize** the collection according to the discovered themes.*

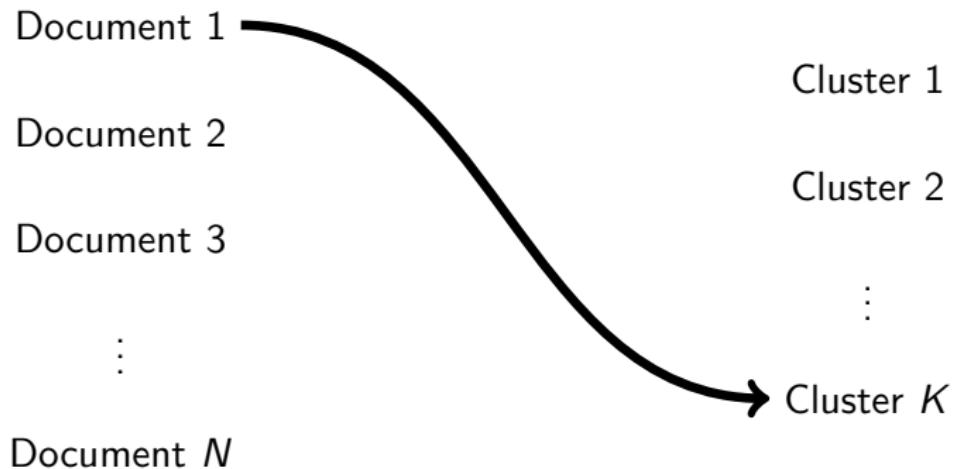
Blei, 2012

Note that in **social science** we often use the outputs from topic models as a **measurement** strategy:

“who pays more attention to education policy, conservatives or liberals?”

Recall: Clustering

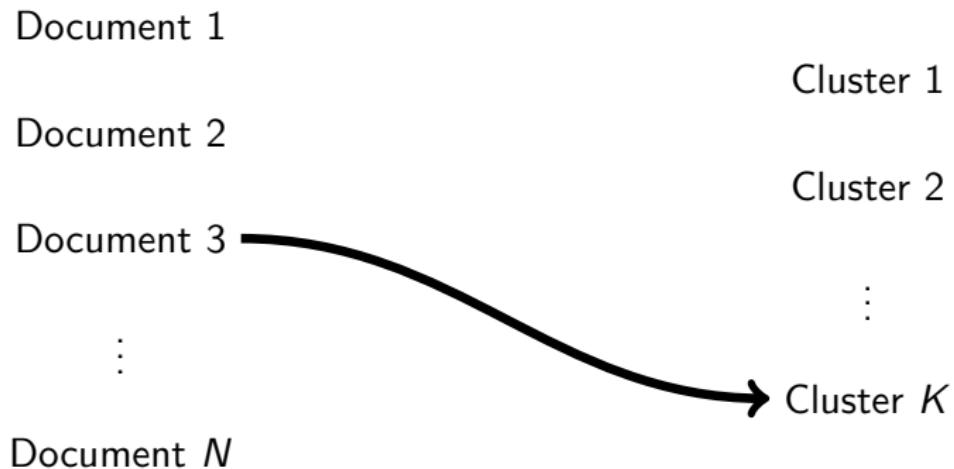
Recall: Clustering



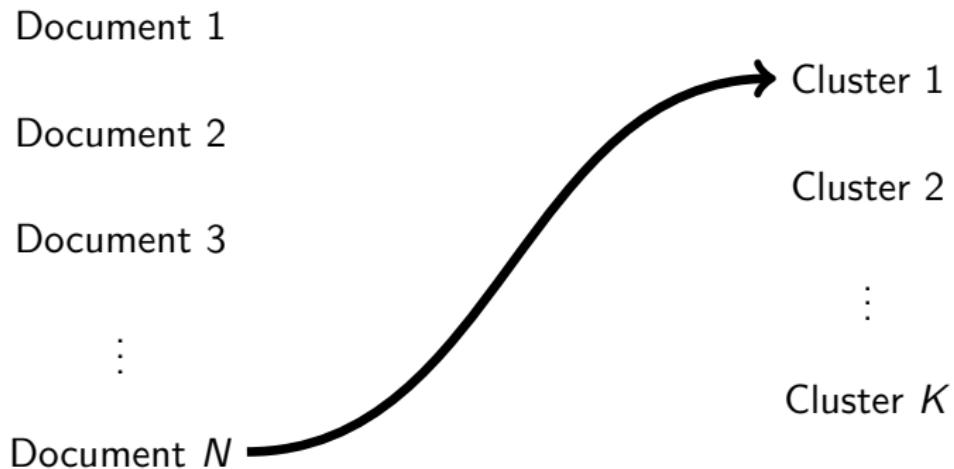
Recall: Clustering



Recall: Clustering



Recall: Clustering



Recall: Clustering

Document 1

Cluster 1

Document 2

Cluster 2

Document 3

⋮

⋮

Cluster K

Document N

Topic Modeling

Topic Modeling

Document 1

Topic 1

Document 2

Topic 2

Document 3

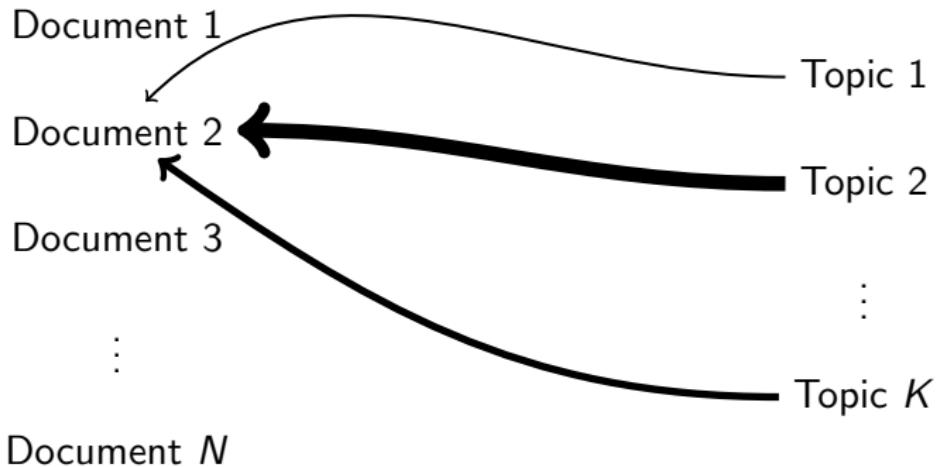
⋮

⋮

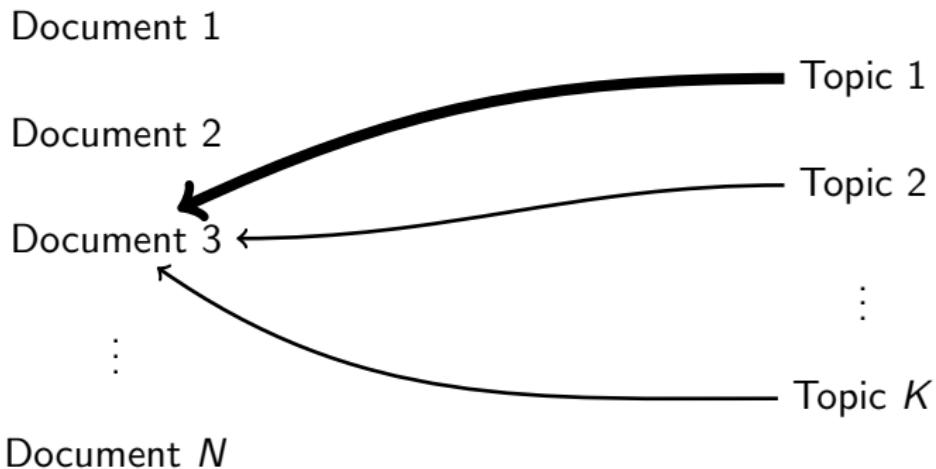
Topic K

Document N

Topic Modeling



Topic Modeling



DGP: intuition

DGP: intuition

Documents exhibit different topics,

DGP: intuition

Documents exhibit different topics, and in different proportions.

DGP: intuition

Documents exhibit different topics, and in different proportions.

E.g. a speech by the Finance minister might be 50% drawn from the trade topic, 40% from the spending topic, 9.9% from the taxation topic, 0.1% from the health topic.

DGP: intuition

Documents exhibit different topics, and in different proportions.

E.g. a speech by the Finance minister might be 50% drawn from the trade topic, 40% from the spending topic, 9.9% from the taxation topic, 0.1% from the health topic.

Think of a **topic** as a **distribution** over a **fixed vocabulary**.

DGP: intuition

Documents exhibit different topics, and in different proportions.

E.g. a speech by the Finance minister might be 50% drawn from the **trade** topic, 40% from the **spending** topic, 9.9% from the **taxation** topic, 0.1% from the **health** topic.

Think of a **topic** as a **distribution** over a **fixed vocabulary**.

E.g. the **trade** topic will have words like import and tariff with high probability.

DGP: intuition

Documents exhibit different topics, and in different proportions.

E.g. a speech by the Finance minister might be 50% drawn from the trade topic, 40% from the spending topic, 9.9% from the taxation topic, 0.1% from the health topic.

Think of a **topic** as a **distribution** over a **fixed vocabulary**.

E.g. the trade topic will have words like import and tariff with high probability.

Technically we assume the topics are generated **first**,

DGP: intuition

Documents exhibit different topics, and in different proportions.

E.g. a speech by the Finance minister might be 50% drawn from the trade topic, 40% from the spending topic, 9.9% from the taxation topic, 0.1% from the health topic.

Think of a **topic** as a **distribution** over a **fixed vocabulary**.

E.g. the trade topic will have words like import and tariff with high probability.

Technically we assume the topics are generated **first**, and the documents are generated second (from those topics).

DGP: intuition

Documents exhibit different topics, and in different proportions.

E.g. a speech by the Finance minister might be 50% drawn from the **trade** topic, 40% from the **spending** topic, 9.9% from the **taxation** topic, 0.1% from the **health** topic.

Think of a **topic** as a **distribution** over a **fixed vocabulary**.

E.g. the **trade** topic will have words like import and tariff with high probability.

Technically we assume the topics are generated **first**, and the documents are generated second (from those topics).

Now, where do the **words** in the documents come from?

Intuition: Generating Words

Intuition: Generating Words

For each document...

Intuition: Generating Words

For each document...

- ① Randomly choose a distribution over topics.

Intuition: Generating Words

For each document...

- ① Randomly choose a **distribution** over topics. That is, choose one of many **multinomial** distributions, each which mixes the topics in different proportions.

Intuition: Generating Words

For each document...

- ① Randomly choose a **distribution** over topics. That is, choose one of many **multinomial** distributions, each which mixes the topics in different proportions.

- ② Then, for every **word** in the document...

Intuition: Generating Words

For each document...

- ① Randomly choose a **distribution** over topics. That is, 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.

Intuition: Generating Words

For each document...

- ① Randomly choose a **distribution** over topics. That is, 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.

First Part

First Part

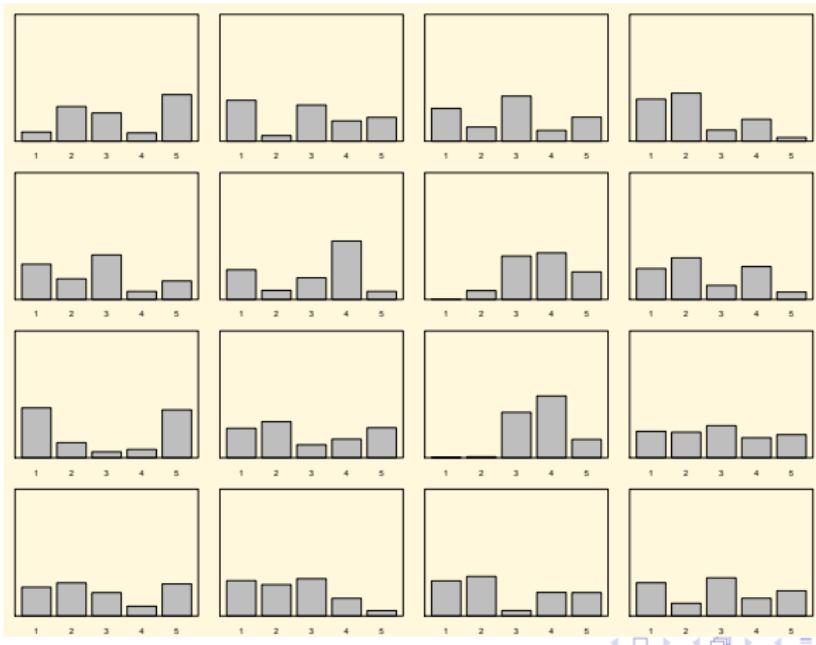
Randomly choose a **distribution** over topics.

First Part

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

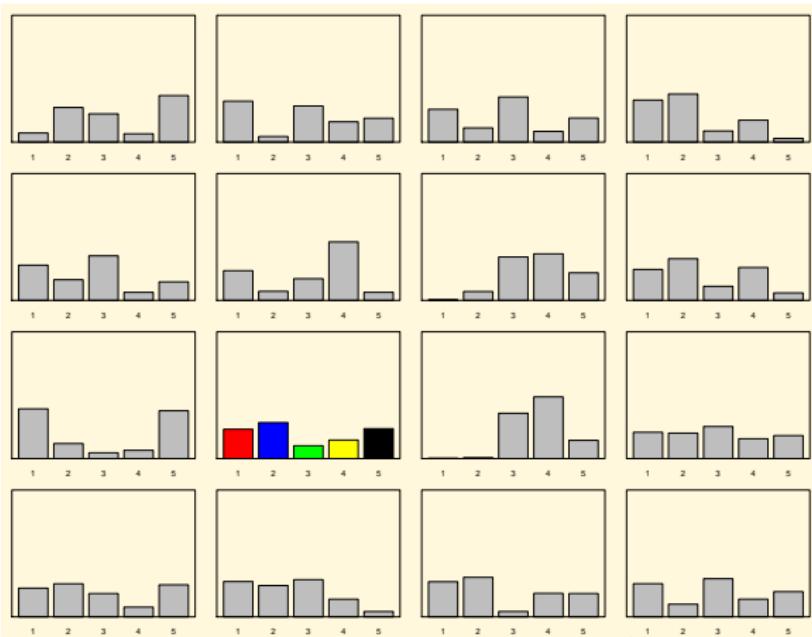
First Part

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



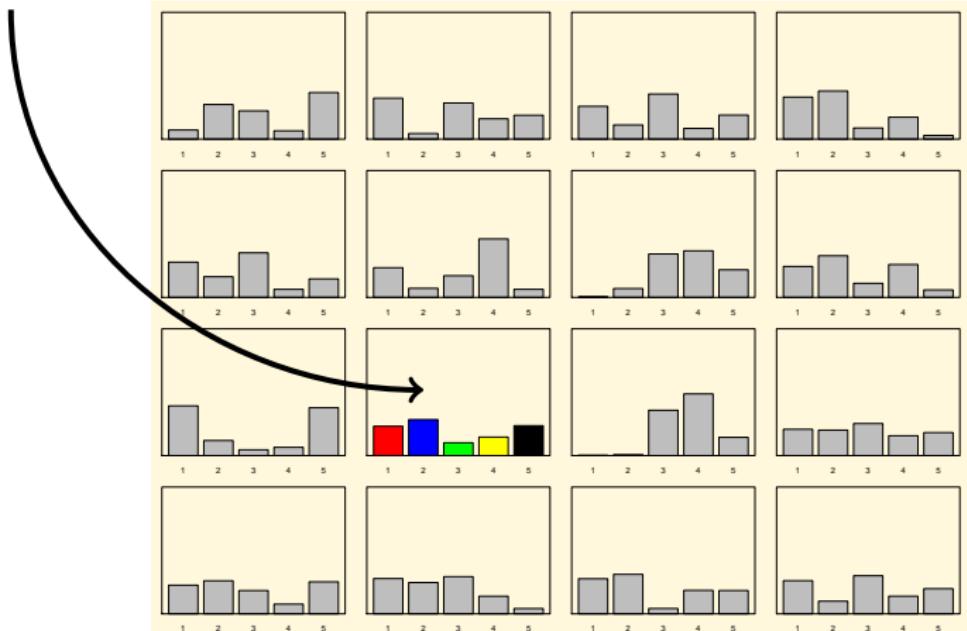
First Part

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



First Part

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



Second Part

Second Part

Then, for every word in the document . . .

Second Part

Then, for every word in the document . . .

- ① Randomly choose a topic from the distribution over topics from step 1.

Second Part

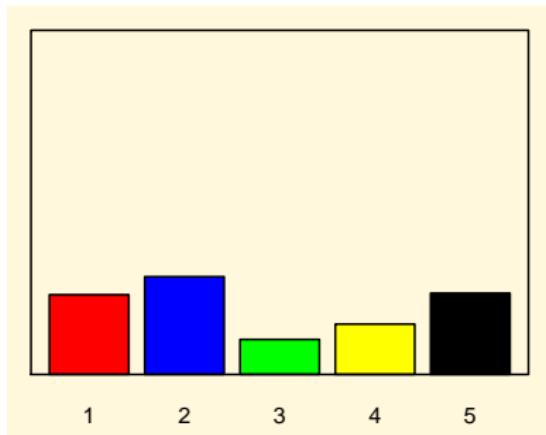
Then, for every word in the document . . .

- 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.

Second Part

Then, for every **word** in the document . . .

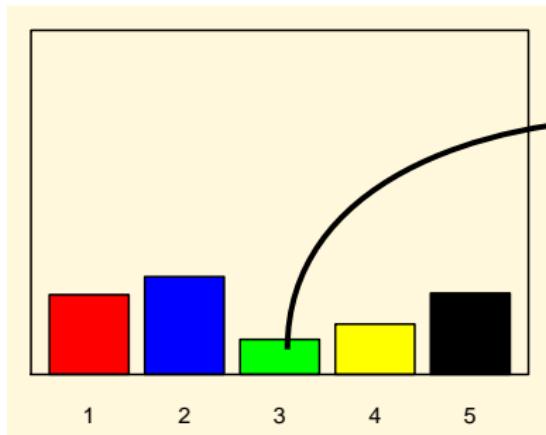
- ① Randomly choose a topic from the distribution over topics from step 1.
- ② Randomly choose a word from the distribution over the vocabulary that the topic implies.



Second Part

Then, for every word in the document . . .

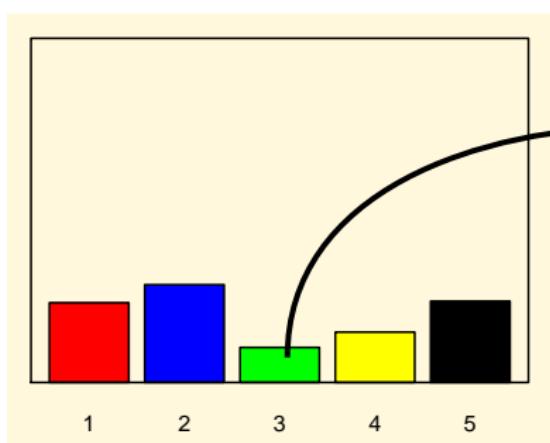
- ① Randomly choose a topic from the distribution over topics from step 1.
- ② Randomly choose a word from the distribution over the vocabulary that the topic implies.



Second Part

Then, for every **word** in the document...

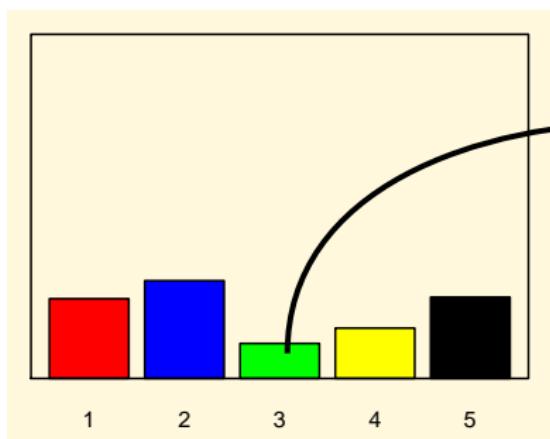
- ① Randomly choose a topic from the distribution over topics from step 1.
- ② Randomly choose a word from the distribution over the vocabulary that the topic implies.



Second Part

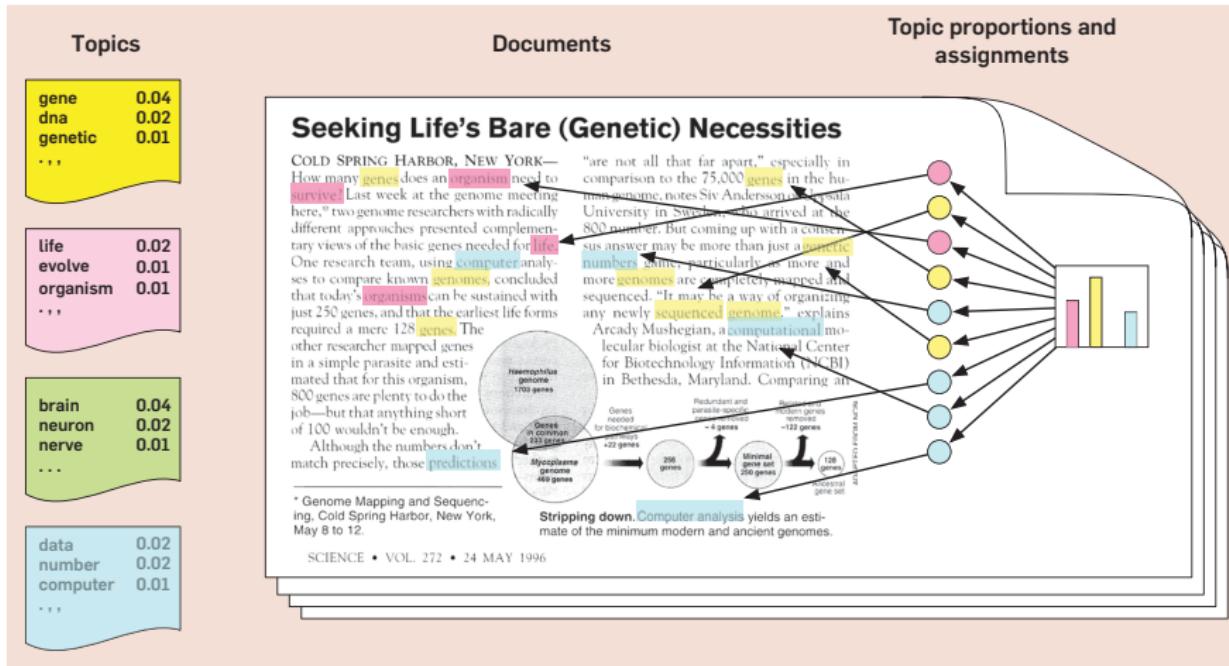
Then, for every word in the document...

- 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.



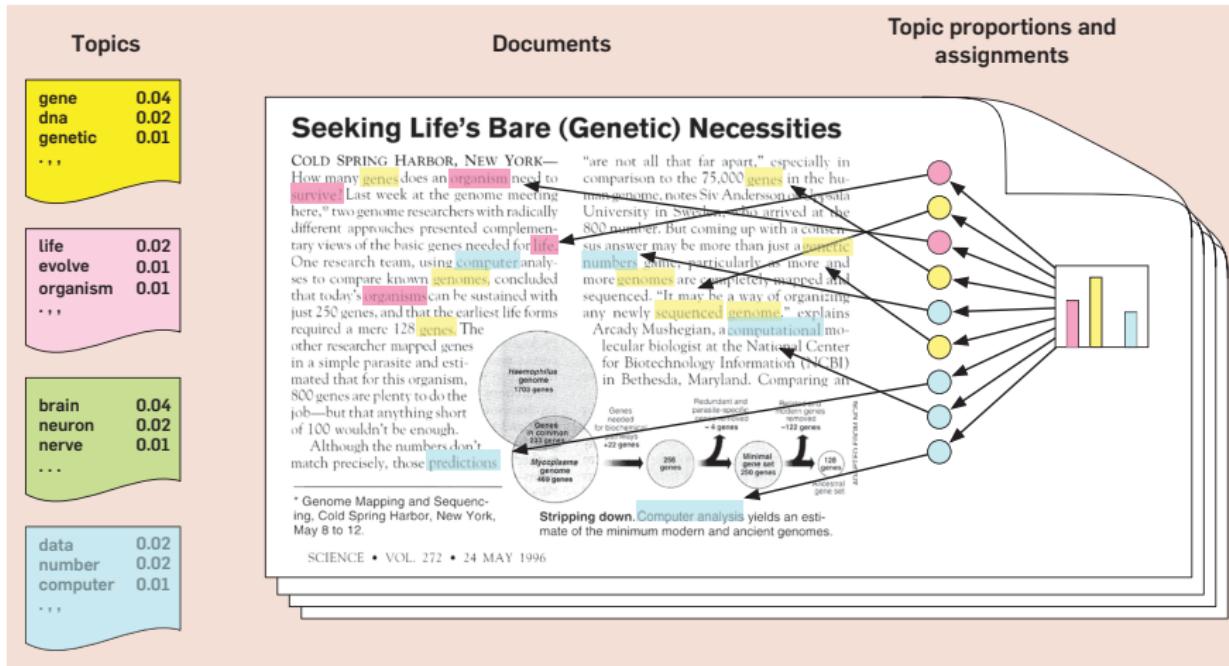
Topic Modeling a Document (Blei, 2012)

Topic Modeling a Document (Blei, 2012)



Note that all documents share same set of topics:

Topic Modeling a Document (Blei, 2012)



Note that all documents share **same set** of topics: but some (e.g. **neuro**) may be (basically) absent in a given document.

Notes

0

Notes

Some of our variables—the documents which contain the words—are observable.

Notes

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

Notes

Some of our variables—the documents which contain the words—are observable. But, topic structure—topics themselves, 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. We use a **Dirichlet** distribution as a prior for that.

Notes

Some of our variables—the documents which contain the words—are observable. But, topic structure—topics themselves, 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. We use a **Dirichlet** distribution as a prior for that.

And Dirichlet is used for the **allocation** of the words in the documents to different topics:

Notes

Some of our variables—the documents which contain the words—are observable. But, topic structure—topics themselves, 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. We use a **Dirichlet** distribution as a prior for that.

And Dirichlet is used for the **allocation** of the words in the documents to different topics: it is used as a prior over the distribution of words (which define the topics).

Notes

Some of our variables—the documents which contain the words—are observable. But, topic structure—topics themselves, 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. We use a **Dirichlet** distribution as a prior for that.

And Dirichlet is used for the **allocation** of the words in the documents to different topics: it is used as a prior over the distribution of words (which define the topics).

→ Latent

Notes

Some of our variables—the documents which contain the words—are observable. But, topic structure—topics themselves, 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. We use a **Dirichlet** distribution as a prior for that.

And Dirichlet is used for the **allocation** of the words in the documents to different topics: it is used as a prior over the distribution of words (which define the topics).

→ Latent Dirichlet

Notes

Some of our variables—the documents which contain the words—are observable. But, topic structure—topics themselves, 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. We use a **Dirichlet** distribution as a prior for that.

And Dirichlet is used for the **allocation** of the words in the documents to different topics: it is used as a prior over the distribution of words (which define the topics).

→ **Latent Dirichlet Allocation.**

Notes

Some of our variables—the documents which contain the words—are observable. But, topic structure—topics themselves, 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. We use a **Dirichlet** distribution as a prior for that.

And Dirichlet is used for the **allocation** of the words in the documents to different topics: it is used as a prior over the distribution of words (which define the topics).

→ Latent Dirichlet Allocation. **LDA**.

A little more formally...

A little more formally...

LDA is a very popular [topic model](#):

A little more formally...

LDA is a very popular **topic model**: a probabilistic procedure to **generate** topics. Thus, a 'generative' model.

A little more formally...

LDA is a very popular **topic model**: a probabilistic procedure to **generate** topics. Thus, a 'generative' model.

There are D documents in the corpus.

A little more formally...

LDA is a very popular **topic model**: a probabilistic procedure to **generate** topics. Thus, a 'generative' model.

There are D documents in the corpus. There are V terms in these D documents.

A little more formally...

LDA is a very popular **topic model**: a probabilistic procedure to **generate** topics. Thus, a 'generative' model.

There are D documents in the corpus. There are V terms in these D documents. For now suppose we **know** the K topic distributions: there are K multinomials containing V elements each.

A little more formally...

LDA is a very popular **topic model**: a probabilistic procedure to **generate** topics. Thus, a ‘generative’ model.

There are D documents in the corpus. There are V terms in these D documents. For now suppose we **know** the K topic distributions: there are K multinomials containing V elements each.

The multinomial distribution for the i th topic is denoted β_i , and $|\beta_i| = V$, meaning that the ‘size’ of this multinomial is equal to the number of different words in the corpus.

So, a little more formally...

So, a little more formally...

For each document...

So, a little more formally...

For each document...

- ① Randomly choose a distribution over topics (multinomial of length K)

So, a little more formally...

For each document...

- ① Randomly choose a **distribution** over topics (multinomial of length K)
- ② Then, for every **word** in the document...

So, a little more formally...

For each document...

- ① Randomly choose a **distribution** over topics (multinomial of length K)
- ② Then, for every **word** in the document...
 - ① Probabilistically draw one of the K topics from the distribution over topics from step 1. E.g. draw β_j

So, a little more formally...

For each document...

- ① Randomly choose a **distribution** over topics (multinomial of length K)
- ② Then, for every **word** in the document...
 - ① Probabilistically draw one of the K topics from the distribution over topics from step 1. E.g. draw β_j
 - ② Probabilistically draw one of the V words from β_j

Aside: Dirichlet distribution

Aside: Dirichlet distribution

The Dirichlet distribution is a [conjugate prior](#) for the [multinomial](#) ('categorical' if you only have one trial) distribution.

Aside: Dirichlet distribution

The Dirichlet distribution is a [conjugate prior](#) for the [multinomial](#) ('categorical' if you only have one trial) distribution. Makes certain calculations easier.

Aside: Dirichlet distribution

The Dirichlet distribution is a **conjugate prior** for the **multinomial** ('categorical' if you only have one trial) distribution. Makes certain calculations easier.

It is parameterized by a vector of positive real numbers α . In principle, one can have $\alpha_1, \dots, \alpha_k$ be different **concentration parameters**, but LDA uses special **symmetric** Dirichlet where all the values of α are the same.

Aside: Dirichlet distribution

The Dirichlet distribution is a **conjugate prior** for the **multinomial** ('categorical' if you only have one trial) distribution. Makes certain calculations easier.

It is parameterized by a vector of positive real numbers α . In principle, one can have $\alpha_1, \dots, \alpha_k$ be different **concentration parameters**, but LDA uses special **symmetric** Dirichlet where all the values of α are the same.

Larger values of α (assuming we are in symmetric case) mean we think (*a priori*) that documents are generally an **even mix** of the topics.

Aside: Dirichlet distribution

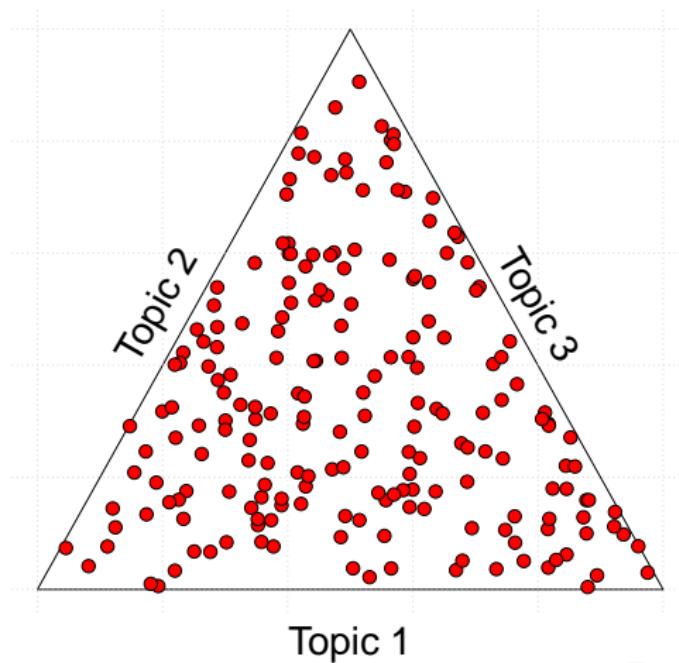
The Dirichlet distribution is a **conjugate prior** for the **multinomial** ('categorical' if you only have one trial) distribution. Makes certain calculations easier.

It is parameterized by a vector of positive real numbers α . In principle, one can have $\alpha_1, \dots, \alpha_k$ be different **concentration parameters**, but LDA uses special **symmetric** Dirichlet where all the values of α are the same.

Larger values of α (assuming we are in symmetric case) mean we think (*a priori*) that documents are generally an **even mix** of the topics. If α is small (less than 1) we think a given document is generally from one or a few topics.

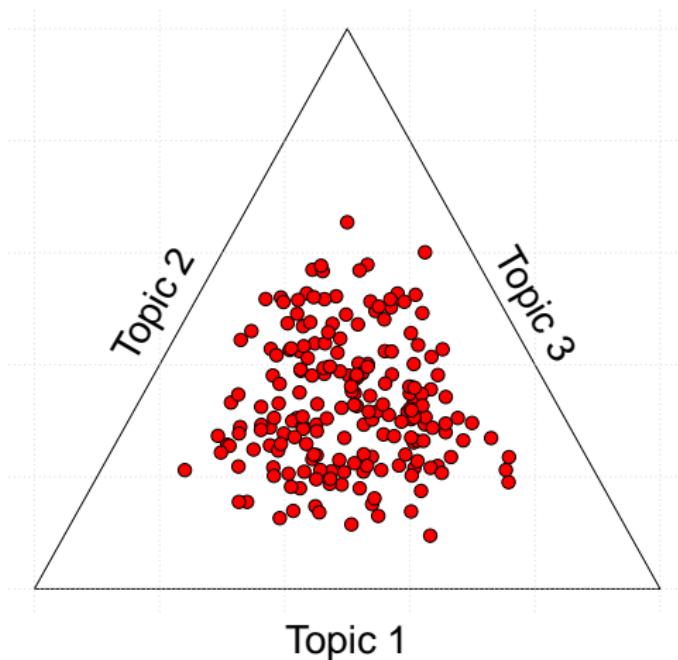
Example of Dirichlet

200 documents, 3 topics, $\alpha = 1$
(uniform)



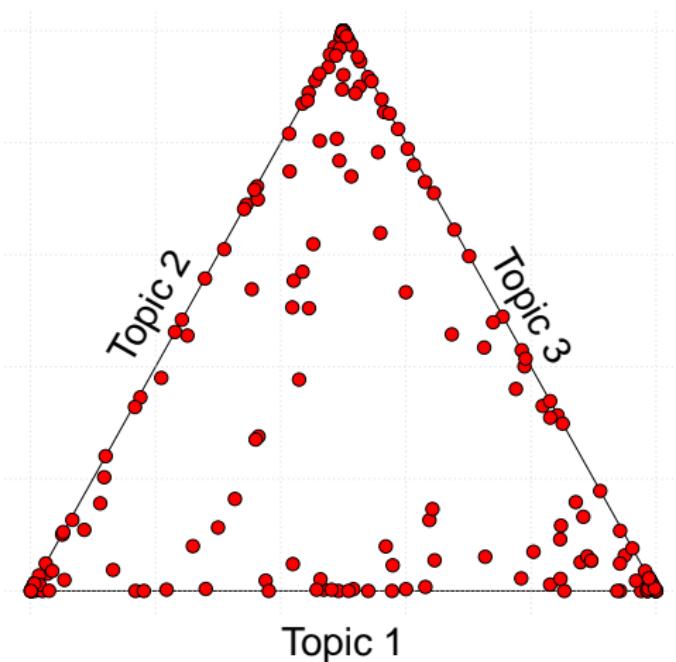
Example of Dirichlet

200 documents, 3 topics, $\alpha = 5$



Example of Dirichlet

200 documents, 3 topics, $\alpha = 0.2$



And actually...

And actually...

We also use a symmetric Dirichlet prior on the per topic word distributions.

And actually...

We also use a symmetric Dirichlet prior on the per topic word distributions. That is, the prior on the β s.

And actually...

We also use a symmetric Dirichlet prior on the per topic word distributions. That is, the prior on the β s.

→ A high concentration parameter means each topic is a mixture of most of the words.

And actually...

We also use a symmetric Dirichlet prior on the per topic word distributions. That is, the prior on the β_i s.

→ A high concentration parameter means each topic is a mixture of most of the words. A low concentration parameter means each topic is a mixture of a few of the words.

And actually...

We also use a symmetric Dirichlet prior on the per topic word distributions. That is, the prior on the β_i s.

→ A high concentration parameter means each topic is a mixture of most of the words. A low concentration parameter means each topic is a mixture of a few of the words.

In practice, one can estimate the concentration parameters, or simply set them at suggested values.

And actually...

We also use a symmetric Dirichlet prior on the per topic word distributions. That is, the prior on the β s.

→ A high concentration parameter means each topic is a mixture of most of the words. A low concentration parameter means each topic is a mixture of a few of the words.

In practice, one can estimate the concentration parameters, or simply set them at suggested values.

We want topic models to be similar as we increase number of topics. Can use asymmetric priors for per-document topic distributions (the θ s).

And actually...

We also use a symmetric Dirichlet prior on the per topic word distributions. That is, the prior on the β s.

→ A high concentration parameter means each topic is a mixture of most of the words. A low concentration parameter means each topic is a mixture of a few of the words.

In practice, one can estimate the concentration parameters, or simply set them at suggested values.

We want topic models to be similar as we increase number of topics. Can use asymmetric priors for per-document topic distributions (the θ s). Asymmetric priors on per-topic word distributions don't do much.

And actually...

We also use a symmetric Dirichlet prior on the per topic word distributions. That is, the prior on the β s.

→ A high concentration parameter means each topic is a mixture of most of the words. A low concentration parameter means each topic is a mixture of a few of the words.

In practice, one can estimate the concentration parameters, or simply set them at suggested values.

We want topic models to be similar as we increase number of topics. Can use asymmetric priors for per-document topic distributions (the θ s). Asymmetric priors on per-topic word distributions don't do much. Wallach et al "Rethinking LDA: Why Priors Matter"

We now know that...

We now know that...

We observe $w_{d,n}$.

We now know that...

We observe $w_{d,n}$. And there are N words in a given document.

We now know that...

We observe $w_{d,n}$. And there are N words in a given document. And D documents in a corpus.

We now know that...

We observe $w_{d,n}$. And there are N words in a given document. And D documents in a corpus.

The $w_{d,n}$ we see depends on $z_{d,n}$, the topic assignment for that word (“this word will be from topic 4”).

We now know that...

We observe $w_{d,n}$. And there are N words in a given document. And D documents in a corpus.

The $w_{d,n}$ we see depends on $z_{d,n}$, the topic assignment for that word (“this word will be from topic 4”).

The $z_{d,n}$ depends on θ_d , the topic mix for a given document d in which the words sit.

We now know that...

We observe $w_{d,n}$. And there are N words in a given document. And D documents in a corpus.

The $w_{d,n}$ we see depends on $z_{d,n}$, the topic assignment for that word (“this word will be from topic 4”).

The $z_{d,n}$ depends on θ_d , the topic mix for a given document d in which the words sit.

The θ_d depends on our prior for the relevant Dirichlet,

We now know that...

We observe $w_{d,n}$. And there are N words in a given document. And D documents in a corpus.

The $w_{d,n}$ we see depends on $z_{d,n}$, the topic assignment for that word ("this word will be from topic 4").

The $z_{d,n}$ depends on θ_d , the topic mix for a given document d in which the words sit.

The θ_d depends on our prior for the relevant Dirichlet, α .

And we know that the actual value that $w_{d,n}$ takes depends on the distribution over words that the relevant topic entails, the β ("the word from topic 4 is "income" in this case")

While the β depends on the prior for the relevant Dirichlet, η

Plate Diagram for LDA

Plate Diagram for LDA

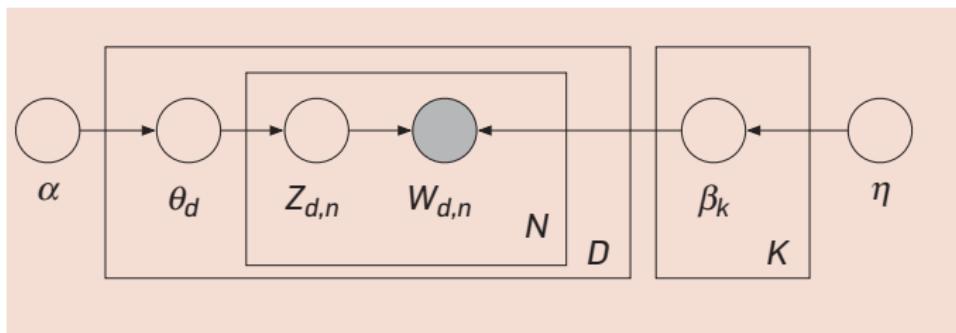
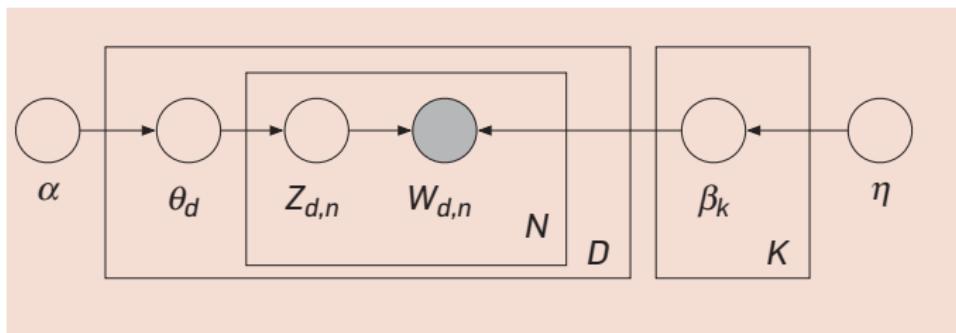
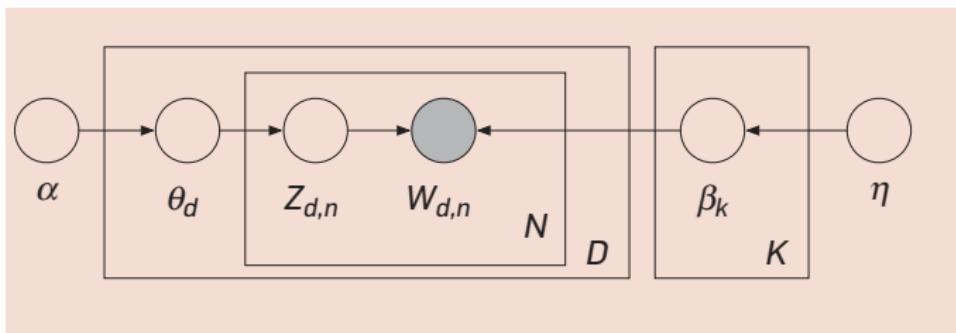


Plate Diagram for LDA



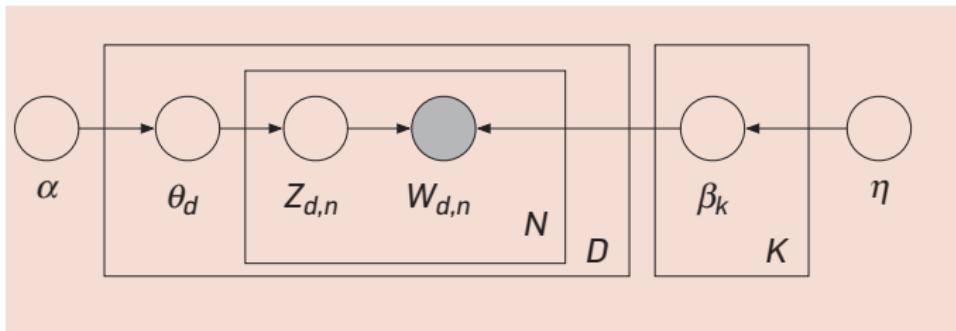
Solid nodes are observed;

Plate Diagram for LDA



Solid nodes are observed; empty nodes are latent.

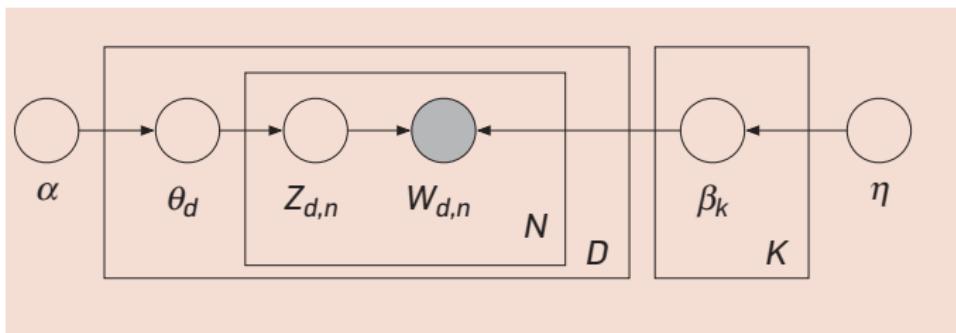
Plate Diagram for LDA



Solid nodes are observed; empty nodes are latent.

Plates imply replication.

Plate Diagram for LDA



Solid nodes are observed; empty nodes are latent.

Plates imply replication.

Note that $w_{d,n}$ depends on $z_{d,n}$ (the mix of topics for that document) and $\beta_{1:K}$ (all the topics in terms of their distributions over the words).

Estimation

Estimation

Ultimately,

Estimation

Ultimately, we will use the observed data, the [words](#),

Estimation

Ultimately, we will use the observed data, the **words**, to make an inference about the **latent** parameters:

Estimation

Ultimately, we will use the observed data, the **words**, to make an inference about the **latent** parameters: the β s, the z s, the θ s.

Estimation

Ultimately, we will use the observed data, the **words**, to make an inference about the **latent** parameters: the β s, the z s, the θ s. That will be a **conditional** probability.

Estimation

Ultimately, we will use the observed data, the **words**, to make an inference about the **latent** parameters: the β s, the z s, the θ s. That will be a **conditional** probability.

We start with the **joint distribution** implied by the problem:

Estimation

Ultimately, we will use the observed data, the **words**, to make an inference about the **latent** parameters: the β s, the z s, the θ s. That will be a **conditional** probability.

We start with the **joint distribution** implied by the problem:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) =$$

Estimation

Ultimately, we will use the observed data, the **words**, to make an inference about the **latent** parameters: the β s, the z s, the θ s. That will be a **conditional** probability.

We start with the **joint distribution** implied by the problem:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) =$$

$$\prod_K^{i=1} p(\beta_i) \prod_D^{d=1} p(\theta_d) \left(\prod_N^{n=1} p(z_{d,n} | \theta_d) p(w_{d,n} | \beta_{1:K}, z_{d,n}) \right)$$

Posterior

Posterior

Generally we want

$$p(\theta|\mathbf{D}, M) = \frac{p(\mathbf{D}, \theta|M)}{\int p(\mathbf{D}, \theta|M)d\theta} = \boxed{\frac{p(\theta|M)p(\mathbf{D}|\theta, M)}{p(\mathbf{D}|M)}}$$

Posterior

Generally we want

$$p(\theta|\mathbf{D}, M) = \frac{p(\mathbf{D}, \theta|M)}{\int p(\mathbf{D}, \theta|M)d\theta} = \boxed{\frac{p(\theta|M)p(\mathbf{D}|\theta, M)}{p(\mathbf{D}|M)}}$$

Here (Blei, 2012), that is

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D} | w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})}$$

Posterior

Generally we want

$$p(\theta|\mathbf{D}, M) = \frac{p(\mathbf{D}, \theta|M)}{\int p(\mathbf{D}, \theta|M)d\theta} = \boxed{\frac{p(\theta|M)p(\mathbf{D}|\theta, M)}{p(\mathbf{D}|M)}}$$

Here (Blei, 2012), that is

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D} | w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})}$$

Can get ‘evidence’ (denominator) by summing joint distribution over
every possible topic structure:

Posterior

Generally we want

$$p(\theta|\mathbf{D}, M) = \frac{p(\mathbf{D}, \theta|M)}{\int p(\mathbf{D}, \theta|M)d\theta} = \boxed{\frac{p(\theta|M)p(\mathbf{D}|\theta, M)}{p(\mathbf{D}|M)}}$$

Here (Blei, 2012), that is

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D} | w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})}$$

Can get ‘evidence’ (denominator) by summing joint distribution over **every possible topic structure**: every possible way of assigning each word to a topic.

Posterior

Generally we want

$$p(\theta|\mathbf{D}, M) = \frac{p(\mathbf{D}, \theta|M)}{\int p(\mathbf{D}, \theta|M)d\theta} = \boxed{\frac{p(\theta|M)p(\mathbf{D}|\theta, M)}{p(\mathbf{D}|M)}}$$

Here (Blei, 2012), that is

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D} | w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})}$$

Can get ‘evidence’ (denominator) by summing joint distribution over **every possible topic structure**: every possible way of assigning each word to a topic. But this is impossible, so simulate/approximate.

Results

Results

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

Results

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

The word distribution for each topic.

Results

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

The word distribution for each topic.

The topic distribution for each document.

Results

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

The word distribution for each topic.

The topic distribution for each document.

Some implementations allow you to estimate e.g. α , in which case this is also returned.

Results

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

The word distribution for each topic.

The topic distribution for each document.

Some implementations allow you to estimate e.g. α , in which case this is also returned. And perhaps some kind of fit statistic(s).

A Manifesto Example

A Manifesto Example

69 UK manifestos.

A Manifesto Example

69 UK manifestos. Some preprocessing.

A Manifesto Example

69 UK manifestos. Some preprocessing. Used `topicmodels` to fit five topics.

A Manifesto Example

69 UK manifestos. Some preprocessing. Used `topicmodels` to fit five topics. Has Gibbs sampling and variational options.

A Manifesto Example

69 UK manifestos. Some preprocessing. Used `topicmodels` to fit five topics. Has Gibbs sampling and variational options.

The (some selected) word distributions for each topic.

A Manifesto Example

69 UK manifestos. Some preprocessing. Used `topicmodels` to fit five topics. Has Gibbs sampling and variational options.

The (some selected) word distributions for each topic. Sum down the columns is one.

A Manifesto Example

69 UK manifestos. Some preprocessing. Used `topicmodels` to fit five topics. Has Gibbs sampling and variational options.

The (some selected) word distributions for each topic. Sum down the columns is one.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
conservative	0.00188	0.00088	0.00185	0.00221	0.00168
party	0.00145	0.00067	0.00066	0.00577	0.00093
general	0.00073	0.00033	0.00018	0.00192	0.00040
election	0.00079	0.00053	0.00022	0.00235	0.00076
manifesto	0.00059	0.00078	0.00032	0.00099	0.00048
:	:	:	:	:	:

Continued...

Continued...

'Top' 6 most frequent words in each topic:

Continued...

'Top' 6 most frequent words in each topic: might help interpretation (!)

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	people	new	[markup]	new	must
2	local	government	people	labour	government
3	government	people	new	government	labour
4	new	continue	work	people	shall
5	tax	can	[markup]	shall	can
6	liberal	conservative	support	britain	policy

Continued...

'Top' 6 most frequent words in each topic: might help interpretation (!)

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	people	new	[markup]	new	must
2	local	government	people	labour	government
3	government	people	new	government	labour
4	new	continue	work	people	shall
5	tax	can	[markup]	shall	can
6	liberal	conservative	support	britain	policy

Up to analyst to label the topics!

Continued...

'Top' 6 most frequent words in each topic: might help interpretation (!)

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	people	new	[markup]	new	must
2	local	government	people	labour	government
3	government	people	new	government	labour
4	new	continue	work	people	shall
5	tax	can	[markup]	shall	can
6	liberal	conservative	support	britain	policy

Up to analyst to label the topics!

Meaningless 'junk' topics not unusual:

Continued...

'Top' 6 most frequent words in each topic: might help interpretation (!)

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	people	new	[markup]	new	must
2	local	government	people	labour	government
3	government	people	new	government	labour
4	new	continue	work	people	shall
5	tax	can	[markup]	shall	can
6	liberal	conservative	support	britain	policy

Up to analyst to label the topics!

Meaningless 'junk' topics not unusual: debate as to whether one has to interpret every topic.

Continued

Continued

The topic distribution for each document...

Continued

The topic distribution for each document...

Continued

The topic distribution for each document...

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
doc 1	0.00009	0.00009	0.00009	0.00009	0.99965
doc 2	0.00011	0.00011	0.00011	0.00011	0.99954
doc 3	0.00010	0.00010	0.00010	0.00010	0.99959
doc 4	0.00006	0.00006	0.00006	0.00006	0.99978
doc 5	0.00002	0.00002	0.00002	0.00002	0.99991
doc 6	0.00019	0.00019	0.00019	0.00019	0.99924
:	:	:	:	:	:

Continued

The topic distribution for each document...

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
doc 1	0.00009	0.00009	0.00009	0.00009	0.99965
doc 2	0.00011	0.00011	0.00011	0.00011	0.99954
doc 3	0.00010	0.00010	0.00010	0.00010	0.99959
doc 4	0.00006	0.00006	0.00006	0.00006	0.99978
doc 5	0.00002	0.00002	0.00002	0.00002	0.99991
doc 6	0.00019	0.00019	0.00019	0.00019	0.99924
:	:	:	:	:	:

Practical Notes I

Practical Notes I

Texts are usually **preprocessed**:

Practical Notes I

Texts are usually **preprocessed**: stop words removed,

Practical Notes I

Texts are usually **preprocessed**: stop words removed, (very) rare tokens removed.

Practical Notes I

Texts are usually **preprocessed**: stop words removed, (very) rare tokens removed. Punctuation often removed.

Practical Notes I

Texts are usually **preprocessed**: stop words removed, (very) rare tokens removed. Punctuation often removed. Stemming seems less common.

Practical Notes I

Texts are usually **preprocessed**: stop words removed, (very) rare tokens removed. Punctuation often removed. Stemming seems less common.

In most social science examples, the **number of topics**, K , is not picked automatically.

Practical Notes I

Texts are usually **preprocessed**: stop words removed, (very) rare tokens removed. Punctuation often removed. Stemming seems less common.

In most social science examples, the **number of topics**, K , is not picked automatically. Analysts select various K s and check that their results are ‘robust’. But see over.

Practical Notes I

Texts are usually **preprocessed**: stop words removed, (very) rare tokens removed. Punctuation often removed. Stemming seems less common.

In most social science examples, the **number of topics**, K , is not picked automatically. Analysts select various K s and check that their results are ‘robust’. But see over.

As with all **unsupervised** learning,

Practical Notes I

Texts are usually **preprocessed**: stop words removed, (very) rare tokens removed. Punctuation often removed. Stemming seems less common.

In most social science examples, the **number of topics**, K , is not picked automatically. Analysts select various K s and check that their results are ‘robust’. But see over.

As with all **unsupervised** learning, interpretation is non-trivial, and requires a lot of validation. Rant: ‘just-so’ stories abound. Lazy analysts conclude whatever they want.

Practical Notes I

Texts are usually **preprocessed**: stop words removed, (very) rare tokens removed. Punctuation often removed. Stemming seems less common.

In most social science examples, the **number of topics**, K , is not picked automatically. Analysts select various K s and check that their results are ‘robust’. But see over.

As with all **unsupervised** learning, interpretation is non-trivial, and requires a lot of validation. Rant: ‘just-so’ stories abound. Lazy analysts conclude whatever they want.

Practical Notes II: Picking k

Practical Notes II: Picking k

Crudely: in social science,

Practical Notes II: Picking k

Crudely: in social science, researchers fit ‘enough’ topics until they see what they think they should.

Practical Notes II: Picking k

Crudely: in social science, researchers fit ‘enough’ topics until they see what they think they should. E.g. a certain topic—like finance suddenly peels off—so stop there.

Practical Notes II: Picking k

Crudely: in social science, researchers fit ‘enough’ topics until they see what they think they should. E.g. a certain topic—like finance suddenly peels off—so stop there.

→ Check findings are robust in the neighborhood: if best model has $k = 35$, check $k = 30 - 40$ yields similar inferences.

Practical Notes II: Picking k

Crudely: in social science, researchers fit ‘enough’ topics until they see what they think they should. E.g. a certain topic—like finance suddenly peels off—so stop there.

→ Check findings are robust in the neighborhood: if best model has $k = 35$, check $k = 30 - 40$ yields similar inferences.

NB: social scientists typically fit far fewer topics than CS, even to same data.

Practical Notes II: Picking k

Crudely: in social science, researchers fit ‘enough’ topics until they see what they think they should. E.g. a certain topic—like finance suddenly peels off—so stop there.

→ Check findings are robust in the neighborhood: if best model has $k = 35$, check $k = 30 - 40$ yields similar inferences.

NB: social scientists typically fit far fewer topics than CS, even to same data.

Picking k , continued...

CS: split into training and test sets.

Picking k , continued...

CS: split into training and test sets. In the **training set**,

Picking k , continued...

CS: split into training and test sets. In the **training set**,

- ① pick some value of k and fit a topic model.

Picking k , continued...

CS: split into training and test sets. In the **training set**,

- ① pick some value of k and fit a topic model.
- ② record value of α (hyperparameter on document specific topic distributions) and word distributions for the topics (the β s)

Picking k , continued...

CS: split into training and test sets. In the **training** set,

- ① pick some value of k and fit a topic model.
- ② record value of α (hyperparameter on document specific topic distributions) and word distributions for the topics (the β s)

We'll write the β s as β , then we want

$$\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\beta, \alpha) = \sum_d \log p(w_d|\beta, \alpha)$$

Picking k , continued...

CS: split into training and test sets. In the **training** set,

- ① pick some value of k and fit a topic model.
- ② record value of α (hyperparameter on document specific topic distributions) and word distributions for the topics (the β s)

We'll write the β s as β , then we want

$$\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\beta, \alpha) = \sum_d \log p(w_d|\beta, \alpha)$$

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

Picking k , continued...

CS: split into training and test sets. In the **training** set,

- ① pick some value of k and fit a topic model.
- ② record value of α (hyperparameter on document specific topic distributions) and word distributions for the topics (the β s)

We'll write the β s as β , then we want

$$\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\beta, \alpha) = \sum_d \log p(w_d|\beta, \alpha)$$

where \mathbf{w} are the words in the **test** set. Higher \mathcal{L} implies better model.

Picking k , continued...

CS: split into training and test sets. In the **training** set,

- ① pick some value of k and fit a topic model.
- ② record value of α (hyperparameter on document specific topic distributions) and word distributions for the topics (the β s)

We'll write the β s as β , then we want

$$\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\beta, \alpha) = \sum_d \log p(w_d|\beta, \alpha)$$

where \mathbf{w} are the words in the **test** set. Higher \mathcal{L} implies better model. Intuition is to calculate likelihood of seeing the test words, given what we know produced the training set.

Picking k , continued...

CS: split into training and test sets. In the **training** set,

- ① pick some value of k and fit a topic model.
- ② record value of α (hyperparameter on document specific topic distributions) and word distributions for the topics (the β s)

We'll write the β s as β , then we want

$$\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\beta, \alpha) = \sum_d \log p(w_d|\beta, \alpha)$$

where \mathbf{w} are the words in the **test** set. Higher \mathcal{L} implies better model. Intuition is to calculate likelihood of seeing the test words, given what we know produced the training set.

Do this for all k .

Picking k , continued...

CS: split into training and test sets. In the **training** set,

- ① pick some value of k and fit a topic model.
- ② record value of α (hyperparameter on document specific topic distributions) and word distributions for the topics (the β s)

We'll write the β s as β , then we want

$$\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\beta, \alpha) = \sum_d \log p(w_d|\beta, \alpha)$$

where \mathbf{w} are the words in the **test** set. Higher \mathcal{L} implies better model. Intuition is to calculate likelihood of seeing the test words, given what we know produced the training set.

Do this for all k .

In practice...

In practice...

Perplexity is popular option

In practice...

Perplexity is popular option

$$\text{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{\text{count of tokens}}\right),$$

In practice...

Perplexity is popular option

$$\text{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{\text{count of tokens}}\right),$$

where lower is better.

In practice...

Perplexity is popular option

$$\text{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{\text{count of tokens}}\right),$$

where lower is better.

In general, $\mathcal{L}(\mathbf{w})$ is intractable,

In practice...

Perplexity is popular option

$$\text{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{\text{count of tokens}}\right),$$

where lower is better.

In general, $\mathcal{L}(\mathbf{w})$ is intractable, but there are ways to approximate it.

In practice...

Perplexity is popular option

$$\text{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{\text{count of tokens}}\right),$$

where lower is better.

In general, $\mathcal{L}(\mathbf{w})$ is intractable, but there are ways to approximate it.

But:

In practice...

Perplexity is popular option

$$\text{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{\text{count of tokens}}\right),$$

where lower is better.

In general, $\mathcal{L}(\mathbf{w})$ is intractable, but there are ways to approximate it.

But: the topic models that hold-out calculations suggest are optimal and not much liked by humans!

In practice...

Perplexity is popular option

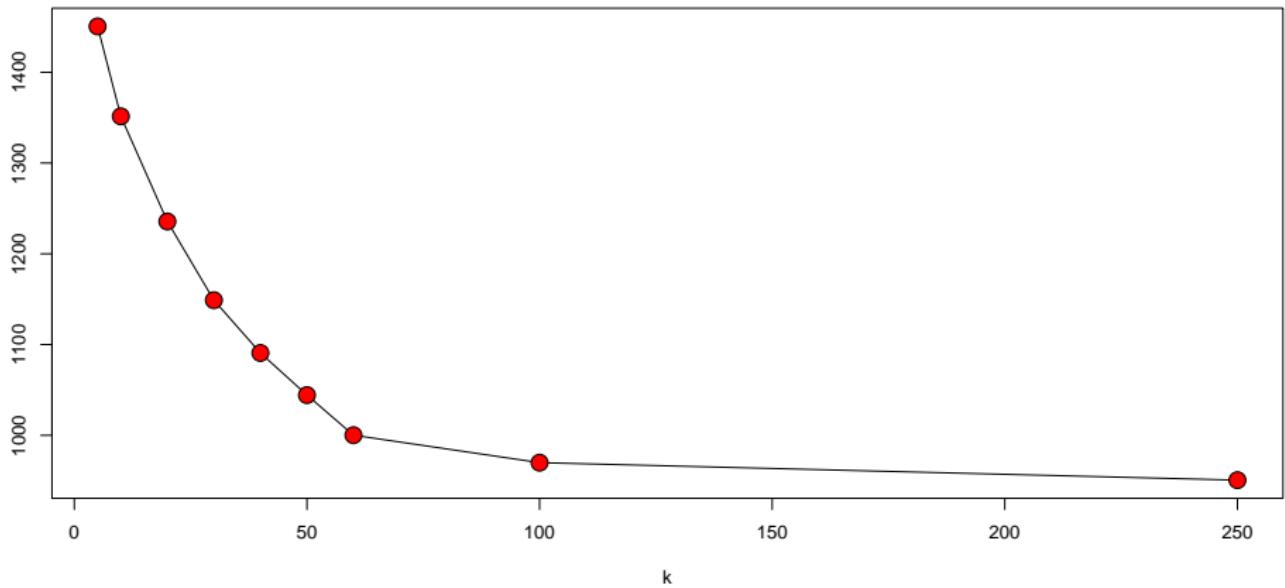
$$\text{perplexity} = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{\text{count of tokens}}\right),$$

where lower is better.

In general, $\mathcal{L}(\mathbf{w})$ is intractable, but there are ways to approximate it.

But: the topic models that hold-out calculations suggest are optimal and not much liked by humans! “Reading Tea Leaves: How Humans Interpret Topic Models” by Chang et al.

Perplexity Likes a Lot of Topics (manifestos)



Pork to Policy (Catalinac, 2016)

Pork to Policy (Catalinac, 2016)



Pork to Policy (Catalinac, 2016)



Japan is a curious IR case:



Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy.



Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area.

Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area. Why?

Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area. Why?

① Rise of China?

Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area. Why?

- ① Rise of China? Need to focus on security.



Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area. Why?

- ① Rise of China? Need to focus on security.
vs.
② Change in Electoral System?

Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area. Why?

- ① Rise of China? Need to focus on security.
vs.
- ② Change in Electoral System? Moved from promising **pork** to having to deliver **policy** as part of Westminster-style polity.

Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area. Why?

- ① Rise of China? Need to focus on security.
vs.
- ② Change in Electoral System? Moved from promising **pork** to having to deliver **policy** as part of Westminster-style polity.

To decide, we need data source that covers all lower house **legislators**

Pork to Policy (Catalinac, 2016)



Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area. Why?

- ① Rise of China? Need to focus on security.
vs.
- ② Change in Electoral System? Moved from promising **pork** to having to deliver **policy** as part of Westminster-style polity.

To decide, we need data source that covers all lower house **legislators** where they set out their **policy priorities** over time.

Pork to Policy (Catalinac, 2016)

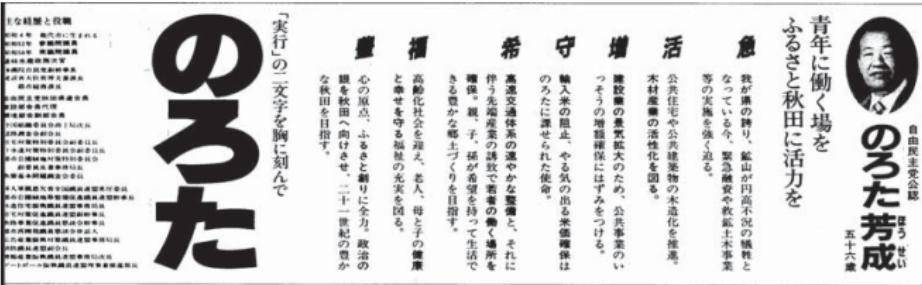


Japan is a curious IR case: wealthy post-war not very interested in foreign policy. Recent times have seen a (re-)emergence in this area. Why?

- ① Rise of China? Need to focus on security.
vs.
- ② Change in Electoral System? Moved from promising **pork** to having to deliver **policy** as part of Westminster-style polity.

To decide, we need data source that covers all lower house **legislators** where they set out their **policy priorities** over time. See if/when they shift priorities.

Manifestos

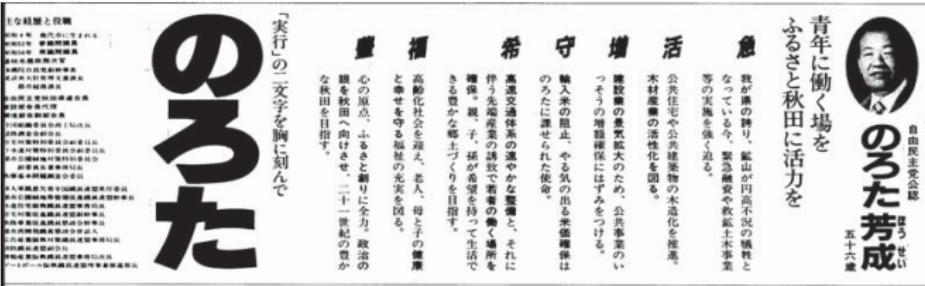


Manifestos



7,497.

Manifestos



7,497. 1986–2009.

Manifestos



7,497. 1986–2009. Standardized form.

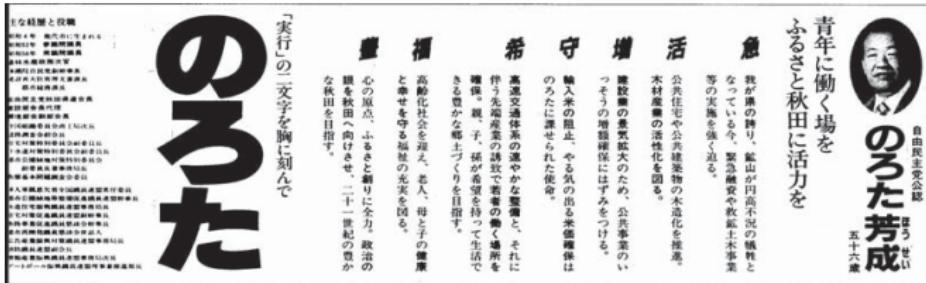
Manifestos



7,497. 1986–2009. Standardized form.

“...instructed to write whatever they want in the form and return it before 5 PM of the first day of the campaign. At least two days before the election, local electoral commissions are required to distribute the forms of all candidates running in the district to all registered voters”

Manifestos

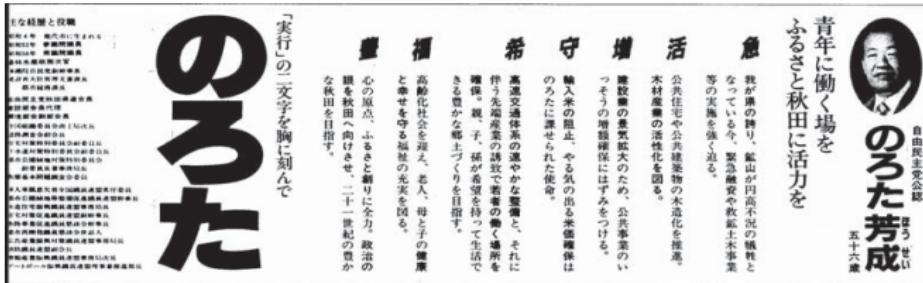


7,497. 1986–2009. Standardized form.

“...instructed to write whatever they want in the form and return it before 5 PM of the first day of the campaign. At least two days before the election, local electoral commissions are required to distribute the forms of all candidates running in the district to all registered voters”

Manifestos were **hand transcribed** from microfilm.

Manifestos



7,497. 1986–2009. Standardized form.

“...instructed to write whatever they want in the form and return it before 5 PM of the first day of the campaign. At least two days before the election, local electoral commissions are required to distribute the forms of all candidates running in the district to all registered voters”

Manifestos were hand transcribed from microfilm. Japanese install of Windows/R used to fit LDA.

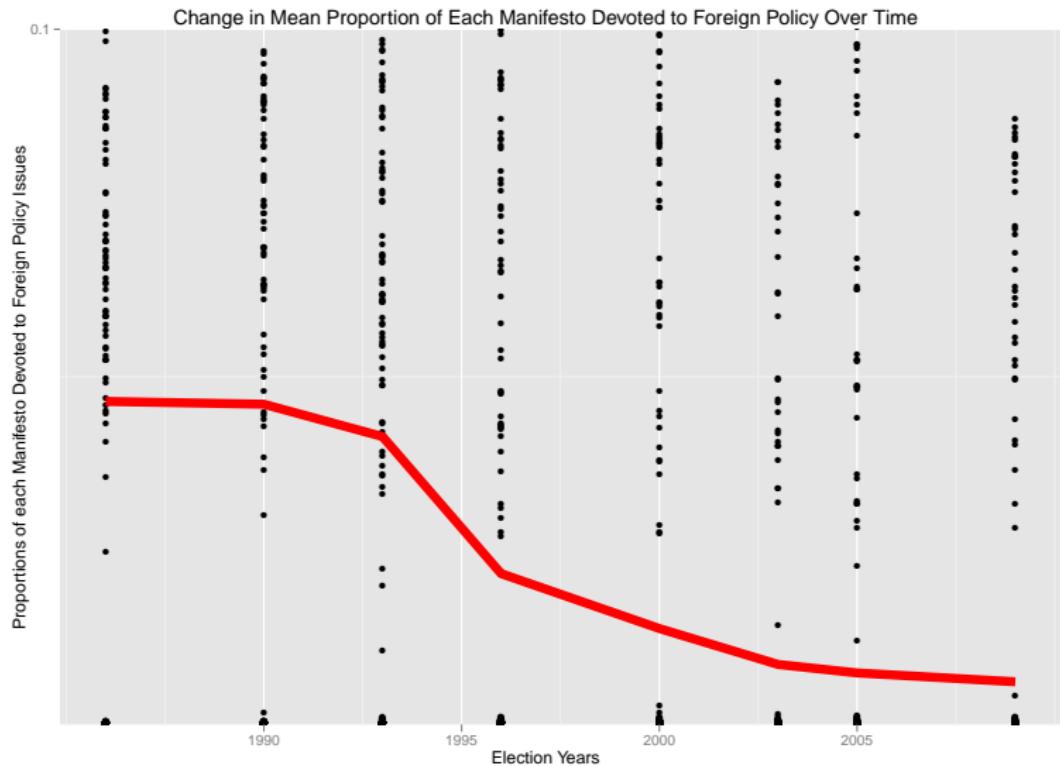
Topic Distribution over Words

Topic Distribution over Words

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
1 改革	年金	推進	区	政治	日本
2 郵政	円	整備	政策	改革	国
3 民営	廃止	図る	地域	国民	外交
4 小泉	改革	つとめる	まち	企業	国家
5 構造	兆	社会	鹿児島	自民党	社会
6 政府	実現	対策	全力	日本	国民
7 官	無駄	振興	選挙	共産党	保障
8 推進	日本	充実	国政	献金	安全
9 民	増税	促進	作り	金権	地域
10 自民党	削減	安定	横浜	党	拉致
11 日本	一元化	確立	対策	選挙	経済
12 制度	政権	企業	中小	禁止	守る
13 民間	子供	実現	発電	憲法	問題
14 年金	地域	中小	推進	腐敗	北朝鮮
15 実現	ひと	育成	エネルギー	団体	教育
16 進める	サラリーマン	制度	企業	区	責任
17 斷行	制度	政治	声	ソ連	力
18 地方	議員	地域	実現	守る	創る
19 止める	金	福祉	活性	平和	安心
20 保障	民主党	事業	自民党	円	目指す
21 財政	年間	改革	地方	反対	誇り
22 作る	一掃	確保	尽くす	真	憲法
23 賛成	郵政	強化	商店	是正	可能
24 社会	道路	教育	いかす	一掃	道
25 国民	交代	施設	全国	悪政	未来
26 公務員	社会保険庁	生活	政党	抜本	ひと
27 力	月額	支援	ひと	定数	再生
28 経済	手当	環境	支援	政党	将来
29 国	談合	発展	経済	金丸	解決
30 安心	吉澤	協議	福祉	改革	其本

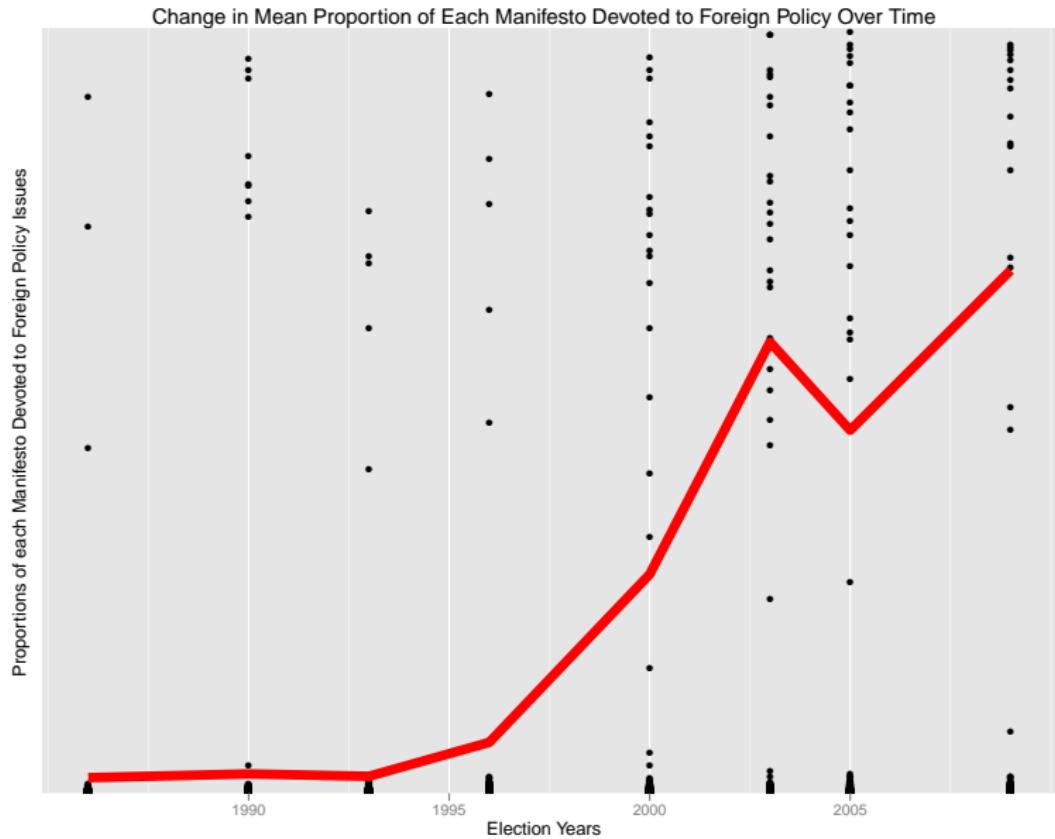
Change in proportion of 'Pork' Topic

Change in proportion of 'Pork' Topic



Change in proportion of 'Foreign Policy' Topic

Change in proportion of 'Foreign Policy' Topic



Special Topics: Structural Topic Model

Structural Topic Model

Structural Topic Model

In general, we have lots of **metadata**:

Structural Topic Model

In general, we have lots of **metadata**: e.g. author covariates, like gender or party membership.

Structural Topic Model

In general, we have lots of **metadata**: e.g. author covariates, like gender or party membership.

But this is non-trivial to include in LDA.

Structural Topic Model

In general, we have lots of **metadata**: e.g. author covariates, like gender or party membership.

But this is non-trivial to include in LDA.

→ STM = LDA + contextual information

Structural Topic Model

In general, we have lots of **metadata**: e.g. author covariates, like gender or party membership.

But this is non-trivial to include in LDA.

→ STM = LDA + contextual information

This allows **more accurate estimation** and

Structural Topic Model

In general, we have lots of **metadata**: e.g. author covariates, like gender or party membership.

But this is non-trivial to include in LDA.

→ STM = LDA + contextual information

This allows **more accurate estimation** and **more interpretable results**.

Structural Topic Model

In general, we have lots of **metadata**: e.g. author covariates, like gender or party membership.

But this is non-trivial to include in LDA.

→ STM = LDA + contextual information

This allows **more accurate estimation** and **more interpretable results**.

Also allows us to ‘test’ hypothesis in more sensible way (though be careful!)

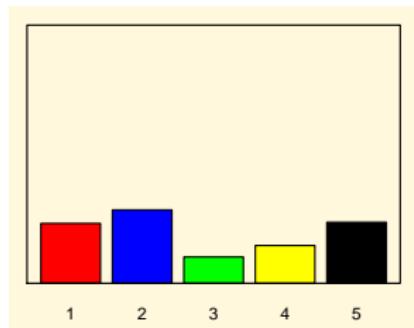
Compare: Per Document Topic Distribution (θ)

Compare: Per Document Topic Distribution (θ)

LDA: each document
has some topic
distribution.

Compare: Per Document Topic Distribution (θ)

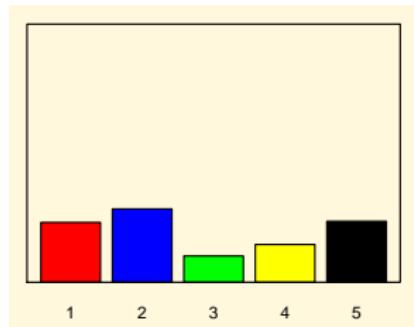
LDA: each document
has some topic
distribution.



Compare: Per Document Topic Distribution (θ)

LDA: each document has some topic distribution.

STM, that topic distribution is a function of the document metadata.

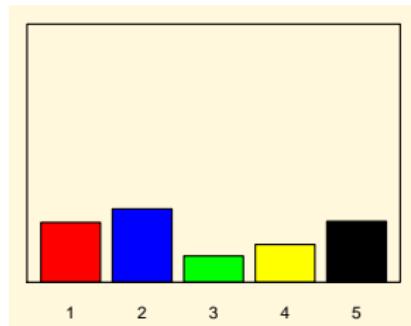


Compare: Per Document Topic Distribution (θ)

LDA: each document has some topic distribution.

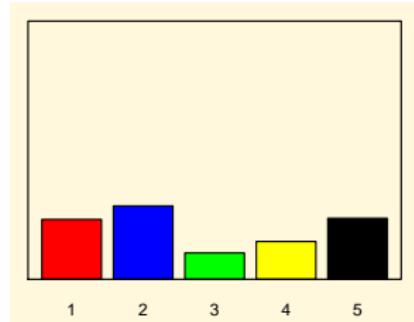
STM, that topic distribution is a function of the document metadata.

e.g. perhaps male author ($X = 0$) documents have different topics relative to female ($X = 1$) author docs.



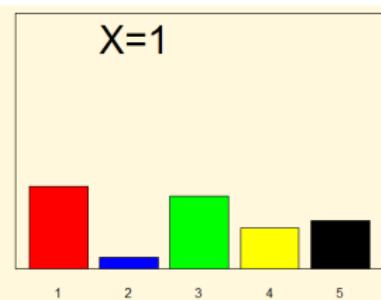
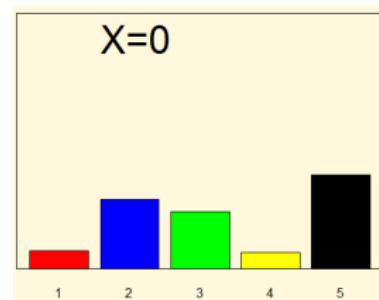
Compare: Per Document Topic Distribution (θ)

LDA: each document has some topic distribution.



STM, that topic distribution is a function of the document metadata.

e.g. perhaps male author ($X = 0$) documents have different topics relative to female ($X = 1$) author docs.



Compare: Per Topic Word Distribution (β)

Compare: Per Topic Word Distribution (β)

LDA: topic ('immigration') has a given distribution over words.

Compare: Per Topic Word Distribution (β)

LDA: topic ('immigration') has a given distribution over words.



Compare: Per Topic Word Distribution (β)

LDA: topic ('immigration') has a given distribution over words.



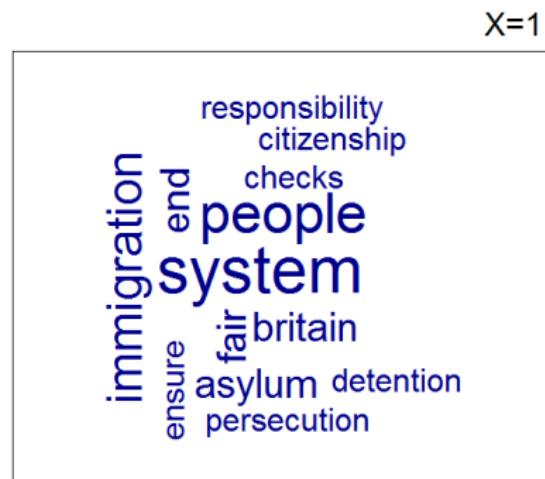
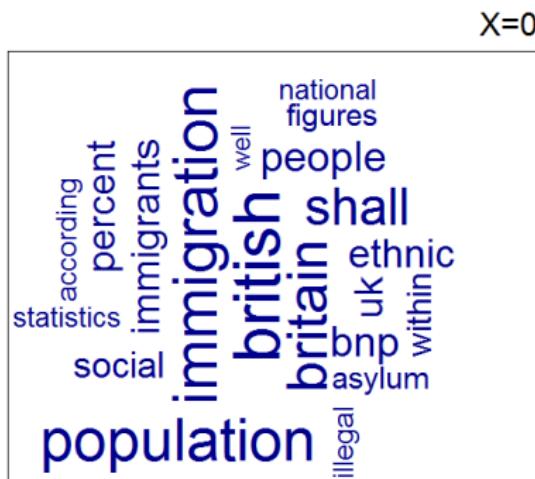
STM: that word distribution is a function of the document metadata.

STM: that word distribution is a function of the document metadata.

e.g. perhaps right parties ($X = 0$) talk about a given topic differently to left ($X = 1$) parties.

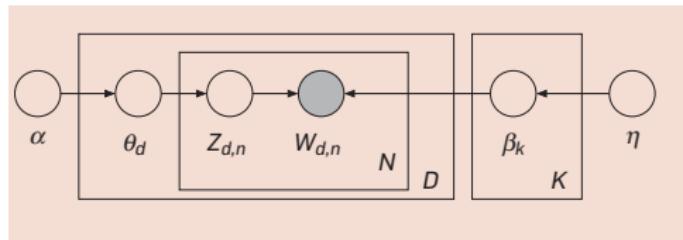
STM: that word distribution is a function of the document metadata.

e.g. perhaps right parties ($X = 0$) talk about a given topic differently to left ($X = 1$) parties.



Compare: Plate Diagram

Compare: Plate Diagram



Compare: Plate Diagram

