

8. Unsupervised Techniques II

DS-GA 1015, Text as Data
Arthur Spirling

March 30, 2021

Where Are We?

Where Are We?



Where Are We?



We've covered the idea of [reducing](#) the data to its 'most important' parts via e.g. SVD

Where Are We?



We've covered the idea of [reducing](#) the data to its 'most important' parts via e.g. SVD

[Now](#) begin to think about comparing models,

Where Are We?



We've covered the idea of **reducing** the data to its 'most important' parts via e.g. SVD

Now begin to think about **comparing models**, and **scaling** positions.

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

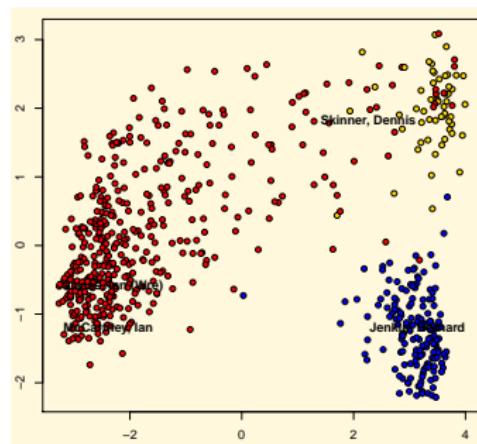
Unsupervised techniques: learning
(hidden or latent) structure in
unlabeled data.

- 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.

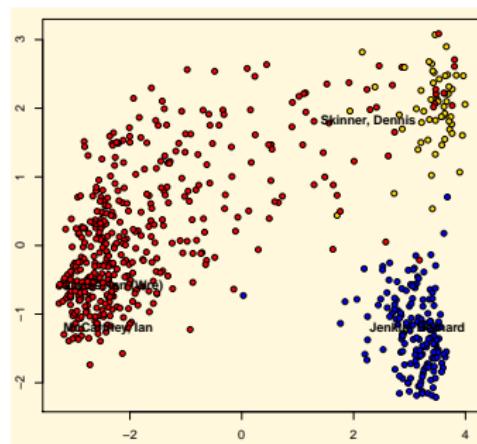
e.g. PCA of legislators's votes: want to see how they are organized—by party? by ideology? by race?



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?

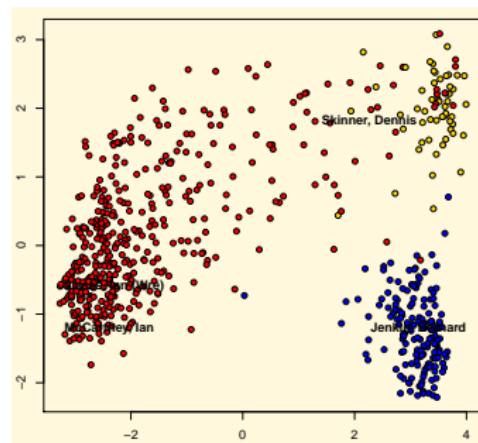


Supervised techniques:

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?

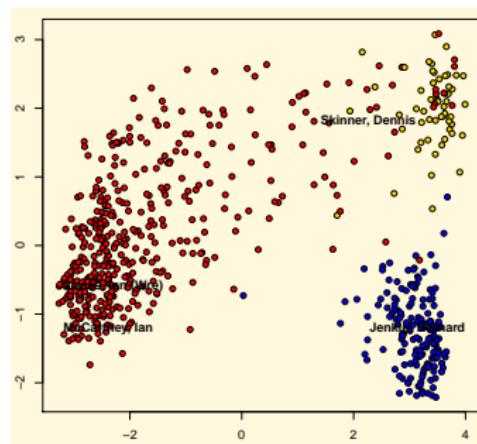


Supervised techniques: learning relationship between inputs and a **labeled** set of outputs.

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?



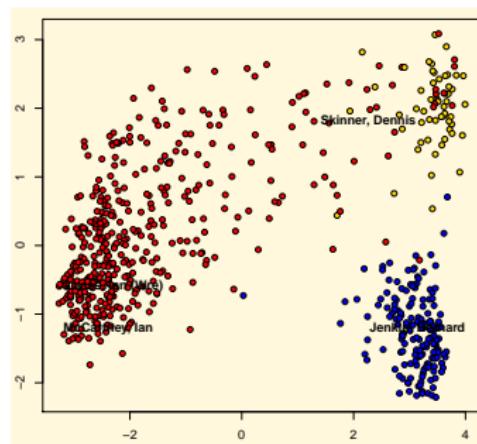
Supervised techniques: learning relationship between inputs and a labeled set of outputs.

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?



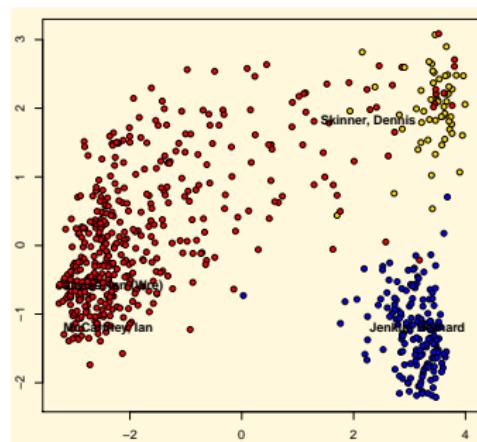
Supervised techniques: learning relationship between inputs and a labeled set of outputs.

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

Back to the House of Commons (1997)...

600 members, 1100 votes.

Back to the House of Commons (1997)...

600 members, 1100 votes.

→ two dimensions is easiest to visualize,

Back to the House of Commons (1997)...

600 members, 1100 votes.

→ two dimensions is easiest to visualize, but not particularly good fit.

Back to the House of Commons (1997)...

600 members, 1100 votes.

→ two dimensions is easiest to visualize, but not particularly good fit.

red points are (so called) Labour party,

Back to the House of Commons (1997)...

600 members, 1100 votes.

→ two dimensions is easiest to visualize, but not particularly good fit.

red points are (so called) Labour party, blue are Conservative (so called) 'opposition',

Back to the House of Commons (1997)...

600 members, 1100 votes.

→ two dimensions is easiest to visualize, but not particularly good fit.

red points are (so called) Labour party, blue are Conservative (so called) 'opposition', and yellow points are (so called) Liberal (so called) Democrats.

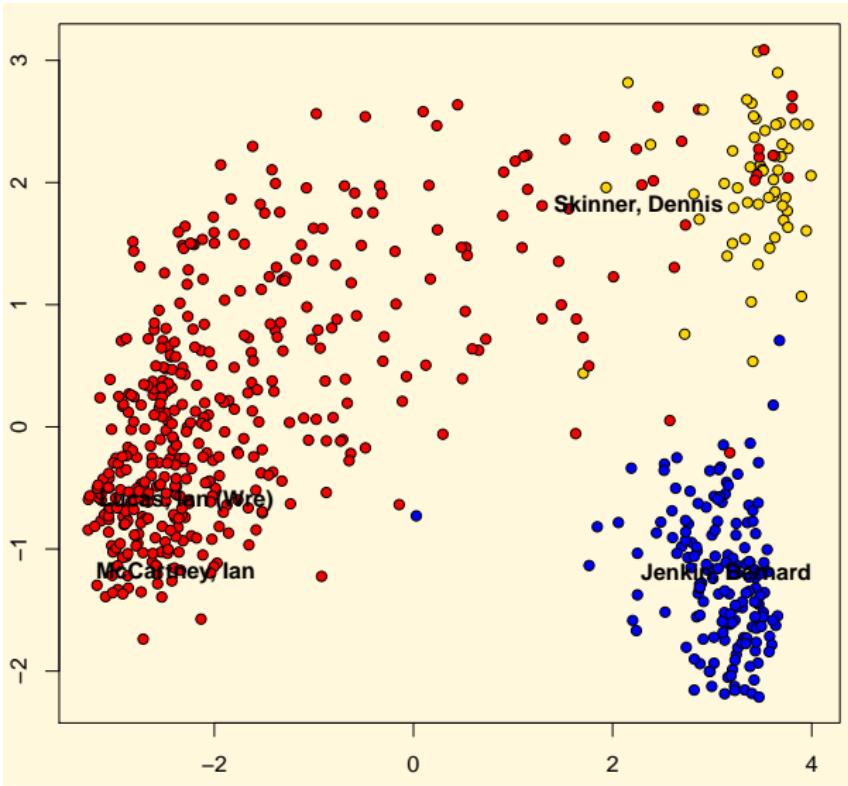
Back to the House of Commons (1997)...

600 members, 1100 votes.

→ two dimensions is easiest to visualize, but not particularly good fit.

red points are (so called) Labour party, blue are Conservative (so called) 'opposition', and yellow points are (so called) Liberal (so called) Democrats.

hmm left winger Dennis Skinner is scored surprisingly close to Conservatives...



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)

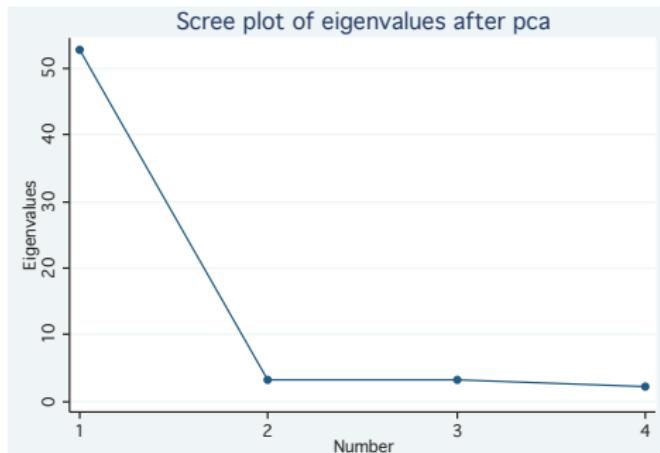
Considers PCA of (preprocessed)
1000-top-vectors for US Senators.

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 (preprocessed)
1000-top-vectors for US Senators.

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



Related Ideas

Related Ideas

Correspondence Analysis: used when data is categorical (e.g. **X** is a table of some kind).

Related Ideas

Correspondence Analysis: used when data is categorical (e.g. **X** is a table of some kind).

Distance methods:

Related Ideas

Correspondence Analysis: used when data is categorical (e.g. **X** is a table of some kind).

Distance methods: display points in lower dimensional space such that proximity in that space represents 'closeness' of observations in original data.

Related Ideas

Correspondence Analysis: used when data is categorical (e.g. **X** is a table of some kind).

Distance methods: display points in lower dimensional space such that proximity in that space represents 'closeness' of observations in original data. Obvious case is **Euclidean distance**.

Related Ideas

Correspondence Analysis: used when data is categorical (e.g. \mathbf{X} is a table of some kind).

Distance methods: display points in lower dimensional space such that proximity in that space represents 'closeness' of observations in original data. Obvious case is **Euclidean distance**.

→ gives rise to square similarity/dissimilarity matrix or **distance matrix**.

Related Ideas

Correspondence Analysis: used when data is categorical (e.g. \mathbf{X} is a table of some kind).

Distance methods: display points in lower dimensional space such that proximity in that space represents 'closeness' of observations in original data. Obvious case is **Euclidean distance**.

→ gives rise to square similarity/dissimilarity matrix or **distance matrix**.

If use eigenvectors ('characteristic' vectors of the matrix) to represent the points, this is **multidimensional scaling**.

Related Ideas

Correspondence Analysis: used when data is categorical (e.g. \mathbf{X} is a table of some kind).

Distance methods: display points in lower dimensional space such that proximity in that space represents 'closeness' of observations in original data. Obvious case is **Euclidean distance**.

→ gives rise to square similarity/dissimilarity matrix or **distance matrix**.

If use eigenvectors ('characteristic' vectors of the matrix) to represent the points, this is **multidimensional scaling**. Simplest version is (linear) **principal coordinates analysis**

Related Ideas

Correspondence Analysis: used when data is categorical (e.g. \mathbf{X} is a table of some kind).

Distance methods: display points in lower dimensional space such that proximity in that space represents 'closeness' of observations in original data. Obvious case is **Euclidean distance**.

→ gives rise to square similarity/dissimilarity matrix or **distance matrix**.

If use eigenvectors ('characteristic' vectors of the matrix) to represent the points, this is **multidimensional scaling**. Simplest version is (linear) **principal coordinates analysis**

Can put more emphasis on representing small distances accurately via e.g. **Sammon** scaling.

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

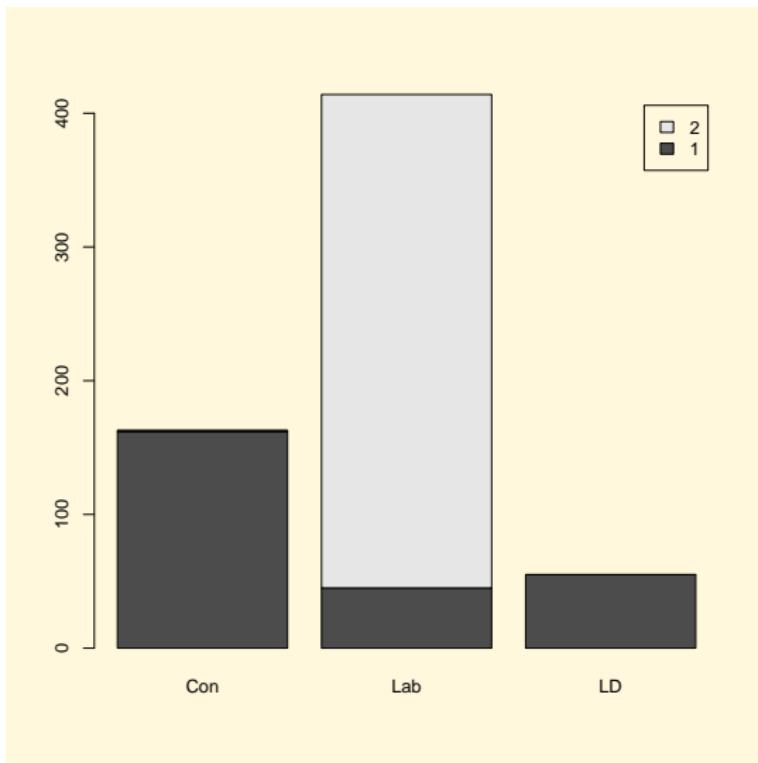
Clustering: look for ‘groups’ in data explicitly.

Partition methods are most common:

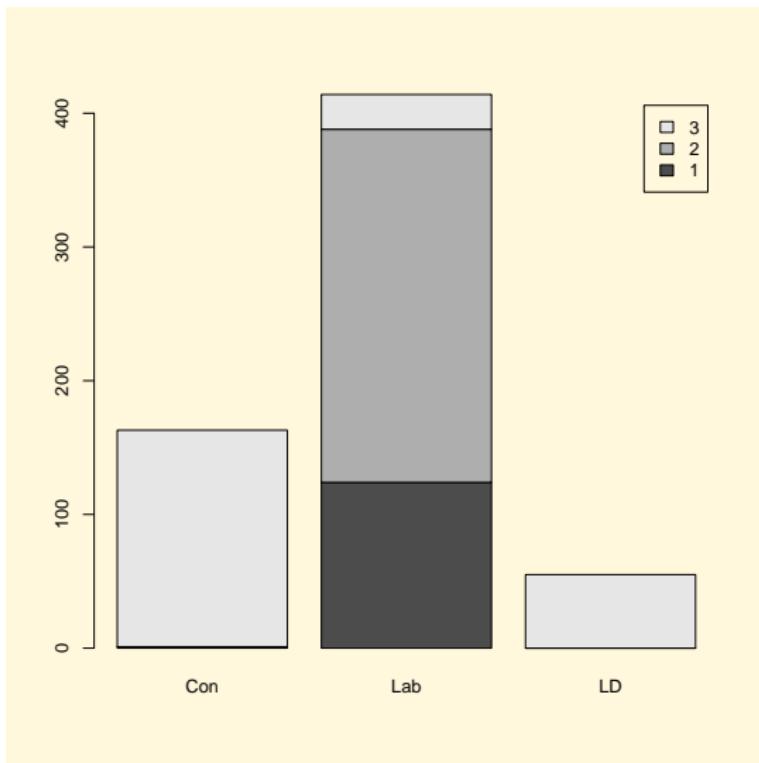
- Include *K-means*, for which one pre-specifies cluster number, and algorithm puts observations into clusters which minimize within cluster sum of squares
- so pick s such that $\sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$ is minimized where μ_i is (vector) mean of points in cluster S_i .
- observations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

k-means on Commons Roll Calls

k-means on Commons Roll Calls



k-means on Commons Roll Calls



Hierarchical Methods

Hierarchical Methods

Successively aggregate groups of observations.

Hierarchical Methods

Successively aggregate groups of observations.

Can be **agglomerative/bottom-up** in sense that everything starts in own cluster and then groups are formed by putting observations together

Hierarchical Methods

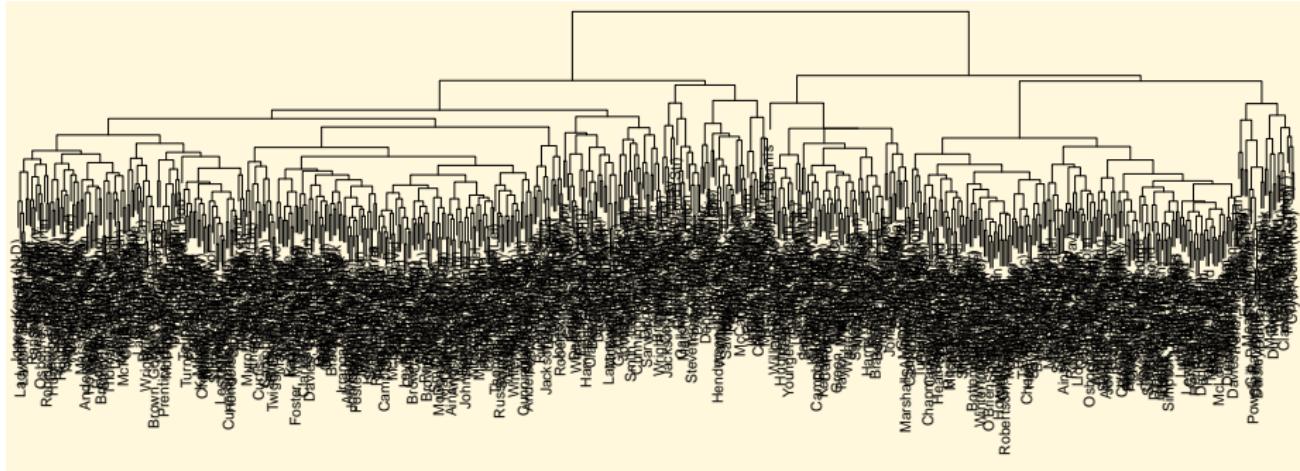
Successively aggregate groups of observations.

Can be **agglomerative/bottom-up** in sense that everything starts in own cluster and then groups are formed by putting observations together

Or **Divisive/top down** in sense that everything starts in same cluster and then splits are performed (typically on one feature) to form clusters.

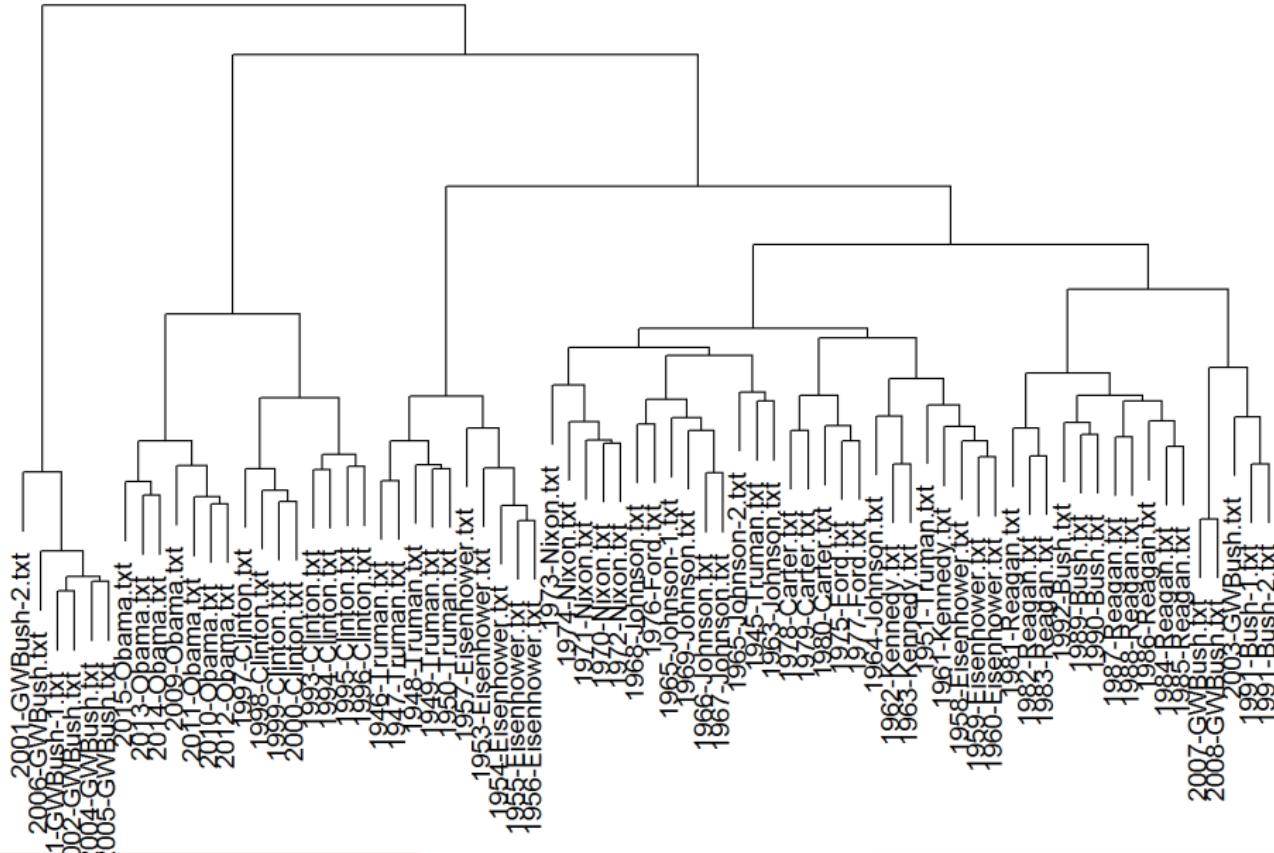
Hierarchical: Commons

Hierarchical: Commons



Hierarchical: SOTU (Frank Evans, dzone.com)

Hierarchical: SOTU (Frank Evans, dzone.com)



Notes

0

Notes

Gaussian assumptions probably off-base for many text problems

Notes

Gaussian assumptions probably off-base for many text problems

$n > p$ **not** met in many text examples so cannot always apply techniques 'off the shelf'

Notes

Gaussian assumptions probably off-base for many text problems

$n > p$ not met in many text examples so cannot always apply techniques 'off the shelf' and missing data not trivial to handle

Notes

Gaussian assumptions probably off-base for many text problems

$n > p$ not met in many text examples so cannot always apply techniques 'off the shelf' and missing data not trivial to handle

Hard to compare across specifications:

Notes

Gaussian assumptions probably off-base for many text problems

$n > p$ not met in many text examples so cannot always apply techniques 'off the shelf' and missing data not trivial to handle

Hard to compare across specifications: e.g. $k = 2$, $k = 3$

Notes

Gaussian assumptions probably off-base for many text problems

$n > p$ not met in many text examples so cannot always apply techniques 'off the shelf' and missing data not trivial to handle

Hard to compare across specifications: e.g. $k = 2$, $k = 3$

No underlying model of human behavior/text generation gives rise to these techniques

Notes

Gaussian assumptions probably off-base for many text problems

$n > p$ not met in many text examples so cannot always apply techniques 'off the shelf' and missing data not trivial to handle

Hard to compare across specifications: e.g. $k = 2$, $k = 3$

No underlying model of human behavior/text generation gives rise to these techniques

“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'.

Steps

Steps

1 standard **preprocessing** of texts,

Steps

- 1 standard **preprocessing** of texts, throwing out very rare and very common terms.

Steps

- 1 standard **preprocessing** of texts, throwing out very rare and very common terms.
- 2 apply very large number of **clustering algorithms**, with multiple specifications of each.

Steps

- 1 standard **preprocessing** of texts, throwing out very rare and very common terms.
- 2 apply very large number of **clustering algorithms**, with multiple specifications of each.
- 3 calculate **distance** between J clusterings as function of number of pairs of documents not placed together in same cluster.

Steps

- 1 standard **preprocessing** of texts, throwing out very rare and very common terms.
- 2 apply very large number of **clustering algorithms**, with multiple specifications of each.
- 3 calculate **distance** between J clusterings as function of number of pairs of documents not placed together in same cluster.
- 4 project this $J \times J$ clustering matrix down to 2D Euclidean space using **Sammon MDS** to preserve small distances better (than large ones)

Steps

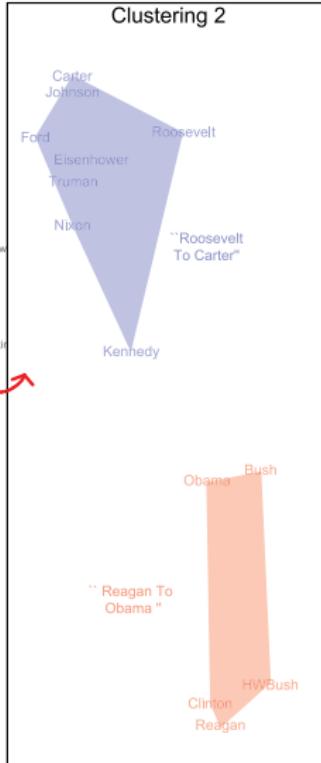
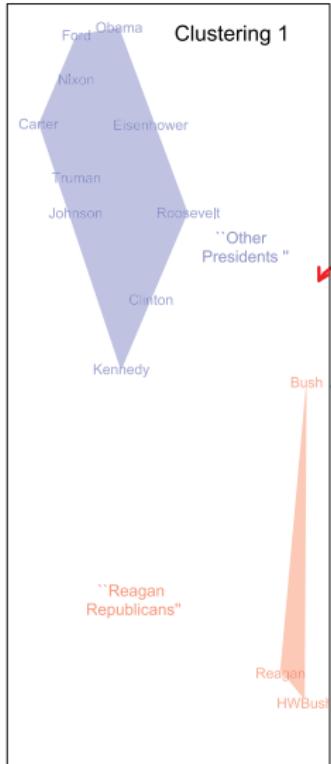
- 1 standard **preprocessing** of texts, throwing out very rare and very common terms.
- 2 apply very large number of **clustering algorithms**, with multiple specifications of each.
- 3 calculate **distance** between J clusterings as function of number of pairs of documents not placed together in same cluster.
- 4 project this $J \times J$ clustering matrix down to 2D Euclidean space using **Sammon MDS** to preserve small distances better (than large ones)
- 5 allow for **local cluster ensemble** which is (a new) clustering composed of combination of clusterings that are nearby any given point in 2D space.

Steps

- 1 standard **preprocessing** of texts, throwing out very rare and very common terms.
- 2 apply very large number of **clustering algorithms**, with multiple specifications of each.
- 3 calculate **distance** between J clusterings as function of number of pairs of documents not placed together in same cluster.
- 4 project this $J \times J$ clustering matrix down to 2D Euclidean space using **Sammon MDS** to preserve small distances better (than large ones)
- 5 allow for **local cluster ensemble** which is (a new) clustering composed of combination of clusterings that are nearby any given point in 2D space.
- 6 visualize for users.

Example: Biographies of Presidents

Example: Biographies of Presidents



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:

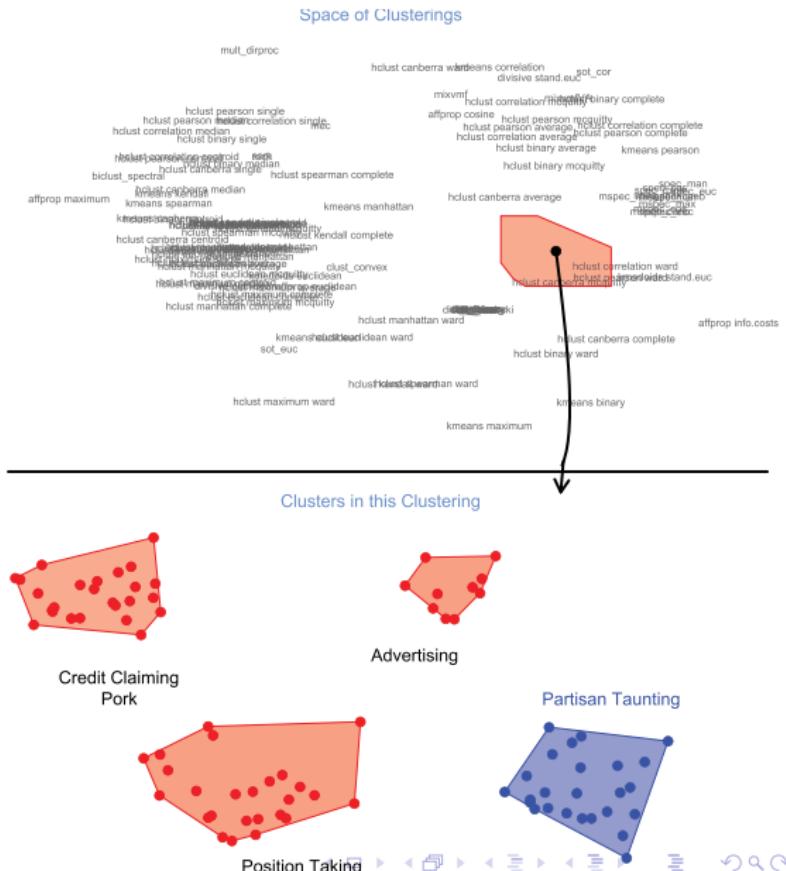
- 1 **Cluster Quality**: randomly draw pairs of documents from **same** cluster and **different** clusters, and ask **human coders** how closely related they are.
- 2 **Discovery Quality**: show scholars the cluster space and see if it improves their understanding of own data

Discovery of Partisan Taunting in Press Releases

Discovery of Partisan Taunting in Press Releases



Discovery of Partisan Taunting in Press Releases



Latent Semantic Analysis

Latent Semantic Analysis (Deerwester et al, 1988)

Latent Semantic Analysis (Deerwester et al, 1988)

Method

Latent Semantic Analysis (Deerwester et al, 1988)

Method (a theory?) for representing **meaning** of words

Latent Semantic Analysis (Deerwester et al, 1988)

Method (a theory?) for representing **meaning** of words

Assumes . . .

Latent Semantic Analysis (Deerwester et al, 1988)

Method (a theory?) for representing **meaning** of words

Assumes . . .

that set of **contexts**

Latent Semantic Analysis (Deerwester et al, 1988)

Method (a theory?) for representing **meaning** of words

Assumes . . .

that set of **contexts** in which a word does/does not appear defines its relationship to other words

Latent Semantic Analysis (Deerwester et al, 1988)

Method (a theory?) for representing **meaning** of words

Assumes . . .

that set of **contexts** in which a word does/does not appear defines its relationship to other words
and that this relationship defines its **meaning**

Latent Semantic Analysis (Deerwester et al, 1988)

Method (a theory?) for representing **meaning** of words

Assumes . . .

that set of **contexts** in which a word does/does not appear defines its relationship to other words
and that this relationship defines its **meaning**

NB very simple to calculate (uses **SVD** of TDM)

Latent Semantic Analysis (Deerwester et al, 1988)

Method (a theory?) for representing **meaning** of words

Assumes . . .

that set of **contexts** in which a word does/does not appear defines its relationship to other words
and that this relationship defines its **meaning**

NB very simple to calculate (uses **SVD** of TDM)

→ which might make us nervous about LSA as

Latent Semantic Analysis (Deerwester et al, 1988)

Method (a theory?) for representing **meaning** of words

Assumes . . .

that set of **contexts** in which a word does/does not appear defines its relationship to other words
and that this relationship defines its **meaning**

NB very simple to calculate (uses **SVD** of TDM)

→ which might make us nervous about LSA as

"a model of the computational processes . . . underlying . . . the acquisition and utilization of knowledge"

Process

Process

Not supervised,

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) **TDM** for some set of texts:

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) **TDM** for some set of texts:
terms are rows, **texts** are columns.

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts:
terms are rows, texts are columns. Cell entries are transformed via product of...

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts:
terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function:

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts: terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts:
terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts:
terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$

(infinite variety)

global weight function: e.g.

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts:
terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts:
terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$ (infinite variety:

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts:
terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$ (infinite variety: this one is '1 + entropy' in some lists)

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts:
terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$ (infinite variety: this one is '1 + entropy' in some lists)

NB take $0 \times \log(0)$ to be zero

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts: terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$ (infinite variety: this one is '1 + entropy' in some lists)

NB take $0 \times \log(0)$ to be zero
where

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts: terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$ (infinite variety: this one is '1 + entropy' in some lists)

NB take $0 \times \log(0)$ to be zero
where n is simply total number of documents;

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts: terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$ (infinite variety: this one is '1 + entropy' in some lists)

NB take $0 \times \log(0)$ to be zero

where n is simply total number of documents; $p_{ij} = \frac{tf_{ij}}{gf_i}$.

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts: terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$ (infinite variety: this one is '1 + entropy' in some lists)

NB take $0 \times \log(0)$ to be zero

where n is simply total number of documents; $p_{ij} = \frac{tf_{ij}}{gf_i}$. Note that gf_i is simply total number of times term i appears in corpus (i.e. over all docs).

Process

Not supervised, in sense that there is no *a priori* dictionary, or grammar, or morphology etc

Begin with (unstemmed, unstopped) TDM for some set of texts: terms are rows, texts are columns. Cell entries are transformed via product of...

local weight function: e.g. $\log(tf_{ij} + 1)$
(infinite variety)

global weight function: e.g. $1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$ (infinite variety: this one is '1 + entropy' in some lists)

NB take $0 \times \log(0)$ to be zero

where n is simply total number of documents; $p_{ij} = \frac{tf_{ij}}{gf_i}$. Note that gf_i is simply total number of times term i appears in corpus (i.e. over all docs).

So...

So...

e.g. a row of some (3 text) corpus TDM is:

So...

e.g. a row of some (3 text) corpus TDM is:

term	doc1	doc2	doc3
dog	1	3	2

So...

e.g. a row of some (3 text) corpus TDM is:

term	doc1	doc2	doc3
dog	1	3	2

- applying LWF gives:

So...

e.g. a row of some (3 text) corpus TDM is:

term	doc1	doc2	doc3
dog	1	3	2

- applying LWF gives: $c(0.69, 1.39, 1.10)$

So...

e.g. a row of some (3 text) corpus TDM is:

term	doc1	doc2	doc3
dog	1	3	2

- applying LWF gives: $c(0.69, 1.39, 1.10)$
- applying GWF gives:

So...

e.g. a row of some (3 text) corpus TDM is:

term	doc1	doc2	doc3
dog	1	3	2

- applying LWF gives: $c(0.69, 1.39, 1.10)$
- applying GWF gives: $p_{i,1} = \frac{1}{6}$ and thus $1/6 \log(1/6)$; $p_{i,2} = \frac{3}{6}$ and thus $3/6 \log(3/6)$; $p_{i,3} = \frac{2}{6}$ and thus $2/6 \log(2/6)$.

So...

e.g. a row of some (3 text) corpus TDM is:

term	doc1	doc2	doc3
dog	1	3	2

- applying LWF gives: $c(0.69, 1.39, 1.10)$
- applying GWF gives: $p_{i,1} = \frac{1}{6}$ and thus $\frac{1}{6} \log(1/6)$; $p_{i,2} = \frac{3}{6}$ and thus $\frac{3}{6} \log(3/6)$; $p_{i,3} = \frac{2}{6}$ and thus $\frac{2}{6} \log(2/6)$.
- we then have $1 + \left(\frac{\frac{1}{6} \log(1/6) + \frac{3}{6} \log(3/6) + \frac{2}{6} \log(2/6)}{\log(3)} \right) = 0.079$

So...

e.g. a row of some (3 text) corpus TDM is:

term	doc1	doc2	doc3
dog	1	3	2

- applying LWF gives: $c(0.69, 1.39, 1.10)$
 - applying GWF gives: $p_{i,1} = \frac{1}{6}$ and thus $1/6 \log(1/6)$; $p_{i,2} = \frac{3}{6}$ and thus $3/6 \log(3/6)$; $p_{i,3} = \frac{2}{6}$ and thus $2/6 \log(2/6)$.
 - we then have $1 + \left(\frac{1/6 \log(1/6) + 3/6 \log(3/6) + 2/6 \log(2/6)}{\log(3)} \right) = 0.079$
- ... which we multiply by the LWF to give:

So...

e.g. a row of some (3 text) corpus TDM is:

term	doc1	doc2	doc3
dog	1	3	2

- applying LWF gives: $c(0.69, 1.39, 1.10)$
 - applying GWF gives: $p_{i,1} = \frac{1}{6}$ and thus $\frac{1}{6} \log(1/6)$; $p_{i,2} = \frac{3}{6}$ and thus $\frac{3}{6} \log(3/6)$; $p_{i,3} = \frac{2}{6}$ and thus $\frac{2}{6} \log(2/6)$.
 - we then have $1 + \left(\frac{\frac{1}{6} \log(1/6) + \frac{3}{6} \log(3/6) + \frac{2}{6} \log(2/6)}{\log(3)} \right) = 0.079$
- ... which we multiply by the LWF to give:

term	doc1	doc2	doc3
dog	0.055	0.110	0.087

Decomposing

Decomposing

We do the above for **every term**

Decomposing

We do the above for **every term**

Now need to produce a **lower dimensional representation** of TDM

Decomposing

We do the above for **every term**

Now need to produce a **lower dimensional representation** of TDM
Use **singular value decomposition**, and get three things:

Decomposing

We do the above for **every term**

Now need to produce a **lower dimensional representation** of TDM

Use **singular value decomposition**, and get three things:

- 1 Matrix corresponding to **terms** as vectors of (orthogonal) factor values
(analogous to 'score' for each observation, i)

Decomposing

We do the above for **every term**

Now need to produce a **lower dimensional representation** of TDM

Use **singular value decomposition**, and get three things:

- 1 Matrix corresponding to **terms** as vectors of (orthogonal) factor values (analogous to 'score' for each observation, i)
- 2 Matrix corresponding to **documents** as vectors of (orthogonal) factor values

Decomposing

We do the above for **every term**

Now need to produce a **lower dimensional representation** of TDM

Use **singular value decomposition**, and get three things:

- 1 Matrix corresponding to **terms** as vectors of (orthogonal) factor values (analogous to 'score' for each observation, i)
- 2 Matrix corresponding to **documents** as vectors of (orthogonal) factor values
- 3 Matrix of scaling values to ensure that multiplying these matrices reconstructs **TDM**

Decomposing

We do the above for **every term**

Now need to produce a **lower dimensional representation** of TDM

Use **singular value decomposition**, and get three things:

- 1 Matrix corresponding to **terms** as vectors of (orthogonal) factor values (analogous to 'score' for each observation, i)
- 2 Matrix corresponding to **documents** as vectors of (orthogonal) factor values
- 3 Matrix of scaling values to ensure that multiplying these matrices reconstructs **TDM**

Idea look at **recomposition** of the TDM based on e.g. first **two dimensions** of the sub-matrices:

Decomposing

We do the above for **every term**

Now need to produce a **lower dimensional representation** of TDM

Use **singular value decomposition**, and get three things:

- 1 Matrix corresponding to **terms** as vectors of (orthogonal) factor values (analogous to 'score' for each observation, i)
- 2 Matrix corresponding to **documents** as vectors of (orthogonal) factor values
- 3 Matrix of scaling values to ensure that multiplying these matrices reconstructs **TDM**

Idea look at **recomposition** of the TDM based on e.g. first **two dimensions** of the sub-matrices:

→ represent the TDM as the mix that each word and each document represents *in terms of* (abstract) concept 1 and (abstract) concept 2.

An Example

An Example



An Example

79 (not many for LSA!)



An Example

79 (not many for LSA!) State of the Union Speeches,



An Example

79 (not many for LSA!) State of the Union Speeches, between 1934 and 2009:



An Example



79 (not many for LSA!) State of the Union Speeches, between 1934 and 2009: 16878 (stemmed, de-punctuated) terms

An Example



79 (not many for LSA!) State of the Union Speeches, between 1934 and 2009: 16878 (stemmed, de-punctuated) terms

Consider **five dimensional** representation

An Example



79 (not many for LSA!) State of the Union Speeches, between 1934 and 2009: 16878 (stemmed, de-punctuated) terms



Consider **five dimensional** representation (that is, reconstruct TDM as mix of five concepts)

An Example



79 (not many for LSA!) State of the Union Speeches, between 1934 and 2009: 16878 (stemmed, de-punctuated) terms

Consider **five dimensional** representation (that is, reconstruct TDM as mix of five concepts—probably over-fitting)

An Example



79 (not many for LSA!) State of the Union Speeches, between 1934 and 2009: 16878 (stemmed, de-punctuated) terms

Consider **five dimensional** representation (that is, reconstruct TDM as mix of five concepts—probably over-fitting)

Q1 What are these documents about?

An Example



79 (not many for LSA!) State of the Union Speeches, between 1934 and 2009: 16878 (stemmed, de-punctuated) terms

Consider **five dimensional** representation (that is, reconstruct TDM as mix of five concepts—probably over-fitting)

- Q1** What are these documents about? What do they have as their ‘highest’ (weighted) words?

An Example



79 (not many for LSA!) State of the Union Speeches, between 1934 and 2009: 16878 (stemmed, de-punctuated) terms

Consider **five dimensional** representation (that is, reconstruct TDM as mix of five concepts—probably over-fitting)

- Q1** What are these documents about? What do they have as their ‘highest’ (weighted) words?

- Q2** How are terms related? What words are closely associated conceptually?

Q1

0

Q1

	1942	1985	2002
original	war	freedom	america
	world	tax	security
	united	american	world
	people	time	american
	forces	growth	terror

Q1

	1942	1985	2002
original	war	freedom	america
	world	tax	security
	united	american	world
	people	time	american
	forces	growth	terror
transformed	1944	dollars	iraq
	japanese	tonight	iraqi
	war	we've	terrorists
	1942	million	qaida
	french	thats	terror
	germans	war	terrorist

Q2

Q2



Q2



words

original | transformed



Q2



words	original	transformed
communist, zarqawi	-0.08	-0.28

Q2



words	original	transformed
communist, zarqawi	-0.08	-0.28
america, freedom	0.06	0.41

Q2



words	original	transformed
communist, zarqawi	-0.08	-0.28
america, freedom	0.06	0.41
camp, david	0.56	0.57

Q2



words	original	transformed
communist, zarqawi	-0.08	-0.28
america, freedom	0.06	0.41
camp, david	0.56	0.57
inflation, unemployment	0.59	0.80

Q2



words	original	transformed
communist, zarqawi	-0.08	-0.28
america, freedom	0.06	0.41
camp, david	0.56	0.57
inflation, unemployment	0.59	0.80



Notes on LSA (LSI)

Notes on LSA (LSI)

Somewhat out of style...

Notes on LSA (LSI)

Somewhat out of style...

- 1 Dimensions not trivial to interpret (as with many PCA style techniques)

Notes on LSA (LSI)

Somewhat out of style...

- 1 Dimensions not trivial to interpret (as with many PCA style techniques)
 - rise of Latent Dirichlet Allocation and topic models.

Notes on LSA (LSI)

Somewhat out of style...

- 1 Dimensions not trivial to **interpret** (as with many PCA style techniques)
→ rise of Latent Dirichlet Allocation and topic models.
- 2 Problems with **polysemy** (multiple meanings of words):

Notes on LSA (LSI)

Somewhat out of style...

- 1 Dimensions not trivial to **interpret** (as with many PCA style techniques)
→ rise of Latent Dirichlet Allocation and topic models.
- 2 Problems with **polysemy** (multiple meanings of words): in LSA,

Notes on LSA (LSI)

Somewhat out of style...

- 1 Dimensions not trivial to **interpret** (as with many PCA style techniques)
→ rise of Latent Dirichlet Allocation and topic models.
- 2 Problems with **polysemy** (multiple meanings of words): in LSA, BOW means word context not accounted for.

Notes on LSA (LSI)

Somewhat out of style...

- 1 Dimensions not trivial to **interpret** (as with many PCA style techniques)
 - rise of Latent Dirichlet Allocation and topic models.
- 2 Problems with **polysemy** (multiple meanings of words): in LSA, BOW means word context not accounted for.
 - rise of **embedding** algorithms that learn meaning/context taking into account context of adjoining tokens.

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



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...

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



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)

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

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")

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**:

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.

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.
→ surely false,

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.
→ surely false, but convenient starting point.

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.
→ 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**:

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.
→ 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

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.
→ 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

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

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

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

0

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: $\Phi(\beta_j^T \mathbf{x}_i - \alpha_j)$.

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.

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.

btw can use parametric bootstrap for uncertainty estimates.

Solution III: More formally...

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

E Step:

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

E Step: use the expected value of *as if* known set of parameters to get a log-likelihood that can be evaluated

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

E Step: use the expected value of *as if* known set of parameters to get a log-likelihood that can be evaluated

M Step:

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

E Step: use the expected value of *as if* known set of parameters to get a log-likelihood that can be evaluated

M Step: maximize that **log-likelihood** by conditioning on that expectation, to get new set of values for *as if* unknown (other) set of parameters.

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

E Step: use the expected value of *as if* known set of parameters to get a log-likelihood that can be evaluated

M Step: maximize that **log-likelihood** by conditioning on that expectation, to get new set of values for *as if* unknown (other) set of parameters.

Iterate until log-likelihood changes very little between iterations:
convergence.

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

E Step: use the expected value of *as if* known set of parameters to get a log-likelihood that can be evaluated

M Step: maximize that **log-likelihood** by conditioning on that expectation, to get new set of values for *as if* unknown (other) set of parameters.

Iterate until log-likelihood changes very little between iterations:
convergence.

As in many such latent variable problems,

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

E Step: use the expected value of *as if* known set of parameters to get a log-likelihood that can be evaluated

M Step: maximize that **log-likelihood** by conditioning on that expectation, to get new set of values for *as if* unknown (other) set of parameters.

Iterate until log-likelihood changes very little between iterations: convergence.

As in many such latent variable problems, must have **identification restrictions** because otherwise too many possible solutions to equation.

Solution III: More formally...

Can think about the problem as being about ‘missing data’

Use **Expectation Maximization (EM)** algorithm (Dempster et al, 1977).

E Step: use the expected value of *as if* known set of parameters to get a log-likelihood that can be evaluated

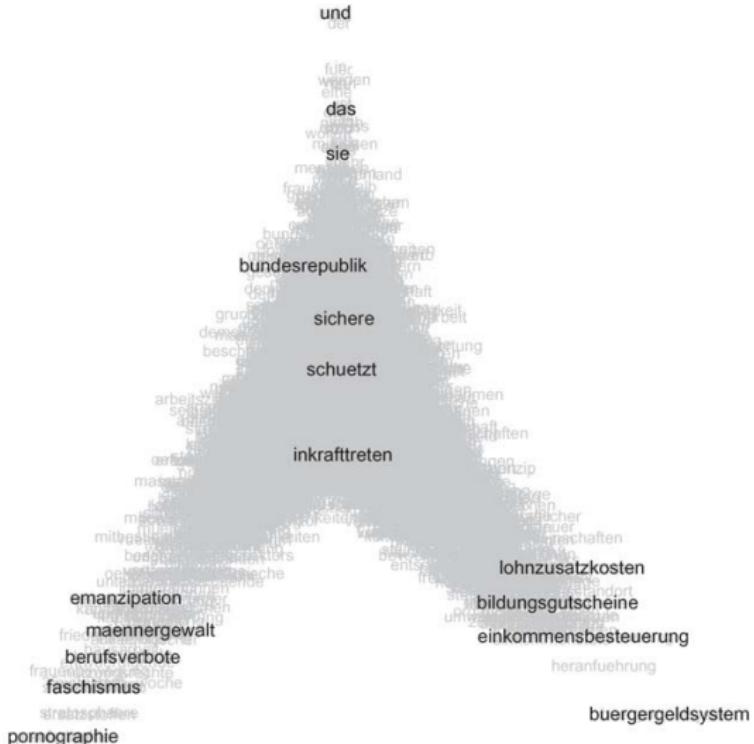
M Step: maximize that **log-likelihood** by conditioning on that expectation, to get new set of values for *as if* unknown (other) set of parameters.

Iterate until log-likelihood changes very little between iterations: convergence.

As in many such latent variable problems, must have **identification restrictions** because otherwise too many possible solutions to equation. So, set $\alpha_1 = 0$ and set party positions to have mean 0 and $sd = 1$.

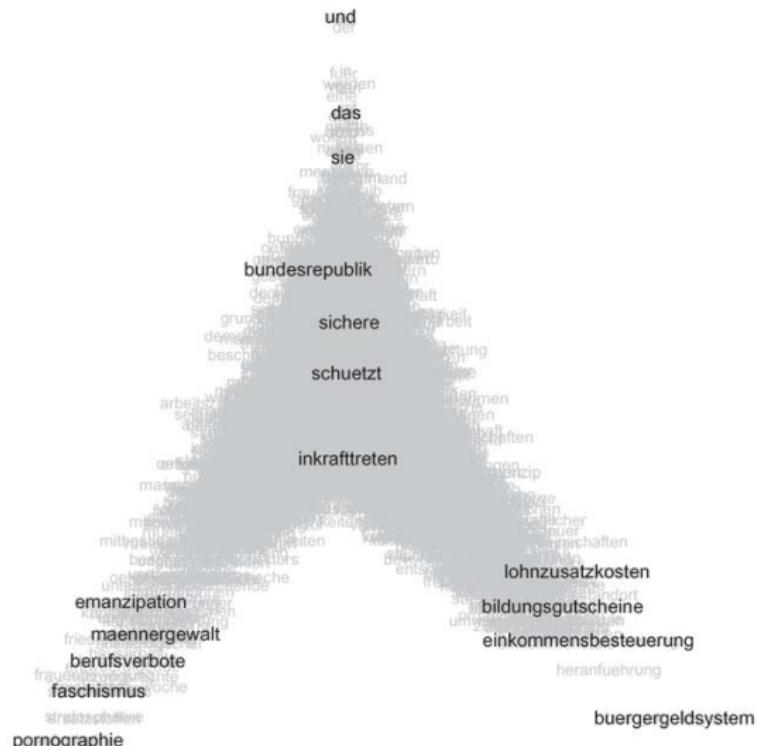
Results

Results



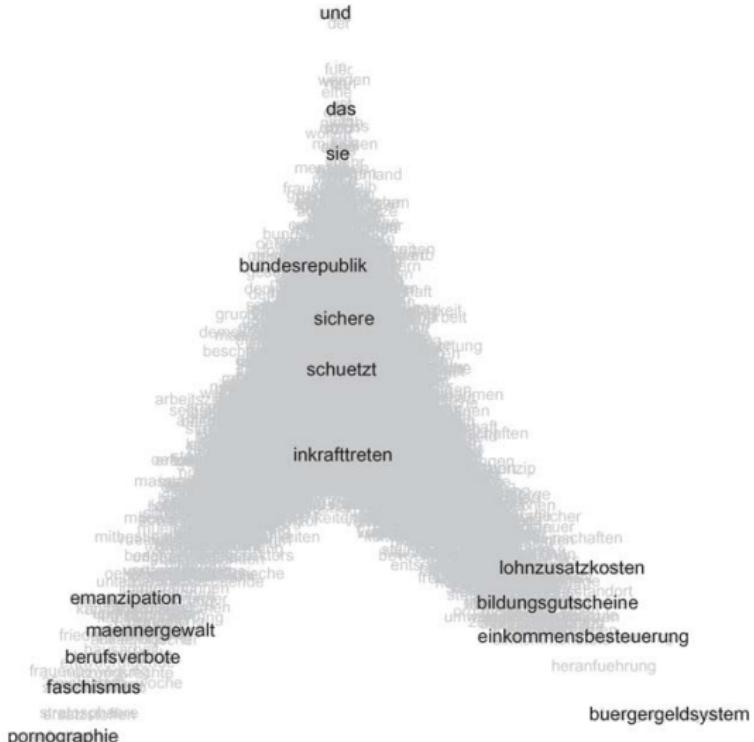
y is word fixed effects:

Results



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

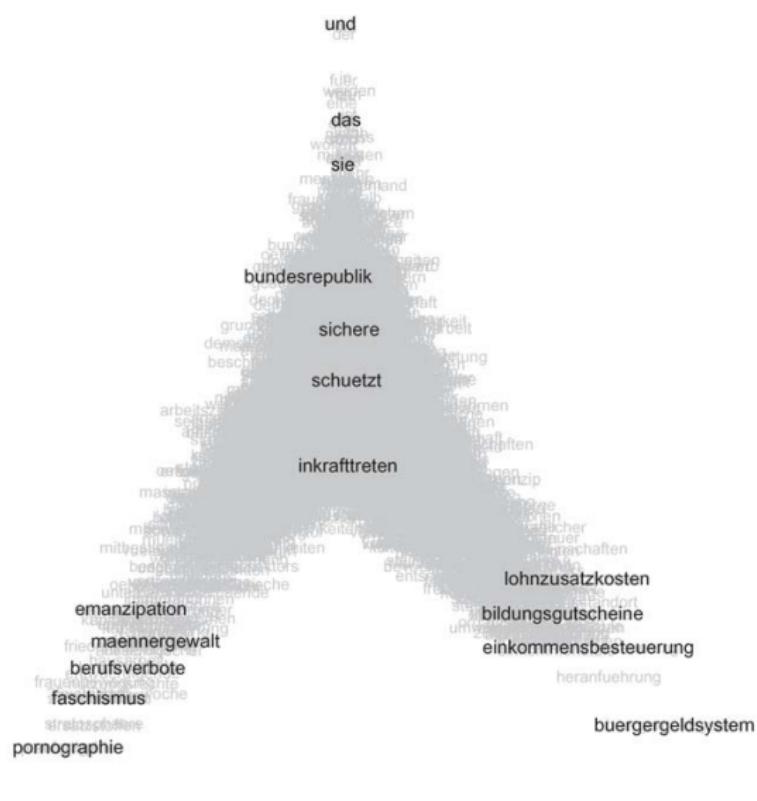
Results



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 (v common).

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

Results II

Results II

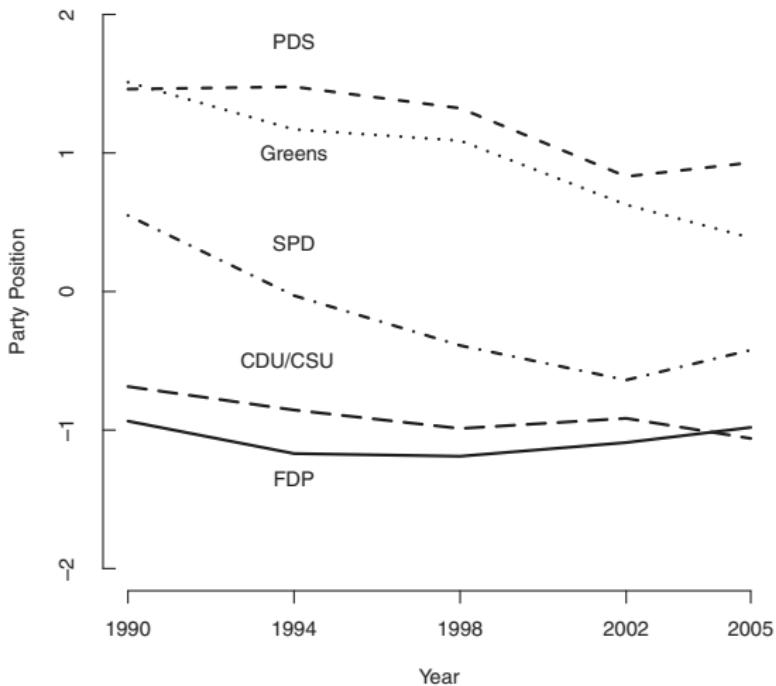
Top 10 Words Placing Parties on the . . .

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



Semi-supervised Techniques

Semi-Supervised Techniques

Semi-Supervised Techniques

May be **prohibitively costly** to provide enough **labeled** data for a supervised learning problem.

Semi-Supervised Techniques

May be **prohibitively costly** to provide enough **labeled** data for a supervised learning problem.

Turns out that **accuracy** of supervised text classifier can be (markedly) improved by adding large pool of **unlabeled** documents to a small number of training documents.

Semi-Supervised Techniques

May be **prohibitively costly** to provide enough **labeled** data for a supervised learning problem.

Turns out that **accuracy** of supervised text classifier can be (markedly) improved by adding large pool of **unlabeled** documents to a small number of training documents.

Intuition: **unlabeled** set provides useful information about **joint probability** over words.

Semi-Supervised Techniques

May be **prohibitively costly** to provide enough **labeled** data for a supervised learning problem.

Turns out that **accuracy** of supervised text classifier can be (markedly) improved by adding large pool of **unlabeled** documents to a small number of training documents.

Intuition: **unlabeled** set provides useful information about **joint probability** over words.

e.g. in labeled data,

Semi-Supervised Techniques

May be **prohibitively costly** to provide enough **labeled** data for a supervised learning problem.

Turns out that **accuracy** of supervised text classifier can be (markedly) improved by adding large pool of **unlabeled** documents to a small number of training documents.

Intuition: **unlabeled** set provides useful information about **joint probability** over words.

e.g. in labeled data, the word “military” is associated with being a Republican speech.

Semi-Supervised Techniques

May be **prohibitively costly** to provide enough **labeled** data for a supervised learning problem.

Turns out that **accuracy** of supervised text classifier can be (markedly) improved by adding large pool of **unlabeled** documents to a small number of training documents.

Intuition: **unlabeled** set provides useful information about **joint probability** over words.

e.g. in labeled data, the word “military” is associated with being a Republican speech. We then use this fact to classify thousands of **unlabeled** speeches.

Semi-Supervised Techniques

May be **prohibitively costly** to provide enough **labeled** data for a supervised learning problem.

Turns out that **accuracy** of supervised text classifier can be (markedly) improved by adding large pool of **unlabeled** documents to a small number of training documents.

Intuition: **unlabeled** set provides useful information about **joint probability** over words.

- e.g. in labeled data, the word “military” is associated with being a Republican speech. We then use this fact to classify thousands of **unlabeled** speeches.
- but then we find that the word “defence” co-occurs with ‘military’ in the **unlabeled** documents (which we just classified)

Semi-Supervised Techniques

May be **prohibitively costly** to provide enough **labeled** data for a supervised learning problem.

Turns out that **accuracy** of supervised text classifier can be (markedly) improved by adding large pool of **unlabeled** documents to a small number of training documents.

Intuition: **unlabeled** set provides useful information about **joint probability** over words.

- e.g. in labeled data, the word “military” is associated with being a Republican speech. We then use this fact to classify thousands of **unlabeled** speeches.
- but then we find that the word “defence” co-occurs with ‘military’ in the **unlabeled** documents (which we just classified)
- use this to build more accurate classifier.

Notes

0

Notes

Treat unlabeled data as having **missing** labels:

Notes

Treat unlabeled data as having **missing** labels: use **EM** algorithm to iterate.

Notes

Treat unlabeled data as having **missing** labels: use **EM** algorithm to iterate.

Can lead to massive **reduction** in need for **labeled** data.

Notes

Treat unlabeled data as having **missing** labels: use **EM** algorithm to iterate.

Can lead to massive **reduction** in need for **labeled** data.

But relies on several important assumptions,

Notes

Treat unlabeled data as having **missing** labels: use **EM** algorithm to iterate.

Can lead to massive **reduction** in need for **labeled** data.

But relies on several important assumptions, connected to idea that supervised examples are like/close to unsupervised documents.

Notes

Treat unlabeled data as having **missing** labels: use **EM** algorithm to iterate.

Can lead to massive **reduction** in need for **labeled** data.

But relies on several important assumptions, connected to idea that supervised examples are like/close to unsupervised documents.

If points aren't close/similar, SSL can give **worse** results.