PCA y Texto como Datos

Big Data y Machine Learning para Economía Aplicada

Ignacio Sarmiento-Barbieri

Universidad de los Andes

Agenda

1. Motivation
2. Principal Component Analysis
   - What are PCAs?
   - PC Interpretation
   - Principal Component Regression (PCR)
3. Text as Data
   - Text Regression
   - Topic Models
     - Label latent semantic analysis
     - Latent Dirichlet Allocation

Agenda

Motivation

- One way to think about almost everything we do is as dimension reduction.
- We are trying to
  - learn from high-dimensional X some low-dimensional summaries that contain the information necessary to make good decisions.
  - model it as having been generated from a small number of components/factors.
- We are attempting to simplify X for its own sake.

Motivation

- Unsupervised learning is often much more challenging.
- The exercise tends to be more subjective, and there is no simple goal for the analysis, such as prediction of a response.
- Unsupervised learning is often performed as part of an exploratory data analysis.
- Furthermore, it can be hard to assess the results obtained from unsupervised learning methods, since there is no universally accepted mechanism for performing cross- validation or validating results on an independent data set.
- There is no way to check our work because we don't know the true answer: the problem is unsupervised.

Agenda

1. Motivation

2. Principal Component Analysis
   - What are PCAs?
   - PC Interpretation
   - Principal Component Regression (PCR)

3. Text as Data
   - Text Regression
   - Topic Models
     - Label latent semantic analysis
     - Latent Dirichlet Allocation

Agenda

Principal Component Analysis

- PCA is an unsupervised learning technique that allows to
  - reduce the dimensionality of data sets,
  - while preserving as much "variability" as possible.
- It is an unsupervised approach, it involves only a set of variables/features $X_1, X_2, \ldots, X_p$, and no associated response Y .

Principal Component Analysis

- For example:
  1. Area
  2. Rooms
  3. Bathrooms
  4. Schools
  5. Crime

Principal Component Analysis

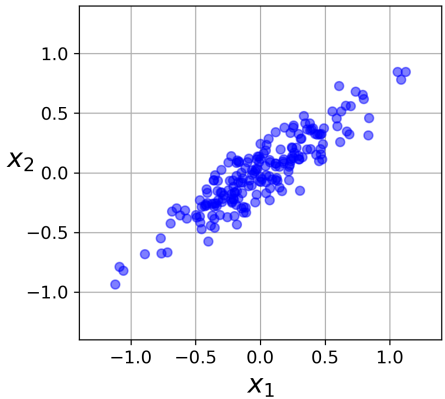
Area	Rooms	Bathrooms	Schools	Crime

PC1	PC2

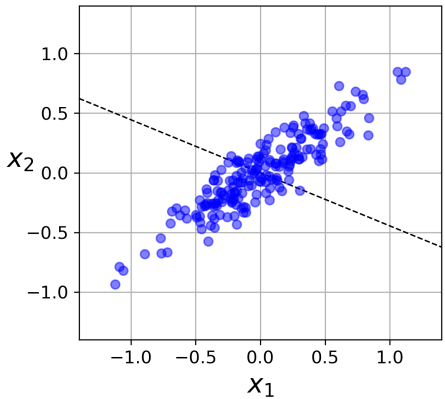
Principal Component Analysis

- PCA finds a low-dimensional representation of a data set that contains as much as possible of the variation.
- The idea is that each of the n observations lives in p-dimensional space, but not all of these dimensions are equally interesting.
- PCA seeks a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension.

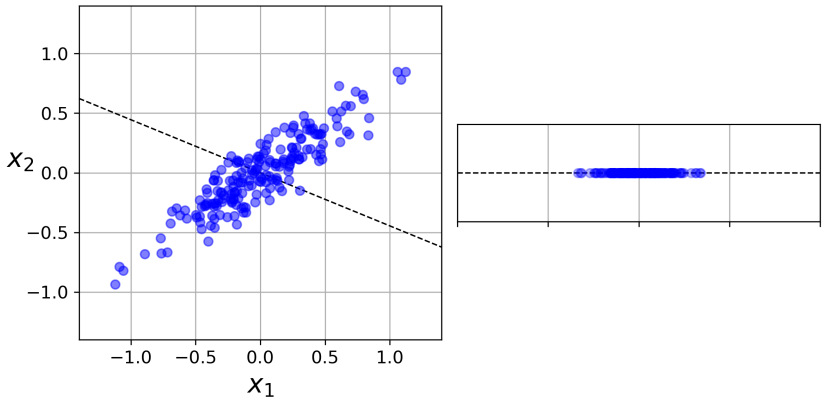
Principal Component Analysis



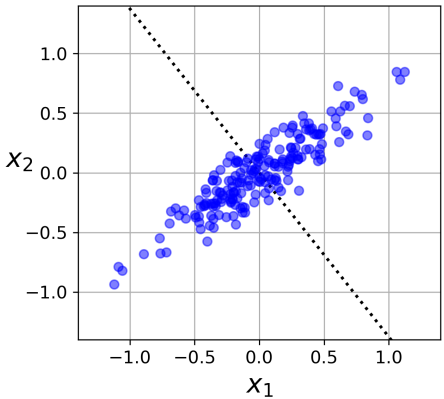
Principal Component Analysis



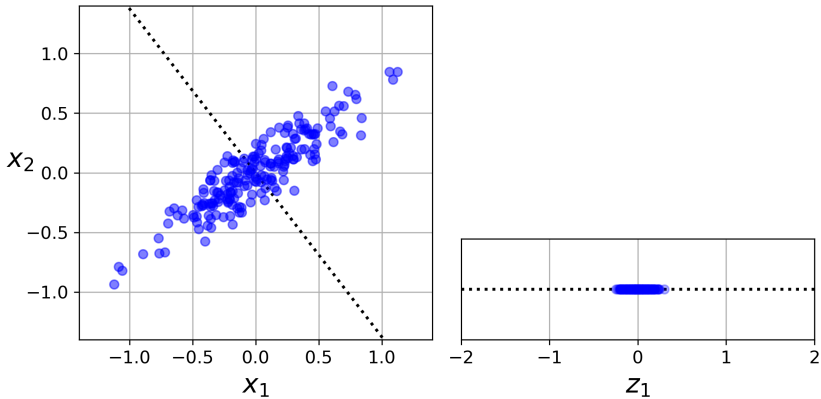
Principal Component Analysis



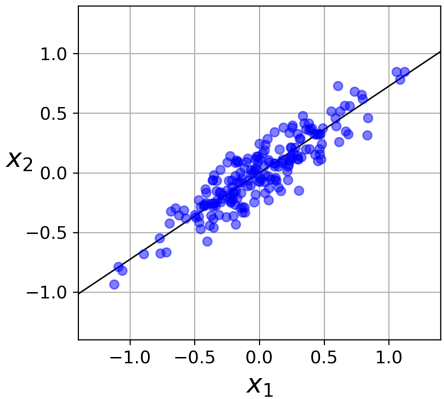
Principal Component Analysis



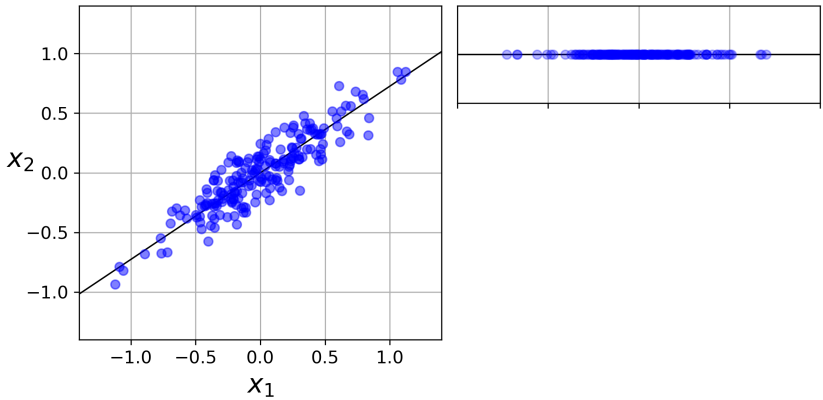
Principal Component Analysis



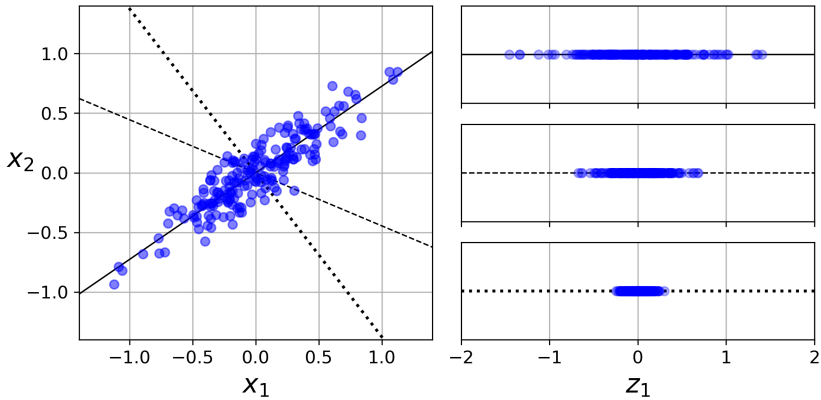
Principal Component Analysis



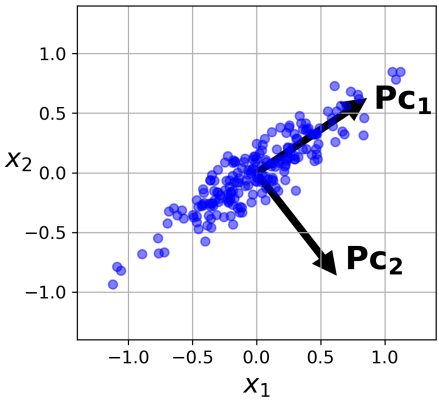
Principal Component Analysis



Principal Component Analysis



Principal Component Analysis



How do we compute the first principal component

- Given a $n \times p$ data set X, how do we compute the first principal component?

Detour: Algebra Review

- Let $A_{m\times m}$. It exists
  - a scalar λ such that $Ap = \lambda p$ for a vector $p_{m\times 1}$,
  - if $p \neq 0$, then λ is an eigenvalue of A.
  - and p is an eigenvector of A corresponding to the eigenvalue λ.
- $A_{m\times m}$ with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$, then:

$$tr(A) = \sum_{i=1}^{m} \lambda_i \tag{3}$$

$$det(A) = \Pi_{i=1}^{m} \lambda_i \tag{4}$$

- If $A_{m\times m}$ has m different eigenvalues, then the associated eigenvectors are all linearly independent.
- Spectral decomposition: $A = P\Lambda P'$

How do we compute the first principal component

- Given a $n \times p$ data set X, how do we compute the first principal component?

$$PC_1 = X\delta_1 \tag{1}$$
$$= \delta_{11}X_1 + \delta_{21}X_2 + \cdots + \delta_{p1}X_p \tag{2}$$

- The idea is to preserve the most information possible
- In other words, we are going to try to generate an index that reproduces (the best it can) the information (variability) of the original variables
- How we do that?

q main components

- Let $\lambda_1, \ldots, \lambda_p$ be the eigenvalues of $S = V(X)$, ordered from highest to lowest,
- $p_1, \ldots, p_p$ the corresponding eigenvectors.
- Call P the matrix of eigenvectors.
- Then $\delta_j = p_j$, $\forall j$ ('loadings' of the principal components =ordered eigenvectors of S).

Relative importance of factors

- Now we want to know the relative importance of factors, to have a way of choosing them

Selection of factors

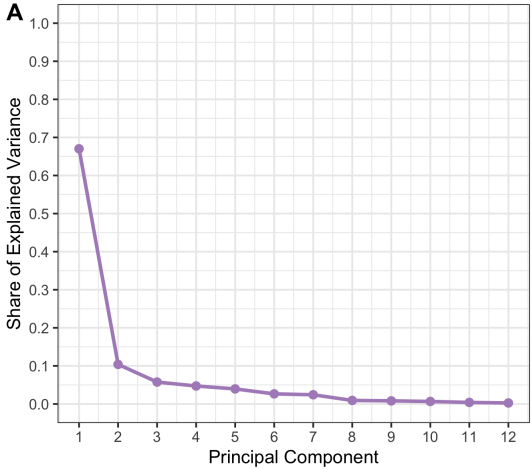
- Although a matrix X of dimension $n \times p$ generally has $min(n-1, p)$ different principal components.
- In practice, we are generally not interested in all the components, but rather stay with the first ones that allow us to visualize or interpret data.
- Indeed, we would like to keep the minimum number that allows us a good understanding of the data.
- The natural question that arises here is whether there is an established way to determine the number of principal components to use.
- Unfortunately, there is no accepted objective way in the literature to answer it.

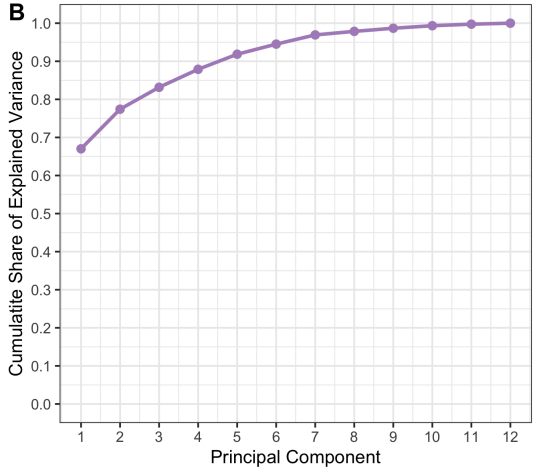
Selection of factors

- However, there are three simple approaches that can guide you in deciding the number of relevant major components.
  - Visual examination of screeplot
  - Kaiser criterion.
  - Proportion of variance explained.

Selection of factors

Screeplot





Selection of factors

Kaiser criterion

- Let the columns of X be standardized, so that each variable has unit variance.

Selection of factors

Proportion of variance explained

- Another approach often used in practice is to impose a threshold a priori and choose the main components based on it.
  - For example, we could define a threshold of 90%, which in the previous example plot would result in 5 main components.
  - Whereas if it were 70% we would have 2 main components.
- The threshold to be defined will depend on the application, the context, and the data set. Thresholds between 70% and 90% are typically used.

PC Computation

- Before I mentioned that data was standardized, that is, re-centered to have zero mean and scaled to have variance one.
- From a strictly mathematical point of view, there is nothing inherently wrong with making linear combinations of variables with different units of measurement.
- However, when we use PCA we seek to maximize variance and the variance is affected by the units of measurement.
- This implies that the principal components based on the covariance matrix S will change if the units of measure of one or more variables change.

PC Computation

- To prevent this from happening, it is common practice to standardize the variables. That is, each X value is re-centered and divided by the standard deviation:

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j} \tag{5}$$

- where $\bar{x}_j$ is the mean and s_j is the standard deviation of column j.
- Then the initial data matrix X is replaced by the standardized data matrix Z.
- Note also that when standardizing the data matrix, the covariance matrix S is simply the original data correlation matrix. This is sometimes referred to in the literature as the PCA correlation matrix.

PC Computation

Uniqueness of the main components

- It is necessary to warn that the ”loadings” of the main components δ are unique except for a sign change.
- This implies that depending on the implementation we can obtain different results from two libraries.
- The ”loadings” will be the same but the signs may differ.
- The signs may differ because each weight specifies a direction in k-dimensional space and the change of sign has no effect on the direction.

PC Computation

- As a practical aside, note that for really big sparse X, R will run out of memory.
- A big data strategy for PCA is to first calculate the covariance matrix for X and then obtain PC rotations as the eigenvalues of this covariance matrix.
  - The first step can be done using sparse matrix algebra.
  - The rotations are then available as

```
## eigen( xvar, symmetric = TRUE)$vec.
```

- There are also approximate PCA algorithms available for fast factorization on big data. See, for example, the `irlba` package for R.

Agenda

PC Interpretation

Caveat

- Component interpretation is hard because PCA focuses on variance, not meaning
- The technique optimally compresses information, but translating this compression into human-understandable concepts requires domain knowledge, simplifying assumptions, and statistical insight.
- As a result, many practitioners focus on explaining variance or ranking the importance of variables instead of finding exact meanings for each component.

Factor Model Interpretation

- Suppose we have p regressors and K=1

$$x_i = hf_i \tag{6}$$

- h is $p \times 1$
- f_i 1×1 and is the factor
- h are the factor loadings
- In this model, the factor f_i affects all regressors x_{ji}
- But the magnitude is specific to the regressor and captured by h

Factor Model Interpretation

Test Scores

$$x_i = hf_i \tag{7}$$

- x_i is a set of test scores for an individual student
- f_i is the student's latent ability
- h is how ability affects the different test scores
  - Some tests may be highly related to ability
  - Some tests may be less related
  - Some may be unrelated (random?)

Factor Model Interpretation

Test Scores

$$x_i = \sum_{m=1}^{k} h_m f_{mi} \tag{8}$$

- There are more than one form of ability
- i.e. literary and mathematical
- In labor economics, there has been hypothesized a distinction between cognitive and non-cognitive ability which has been very useful in explaining wage patterns (some jobs require one or the other, and some both (e.g. surgeon)

Factor Interpretation: Examples

Agenda

Principal Component Regression (PCR)

- Now that you've learned how to fit PCA models, what are they good for?
- In some settings, as in the previous political science example, the factors themselves have clear meaning and can be useful in their own right for understanding complex systems.
- More commonly, unfortunately, the factors are of dubious origin or interpretation.
- However, they can still be useful as inputs to a regression system.
- Indeed, this is the primary practical function for PCA, as the first stage of principal components regression (PCR).

Principal Component Regression (PCR)

- The concept of PCR is simple:
  - Instead of doing $y \to X$,
  - Use a lower-dimension set of principal components as covariates.
- This is a fruitful strategy for a few reasons:
  - PCA reduces dimension, which is usually good.
  - The PCs are independent, so you have no multicollinearity and the final regression is easy to fit.

Principal Component Regression (PCR)

- The disadvantage of PCR is that PCA will be driven by the dominant sources of variation in X.
- If the response is connected to these dominant sources of variation, PCR works well.
- If it is more of a "needle in the haystack response," driven by a small number of inputs, then PCR will not work well.
- In practice, you do not know what scenario you are in

Principal Component Regression (PCR)

- How many PC do we use?
  - When PCA was used as a dimensionality reduction tool *per se* we had some guidelines...
- Should we do the same here?

Principal Component Regression (PCR)

- How many PC do we use?
  - When PCA was used as a dimensionality reduction tool *per se* we had some guidelines...
- Should we do the same here?
- In PCR the approach is slightly different
  - Construct $min(n-1, p)$ components
  - Use K fold crossvalidation adding 1 PC at a time
  - Choose the model with the lowest out of sample MSE
- Because the PCs are ordered (by their variance) and independent, this works better than subset selection on the raw dimensions of X_i.

Principal Component Regression (PCR)

- An alternative mechanism is run a lasso on the full set of PCs (works best in practice).
- This procedure makes it easy to incorporate other information in addition to the PCs.
- For example, one tactic that works well in practice is to put both PC and Xs into the lasso model matrix.
  - This then allows the regression to make use of the underlying factor structure in X and still pick up individual X_j signals that are related to y.
  - This hybrid strategy is a solution to the disadvantage of PCR mentioned earlier—that it will only pick up dominant sources of variation in X.

Principal Component Regression (PCR)

Summary of the steps

- Given a sample of regression input observations x_i, accompanied by output labels y_i for some subset of these observations:
  1. Fit PCA on the full set of X inputs to obtain PC of length $min(n-1, p)$.
  2. For the labeled subset, run a lasso regression for y on f (PC).
     - Alternatively, regress y on f and Xs to allow simultaneous selection between PCs and raw inputs.
  3. To predict for a new X_{new}, use the rotations from step 1 to get $f = \delta X_{new}$ and then feed these scores into the regression fit from step 2.

PCR Example

Agenda

1. Motivation

2. Principal Component Analysis
   - What are PCAs?
   - PC Interpretation
   - Principal Component Regression (PCR)

3. Text as Data
   - Text Regression
   - Topic Models
     - Label latent semantic analysis
     - Latent Dirichlet Allocation

Text as Data: The Big Picture

- We generate vast quantities of raw unstructured text.
- As the costs of storage drop and as more conversations and records move to digital platforms, we accumulate massive corpora that track communications:
  - customer conversations,
  - product descriptions or reviews,
  - news,
  - comments, blogs, tweets, etc...
- The information in text is a rich complement to the more structured variables contained in a traditional transaction or customer database.

Text as Data: The Big Picture

- We generate vast quantities of raw unstructured text.
- As the costs of storage drop and as more conversations and records move to digital platforms, we accumulate massive corpora that track communications:
  - customer conversations,
  - product descriptions or reviews,
  - news,
  - comments, blogs, tweets, etc...
- The information in text is a rich complement to the more structured variables contained in a traditional transaction or customer database.
- Social scientists have also woken up to the potential of such data and recent years have seen an explosion in studies that make use of text as data.

Text as Data: Motivation

THE JOURNAL OF FINANCE • VOL. LXII, NO. 3 • JUNE 2007

Giving Content to Investor Sentiment: The Role of Media in the Stock Market

PAUL C. TETLOCK*

ABSTRACT

I quantitatively measure the interactions between the media and the stock market using daily content from a popular *Wall Street Journal* column. I find that high media pessimism predicts downward pressure on market prices followed by a reversion to fundamentals, and unusually high or low pessimism predicts high market trading volume. These and similar results are consistent with theoretical models of noise and liquidity traders, and are inconsistent with theories of media content as a proxy for new information about fundamental asset values, as a proxy for market volatility, or as a sideshow with no relationship to asset markets.

Text as Data: Motivation

Econometrica, Vol. 78, No. 1 (January, 2010), 35–71

WHAT DRIVES MEDIA SLANT? EVIDENCE FROM U.S. DAILY NEWSPAPERS

BY MATTHEW GENTZKOW AND JESSE M. SHAPIRO[1]

We construct a new index of media slant that measures the similarity of a news outlet's language to that of a congressional Republican or Democrat. We estimate a model of newspaper demand that incorporates slant explicitly, estimate the slant that would be chosen if newspapers independently maximized their own profits, and compare these profit-maximizing points with firms' actual choices. We find that readers have an economically significant preference for like-minded news. Firms respond strongly to consumer preferences, which account for roughly 20 percent of the variation in measured slant in our sample. By contrast, the identity of a newspaper's owner explains far less of the variation in slant.

KEYWORDS: Bias, text categorization, media ownership.

Text as Data

- Raw text (e.g., words or characters) needs to be transformed into a numeric form for the model to process.
  - Bag of Words (BoW) and DTMs
  - Word Embeddings

Text as Data

- El modelo BoW es una forma sencilla de representar texto como una colección de palabras ignorando la gramática y el orden de las palabras. En BoW, lo único que importa es la frecuencia de cada palabra en el documento. Por ejemplo 3 documentos
  - El sol es una estrella.
  - Un buen viajante no tiene planes.
  - Juan tiene una mascota nueva
- BoW extrae las palabras únicas (vocabulario) de los documentos: el, sol, es, una, estrella, un, buen, viajante, no, tiene, planes, Juan, mascota, nueva. (14 words).
- Cada documento se representa por la frecuencia de estas palabras, formando un vector para cada uno.

Text as Data: DTM

- Una DTM es una matriz que organiza estas representaciones BoW para múltiples documentos.
- Cada fila de la matriz corresponde a un documento, y cada columna representa una palabra del vocabulario global (de todos los documentos).
- Los valores dentro de la matriz son las frecuencias de las palabras en cada documento.

Text as Data: DTM

- However, we can think that not all the words in a document have the same weight.
- El TF-IDFVectorizer (term frequency-inverse document frequency) incorporates this notion

$$TF - IDF_{ij} = tf_{ij} \times \left(\log \left(\frac{1+N}{1+df_i} \right) + 1 \right)$$

where:

- tf_{ij} is the frequency of word i in the j document
- df_{ij} number of documents that have the word i
- N number of documents

Text as Data: Text cleaning and tokenization

- However not all words are useful in the DTM, we can follow certain rules to clean these up.
- The text cleaning process, within the scope of text mining, consists of eliminating everything from the text that does not provide information about its theme, structure or content.
- There is no single way to do it, it largely depends on the purpose of the analysis and the source from which the text comes.
- Tokenizing a text consists of dividing the text into the units that make it up, ending with the simplest element with its own meaning for the analysis in question

Text as Data: Text cleaning and tokenization

Some steps include

- Convert to lowercase, drop numbers, punctuation, etc ...
  Always application specific: e.g., don't drop :-) from tweets.
- Remove a list of stop words containing irrelevant tokens .
  If, and, but, who, what, the, they, their, a, or, ...

  Be careful: one person's stopword is another's key term.
- Remove words that are super rare (in say $< \frac{1}{2}\%$, or $< 15\%$ of docs; this is application specific). For example, if Argentine occurs only once, it's useless for comparing documents.

Text as Data: Text cleaning and tokenization

Stemming and lemmatization

- For grammatical reasons, documents are going to use different forms of a word, such as *organize, organizes, and organizing.*
- Additionally, there are families of derivationally related words with similar meanings, such as *democracy, democratic, and democratization.*
- In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set.

Text as Data: Text cleaning and tokenization

Stemming and lemmatization

- The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For instance:
  - am, are, is $\Rightarrow$ be
  - car, cars, car's, cars' $\Rightarrow$ car
- The result of this mapping of text will be something like:
  - the boy's cars are different colors $\Rightarrow$
  - the boy car be differ color

Text as Data: Text cleaning and tokenization

- However, the two differ in their flavor.
- Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.
- Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma .
- For example:
  If confronted with the token `saw`, stemming might return just `s` whereas lemmatization would attempt to return either `see` or `saw` depending on whether the use of the token was as a verb or a noun.

The n-gram language model

- An n-gram language model is one that describes a dialect through transition probabilities on n consecutive words.
- An n-gram tokenization counts length-n sequences of words.
  A unigram is a word, bigrams are transitions between words.
  e.g., `world.stage`, `stage.men`, `men.women`, `women.play`, ...
- This can give you rich language data, but be careful: n-gram token vocabularies are very high dimensional (p^n)
- More generally, you may have domain specific 'clauses' that you wish to tokenize.
- There is always a trade-off between complexity and generality.

Agenda

1. Motivation
2. Principal Component Analysis
   - What are PCAs?
   - PC Interpretation
   - Principal Component Regression (PCR)
3. Text as Data
   - Text Regression
   - Topic Models
     - Label latent semantic analysis
     - Latent Dirichlet Allocation

Text Regression

- Once you have text in a numeric format, we can use all the tools we learned so far
- For example: Find words more related to Republicans

$$Republican\ Share = f(X\beta) + u \tag{9}$$

- where X is the cleaned DTM

Example

Agenda

1. Motivation
2. Principal Component Analysis
   - What are PCAs?
   - PC Interpretation
   - Principal Component Regression (PCR)
3. Text as Data
   - Text Regression
   - Topic Models
     - Label latent semantic analysis
     - Latent Dirichlet Allocation

Topic Models

- Since text is super high dimensional we can use PCA
- Versions of this algorithm were referred to under the label latent semantic analysis.

Example

Latent Dirichlet Allocation

- The approach of using PCA to factorize text was common before the 2000s.
- However, this changed with the introduction of topic modeling, also known as Latent Dirichlet Allocation (LDA), by Blei et al. in 2003.
- These authors pointed out that the squared error loss (i.e., Gaussian model) implied by PCA is inappropriate for analysis of sparse word-count data.
- Instead, they proposed you take the bag-of-words representation seriously and model token counts as realizations from a multinomial distribution.

Latent Dirichlet Allocation

TRANSPARENCY AND DELIBERATION WITHIN THE FOMC: A COMPUTATIONAL LINGUISTICS APPROACH*

STEPHEN HANSEN
MICHAEL MCMAHON
ANDREA PRAT

How does transparency, a key feature of central bank design, affect monetary policy makers' deliberations? Theory predicts a positive discipline effect and negative conformity effect. We empirically explore these effects using a natural experiment in the Federal Open Market Committee in 1993 and computational linguistics algorithms. We first find large changes in communication patterns after transparency. We then propose a difference-in-differences approach inspired by the career concerns literature, and find evidence for both effects. Finally, we construct an influence measure that suggests the discipline effect dominates. *JEL Codes*: E52, E58, D78.

Latent Dirichlet Allocation

THE PARTICIPATION DIVIDEND OF TAXATION:
HOW CITIZENS IN CONGO ENGAGE MORE WITH
THE STATE WHEN IT TRIES TO TAX THEM*

JONATHAN L. WEIGEL

This article provides evidence from a fragile state that citizens demand more of a voice in the government when it tries to tax them. I examine a field experiment randomizing property tax collection across 356 neighborhoods of a large Congolese city. The tax campaign was the first time most citizens had been registered by the state or asked to pay formal taxes. It raised property tax compliance from 0.1% in control to 11.6% in treatment. It also increased political participation by about 5 percentage points (31%): citizens in taxed neighborhoods were more likely to attend town hall meetings hosted by the government or submit evaluations of its performance. To participate in these ways, the average citizen incurred costs equal to their daily household income, and treated citizens spent 43% more than control. Treated citizens also positively updated about the provincial government, perceiving more revenue, less leakage, and a greater responsibility to provide public goods. The results suggest that broadening the tax base has a "participation dividend," a key idea in historical accounts of the emergence of inclusive governance in early modern Europe and a common justification for donor support of tax programs in weak states. *JEL* Codes: H20, P48, D73.

Latent Dirichlet Allocation

TABLE VII

TOPICS OF CITIZEN COMMENTS AT TOWN HALLS AND WRITTEN-IN COMMENTS ON SUBMITTED EVALUATIONS

Order	(1)	(2)	(3)	(4)	(5)
Panel A: Topics of citizen comments at town hall meetings					
1	pay	tax	necessary	pay	pay
2	necessary	population	population	take	must
3	population	necessary	collectors	without	population
4	tax	pay	pay	decision	why
5	why	know	know	why	others
6	agents	do	see	necessary	collectors
7	time	collectors	tax	participation	agents
8	collectors	why	without	tax	nothing
9	communes	nothing	information campaign	others	participation
10	manager	schools	transparency	agents	tax
Panel B: Topics of written-in comments on submitted evaluations					
1	government	government	government	government	government
2	water	provincial	provincial	provincial	province
3	ask	should	should	work	country
4	roads	more	population	province	leaders
5	electricity	work	especially	do	population
6	improve	public	erosion	better	good
7	jobs	goods	needs	ask	ask
8	people	concerning	people	would	development
9	more	ask	security	central	love
10	who	because	take	Kasaï	could

Notes. This table reports the first ten words in each of the five main topics identified by latent Dirichlet allocation (Blei, Ng, and Jordan 2003) applied to two sources of text that offer insight into citizens' reasons

Latent Dirichlet Allocation

- Blei et al. proposed you take the bag-of-words representation seriously and model token counts as realizations from a multinomial distribution.
- Topic models are built on a simple document generation process:
  - For each word, pick a "topic" k. This topic is defined through a probability vector over words, say, θ_k with probability θ_{kj} for each word j.
  - Then draw the word according to the probabilities encoded in θ_k .
- After doing this over and over for each word in the document, you have proportion ω_{i1} from topic 1, ω_{i2} from topic 2, and so on.

Latent Dirichlet Allocation

- This basic generation process implies that the full vector of word counts, x_i, has a multinomial distribution:

$$x_i \sim MN(\omega_{i1}\theta_1 + \cdots + \omega_{iK}\theta_K, m_i) \tag{10}$$

- where $m_i = \sum_j x_{ij}$ is the total document length and, for example,
- the probability of word j in document i will be $\sum_k \omega_{ik}\theta_{kj}$

Example