Lecture 1: Introduction to Probabilistic Learning

EPFL Summer School in Computational Methods for Economists

Stephen Hansen
University of Oxford / Imperial College London

Introduction

As much as 90% of usable data is unstructured, and much of this is text.

Why is text useful for economists?

1. Often the medium through which agents in the economy express their views.
2. Added source of richness even when complementary quantitative data exist.
3. Available at relatively high frequency and in situations where traditional data are absent.

There is an increasing interest in treating text quantitatively,[1] but these methods are still in their infancy.

[1] Gentzkow, Kelly, and Taddy (forthcoming, JEL) "Text as Data".

Selected Applications

1. Polarization (Gentzkow and Shapiro, 2010, ECMA; Gentzkow, Shapiro, and Taddy, forthcoming, ECMA).
2. Uncertainty (Baker, Bloom, and Davis, 2016, QJE).
3. Forecasting (Larsen and Thorsrud, 2019, JoE; Mueller and Rauh, forthcoming, APSR).
4. Transparency and monetary policy (Hansen, McMahon, and Prat, 2018, QJE).
5. Idea exchange (Iaria, Schwarz, and Waldinger, 2018, QJE).
6. Product market competition (Hoberg and Phillips, 2016, JPE).
7. Labor force skill (Hershbein and Kahn, 2018, AER).

Frameworks for Text Analysis

Historically, dictionary-based approaches have dominated in economics and finance, and will continue to play an important role.

In computer science and AI, neural networks have grown remarkably in popularity over the last ten years.

These lectures instead focus on probabilistic (aka generative) models of text. Motivation:

1. Relevant keywords not always apparent.
2. Well-defined joint distribution allows for well-defined inference.
3. Akin to structural models in economics.
4. Social scientists often not in “Big Data” environments.
5. We often care more about content than language *per se*.

Plan

1. Introduce basic ideas in unsupervised learning for text data.
2. Latent Dirichlet allocation (LDA): model, inference, and applications.
3. Supervised learning with text.

Notation

The corpus is composed of D documents indexed by d.

After pre-processing, each document is a finite, length-N_d list of terms $\mathbf{w}_d = (w_{d,1}, \dots, w_{d,N_d})$ with generic element $w_{d,n}$.

Let $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_D)$ be a list of all terms in the corpus, and let $N \equiv \sum_d N_d$ be the total number of terms in the corpus.

Suppose there are V **unique** terms in $\mathbf{w}$, where $1 \leq V \leq N$, each indexed by v.

We can then map each term in the corpus into this index, so that $w_{d,n} \in \{1, \dots, V\}$.

Let $x_{d,v} \equiv \sum_n \mathbb{1}(w_{d,n} = v)$ be the count of term v in document d.

FOMC Example

Running example is corpus of verbatim FOMC transcripts from the era of Alan Greenspan:

- 149 meetings from August 1987 through January 2006.
- A document is a single statement by a speaker in a meeting (46,502).
- Baseline data has 6,249,776 total words and 26,030 unique words.
- Metadata include macro conditions, speaker characteristics, etc.

Document-Term Matrix

A popular quantitative representation of text is the *document-term matrix* $\mathbf{X}$, which collects the counts $x_{d,v}$ into a $D \times V$ matrix.

The key characteristics of the document-term matrix are its:

1. High dimensionality
2. Sparsity

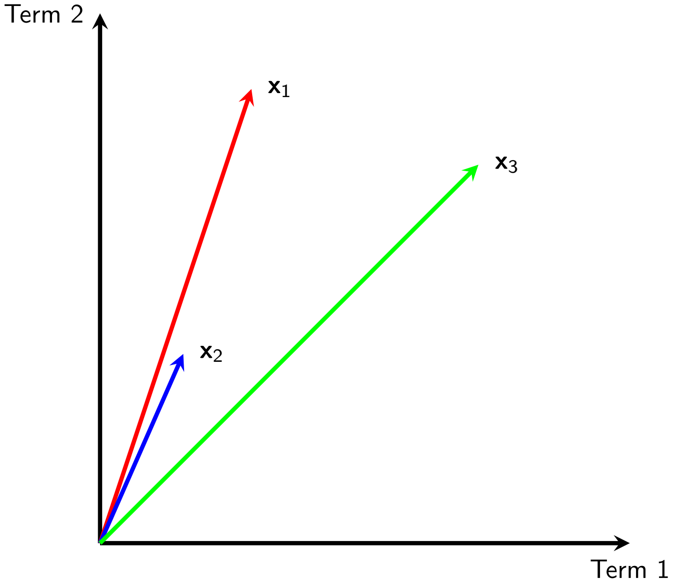
Vector Space Model

One can view the rows of the document-term matrix as vectors lying in a V-dimensional space.

The basis for the vector space is $e_1, \ldots, e_V$.

The question of interest is how to measure the similarity of two documents in the vector space, and whether unsupervised learning can help with this.

Three Documents



Cosine Similarity

Define the cosine similarity between documents i and j as

$$CS(i,j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}$$

1. Since document vectors have no negative elements $CS(i,j) \in [0,1]$.
2. $\mathbf{x}_i / \|\mathbf{x}_i\|$ is unit-length, correction for different distances.

Information Retrieval

The problem of *synonomy* is that several different words can be associated with the same topic. Cosine similarity between following documents?

school	university	college	teacher	professor
0	5	5	0	2
school	university	college	teacher	professor
10	0	0	4	0

The problem of *polysemy* is that the same word can have multiple meanings. Cosine similarity between following documents?

tank	seal	frog	animal	navy	war
10	10	3	2	0	0
tank	seal	frog	animal	navy	war
10	10	0	0	4	3

If we correctly map words into topics, comparisons become more accurate.

Simple Probability Model

Consider the list of terms $\mathbf{w} = (w_1, \ldots, w_N)$ where $w_n \in \{1, \ldots, V\}$.

Suppose that each term is iid, and that $\Pr[\, w_n = v\,] = \beta_v \in [0,1]$.

Let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_V) \in \Delta^{V-1}$ be the parameter vector we want to estimate.

The probability of the data given the parameters is

$$\Pr[\,\mathbf{w} \mid \boldsymbol{\beta}\,] = \prod_n \sum_v \mathbb{1}(w_n = v)\beta_v = \prod_v \beta_v^{x_v}$$

where x_v is the count of term v in $\mathbf{w}$.

Note that term counts are a sufficient statistic for $\mathbf{w}$ in estimating $\boldsymbol{\beta}$.

Maximum Likelihood Inference

We can estimate $\boldsymbol{\beta}$ with maximum likelihood. The Lagrangian is

$$\mathfrak{L}(\boldsymbol{\beta}, \lambda) = \underbrace{\sum_v x_v \log(\beta_v)}_{\text{log-likelihood}} + \lambda \underbrace{\left(1 - \sum_v \beta_v\right)}_{\text{Constraint on } \boldsymbol{\beta}} .$$

First order condition is $\frac{x_v}{\beta_v} - \lambda = 0 \Rightarrow \beta_v = \frac{x_v}{\lambda}$.

Constraint gives $\frac{\sum_v x_v}{\lambda} = 1 \Rightarrow \lambda = \sum_v x_v = N$.

So MLE estimate is $\widehat{\beta}_v = \frac{x_v}{N}$, the frequency of term v in list of terms.

Multinomial Mixture Model

We now want to introduce a model of document heterogeneity over a latent topical space.

Suppose that there are K separate term distributions $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_K$, each of which represent a topic.

Each document d in the corpus has a latent topic assignment $z_d \in \{1, \dots, K\}$.

$\Pr [\, z_d = k \,] = \rho_k$ and is independent across documents.

This defines a *multinomial mixture model*.

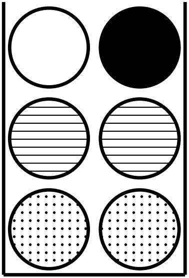
Topics as Urns

○ = wage

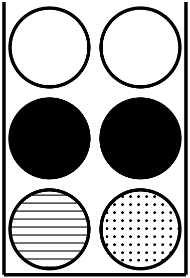
▤ = price

● = employ

⠿ = increase

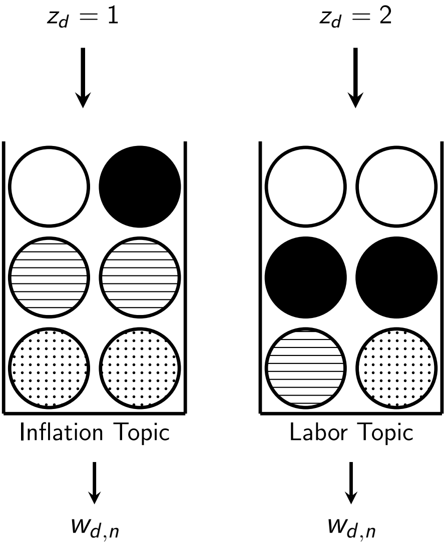


“Inflation” Topic



“Labor” Topic

Mixture Model for Document



$z_d = 1$

$z_d = 2$

Inflation Topic

Labor Topic

$w_{d,n}$

$w_{d,n}$

Likelihood Function

Suppose that $z_d = k$. Then the probability of $\mathbf{x}_d$ is $\prod_v (\beta_{k,v})^{x_{d,v}}$.

To compute the unconditional probability of document d, we need to marginalize over the latent assignment variable z_d

$$\begin{aligned}
\Pr[\mathbf{x}_d \mid \boldsymbol{\rho}, \boldsymbol{\beta}] &= \sum_k \Pr[z_d = k \mid \boldsymbol{\rho}, \boldsymbol{\beta}] \Pr[\mathbf{x}_d \mid z_d = k, \boldsymbol{\rho}, \boldsymbol{\beta}] \\
&= \sum_k \rho_k \prod_v (\beta_{k,v})^{x_{d,v}}.
\end{aligned}$$

By independence of latent variables across documents, the likelihood of entire corpus is

$$L(\mathbf{X} \mid \boldsymbol{\rho}, \boldsymbol{\beta}) = \prod_d \sum_k \rho_k \prod_v (\beta_{k,v})^{x_{d,v}}$$

so the log-likelihood is

$$\ell(\mathbf{X} \mid \boldsymbol{\rho}, \boldsymbol{\beta}) = \sum_d \log \left(\sum_k \rho_k \prod_v (\beta_{k,v})^{x_{d,v}} \right).$$

Estimation via EM Algorithm

We cannot directly maximize the log-likelihood ℓ due to the sum in the logarithm. But if we knew the values of z_d the problem would be easy.

Let the *complete data log-likelihood* be

$$\ell_{\mathrm{comp}}\left(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\rho}, \boldsymbol{\beta}\right) = \sum_d \sum_k \mathbb{1}(z_d = k) \left[\log(\rho_k) + \sum_v x_{d,v} \log\left(\beta_{k,v}\right) \right].$$

The expectation-maximization algorithm consists of alternating between the following steps:

1. (E-step). Compute distributions over z_d given current parameter values, and use them to compute the expected value of ℓ_{comp}.
2. (M-step). Update parameter values by maximizing the expected value of ℓ_{comp} computed at the E step.

Expectation Step

Compute expected value of the complete data log-likelihood with respect to the latent variables given the current value of the parameters $\boldsymbol{\rho}^i$ and $\boldsymbol{\beta}^i$ and data.

Expectation Step

Compute expected value of the complete data log-likelihood with respect to the latent variables given the current value of the parameters $\boldsymbol{\rho}^i$ and $\boldsymbol{\beta}^i$ and data.

Clearly $\mathbb{E}\left[\mathbb{1}(z_d = k) \,\middle|\, \boldsymbol{\rho}^i, \boldsymbol{\beta}^i, \mathbf{X} \right] = \Pr\left[z_d = k \,\middle|\, \boldsymbol{\rho}^i, \boldsymbol{\beta}^i, \mathbf{X} \right] \equiv \widehat{z}_{d,k}$.

By Bayes' Rule we have that

$$\begin{aligned} \widehat{z}_{d,k} = &\Pr\left[z_d = k \,\middle|\, \boldsymbol{\rho}^i, \boldsymbol{\beta}^i, \mathbf{x}_d \right] \propto \\ &\Pr\left[\mathbf{x}_d \,\middle|\, \boldsymbol{\rho}^i, \boldsymbol{\beta}^i, z_d = k \right] \Pr\left[z_d = k \,\middle|\, \boldsymbol{\rho}^i, \boldsymbol{\beta}^i \right] = \rho_k \prod_v \left(\beta_{k,v} \right)^{x_{d,v}}. \end{aligned}$$

So the expected complete log-likelihood becomes

$$Q(\boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{\rho}^i, \boldsymbol{\beta}^i) = \sum_d \sum_k \widehat{z}_{d,k} \left[\log(\rho_k) + \sum_v x_{d,v} \log\left(\beta_{k,v} \right) \right]$$

Maximization Step

Maximize the expected complete log-likelihood with respect to $\boldsymbol{\rho}$ and $\boldsymbol{\beta}$.

Maximization Step

Maximize the expected complete log-likelihood with respect to $\boldsymbol{\rho}$ and $\boldsymbol{\beta}$.

The Lagrangian for this problem is

$$Q(\boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{\rho}^i, \boldsymbol{\beta}^i) + \nu \left(1 - \sum_k \rho_k\right) + \sum_k \lambda_k \left(1 - \sum_v \beta_{k,v}\right).$$

Maximization Step

Maximize the expected complete log-likelihood with respect to $\boldsymbol{\rho}$ and $\boldsymbol{\beta}$.

The Lagrangian for this problem is

$$Q(\boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{\rho}^i, \boldsymbol{\beta}^i) + \nu \left(1 - \sum_k \rho_k\right) + \sum_k \lambda_k \left(1 - \sum_v \beta_{k,v}\right).$$

Standard maximization gives

$$\rho_k^{i+1} = \frac{\sum_d \hat{z}_{d,k}}{\sum_k \sum_d \hat{z}_{d,k}},$$

or the average probability that documents have topic k and

$$\beta_{k,v}^{i+1} = \frac{\sum_d \hat{z}_{d,k} x_{d,v}}{\sum_d \hat{z}_{d,k} \sum_v x_{d,v}},$$

or the expected number of times documents of type k generate term v over the expected number of words generated by type k documents.

Example

Let $K = 2$ and consider the corpus of 1,232 paragraphs of State-of-the-Union Addresses since 2000.

Topic	Top Terms
0	tax.job.help.must.congress.need.health.care.busi.let.school.time
1	world.countri.secur.must.terrorist.iraq.state.energi.help.unit

$(\rho_0, \rho_1) = (0.42, 0.58)$.

No *ex ante* labels on clusters, so any interpretation is *ex post*, and potentially subjective, judgment on the part of the researcher.

Why does the EM Algorithm Work? (from Bishop 2006)

Suppose we have a probability model with observed data $\mathbf{X}$, unobserved data $\mathbf{Z}$, and parameters $\boldsymbol{\theta}$.

We want to maximize the function $p(\mathbf{X} \mid \boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$.

Imagine an arbitrary distribution over the latent variables $q(\mathbf{Z})$. The following decomposition then holds (for all $\boldsymbol{\theta}$ and q):

$$\log[p(\mathbf{X} \mid \boldsymbol{\theta})] = \underbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \log\left[\frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{q(\mathbf{Z})}\right]}_{\text{Lower Bound}} +$$

$$\underbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \log\left[\frac{q(\mathbf{Z})}{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})}\right]}_{\mathrm{KL}[q(\mathbf{Z}) \| p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta})] \geq 0}$$

Expectation Step

Suppose we are at the jth iteration of the algorithm with current parameter estimates $\boldsymbol{\theta}^j$.

In the expectation step, we set $q^j(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^j)$, which implies the lower bound is tight.

Using the above decomposition, we obtain

$$\log[p(\mathbf{X} \mid \boldsymbol{\theta}^j)] = \sum_{\mathbf{Z}} q^j(\mathbf{Z}) \log \left[\frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}^j)}{q^j(\mathbf{Z})} \right]$$

Maximization Step

In the maximization step, we maximize the lower bound with respect to $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{j+1} \in \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{Z}} q^j(\mathbf{Z}) \log \left[\frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{q^j(\mathbf{Z})} \right]$$

The objective function can be written as

$$\underbrace{\sum_{\mathbf{Z}} q^j(\mathbf{Z}) \log[p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})]}_{\text{expected complete data log-likelihood}} - \underbrace{\sum_{\mathbf{Z}} q^j(\mathbf{Z}) \log[q^j(\mathbf{Z})]}_{\text{(negative) entropy}}$$

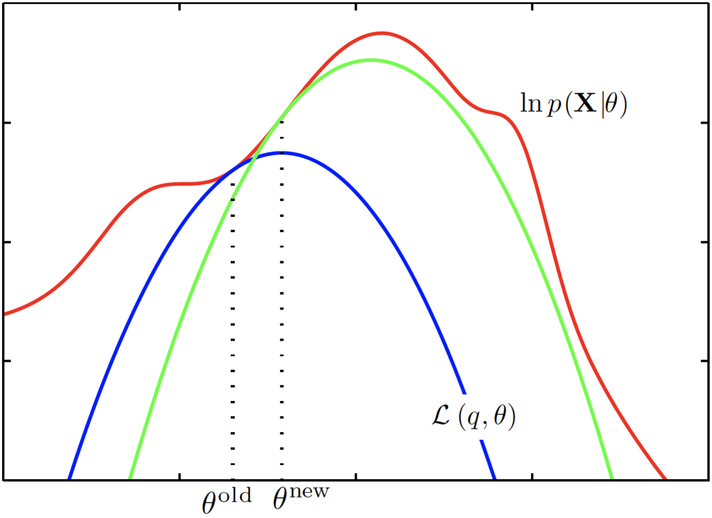
Maximizing the lower bound with respect to $\boldsymbol{\theta}$ is equivalent to maximizing the expected complete data log-likelihood.

At the new parameters $\boldsymbol{\theta}^{j+1}$, we must have

$$p(\mathbf{X} \mid \boldsymbol{\theta}^{j+1}) \geq \sum_{\mathbf{Z}} q^j(\mathbf{Z}) \log \left[\frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}^{j+1})}{q^j(\mathbf{Z})} \right] \geq$$

$$\sum_{\mathbf{Z}} q^j(\mathbf{Z}) \log \left[\frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}^{j})}{q^j(\mathbf{Z})} \right] = p(\mathbf{X} \mid \boldsymbol{\theta}^{j})$$

Illustration (source: Bishop 2006)



Mixed-Membership Models

In the multinomial mixture model, documents are associated with a single topic.

In practice, we might imagine that documents cover more than one topic.

Examples: State-of-the-Union Addresses discuss domestic and foreign policy; monetary policy speeches discuss inflation and growth.

Models that associated observations with more than one latent variable are called *mixed-membership* models. Also relevant outside of text mining: in models of group formation, agents can be associated with different latent communities (sports team, workplace, church, etc).

Probabilistic Mixed-Membership Model

In the probabilistic LSA model of Hofmann (1999), we allow for word-level mixtures according to document-specific topic shares $\theta_{d,k}$.

Instead of assigning each document to a topic, we assign each word in each document to a topic.

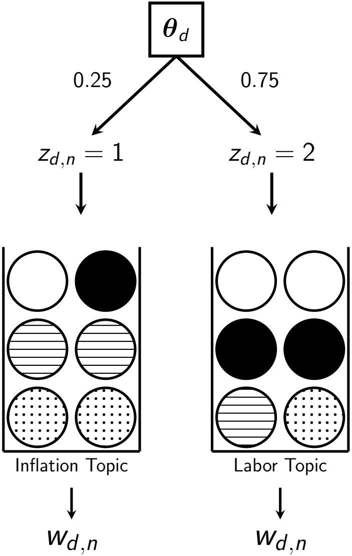
Let $z_{d,n} \in \{1, \dots, K\}$ be the topic assignment of $w_{d,n}$, where $\Pr[\, z_{d,n} = k \,] = \theta_{d,k}$.

The likelihood function for this model is

$$\prod_d \prod_n \sum_{z_{d,n}} \Pr\left[\, w_{d,n} \,\middle|\, \boldsymbol{\beta}_{z_{d,n}} \right] \Pr[\, z_{d,n} \mid \boldsymbol{\theta}_d \,]$$

Can estimate via EM algorithm, but large number of parameters makes it prone to overfitting.

Mixed-Membership Model for Document



Conclusion

We have introduced basic ideas in probability models for text.

Latent variable models yield intractable likelihoods, but the EM algorithm allows for indirect optimization.

In high-dimensional spaces, the number of parameters in the likelihood function can grow large.