Econ 2148, fall 2019
Text as data

Maximilian Kasy

Department of Economics, Harvard University

Agenda

- One big contribution of machine learning methods to econometrics is that they make new forms of data amenable to quantitative analysis: Text, images, ...
- We next discuss some methods for turning text into data.
- Key steps:
  1. Converting corpus of documents into numerical arrays.
  2. Extracting some compact representation of each document.
  3. Using this representation for further analysis.
- Two approaches for step 2:
  1. Supervised:
     E.g., Lasso prediction of outcomes based on word counts.
  2. Unsupervised:
     E.g., topic models, “latent Dirichlet allocation.”

Takeaways for this part of class

- To make text (or other high-dimensional discrete data) amenable to statistical analysis, we need to generate low-dimensional summaries.
- Supervised approach:
  1. Regress observed outcome Y on high-dimensional description $\boldsymbol{w}$.
     Use appropriate regularization and tuning.
  2. Impute predicted $\hat{Y}$ for new realizations $\boldsymbol{w}$.
- Unsupervised approach:
  1. Assume texts are generated from distributions
     corresponding to topics.
  2. Impute unobserved topics.
- Topic models are a special case of hierarchical models.
  These are useful in many settings.

Notation

- **Word**: Basic unit, out of a vocabulary indexed by $v \in \{1, \ldots, V\}$.
  Represent words by unit vectors, $w = \delta_v$.
- **Document**: A sequence of N words,
  $$\boldsymbol{w} = (w_1, w_2, \ldots, w_N).$$
- **Corpus**: A collection of M documents,
  $$\boldsymbol{D} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}.$$

Introduction

- Many sources of digital text for social scientists:
  - political news, social media, political speeches,
  - financial news, company filings,
  - advertisements, product reviews, ...
- Very high dimensional: For a document of N words from a vocabulary of size V, there are V^N possibilities.
- Three steps:
  1. Represent text as numerical array $\boldsymbol{w}$.
     (Drop punctuation and rare words, count words or phrases.)
  2. Map array to an estimate of a latent variable.
     (Predicted outcome or classification to topics.)
  3. Use the resulting estimates for further analysis.
     (Causal or other.)

Representing text as data

- Language is very complex. Context, grammar, ...
- Quantitative text analysis discards most of this information.

Data preparation steps:

1. Divide corpus $\boldsymbol{D}$ into documents j, such as
   - the news of a day, individual news articles,
   - all the speeches of a politician, single speeches, ...
2. Pre-process documents:
   - Remove punctuation and tags,
   - remove very common words (“the, a,” “and, or,” “to be,” ...),
   - remove very rare words (occurring less than k times),
   - stem words, replacing them by their root.

N-grams

3. Next, convert resulting documents into numerical arrays $\boldsymbol{w}$.

- Simplest version:
  Bag of words. Ignore sequence.
  w_v is the count of word v, for every v in the vocabulary.
- Somewhat more complex:
  $w_{vv'}$ is the count of ordered occurrence of the words v, v',
  for every such “bigram.”
- Can extend this to N-grams, i.e., sequences of N words.
  But $N > 2$ tends to be too unwieldy in practice.

Dimension reduction

- Goal: Represent high-dimensional $\boldsymbol{w}$ by some low-dimensional summary.
- 4 alternative approaches:

1. Dictonary-based: Just define a mapping $g(\boldsymbol{w})$.
2. Predict observed outcome Y based on $\boldsymbol{w}$.
   Use predicted $\hat{Y}$ as summary.
   "Supervised learning."
3. Predict $\boldsymbol{w}$ based on observed outcome Y.
   "Generative model." Invert to get $\hat{Y}$.
4. Predict $\boldsymbol{w}$ based on unobserved latent θ.
   "Topic models." Impute $\hat{\theta}$ and use as summary.
   "Unsupervised learning."

Text regression

- Suppose we observe outcomes Y for a subset of documents.
- We want to
  - Estimate $E[Y|\boldsymbol{w}]$ for this subset,
  - impute $\hat{Y} = E[Y|\boldsymbol{w}]$ for new draws of $\boldsymbol{w}$.
- $\boldsymbol{w}$ is (very) high-dimensional, so we can't just run OLS.
- Instead, use penalized regression:

$$\begin{aligned}\hat{\beta} &= \underset{\beta}{\operatorname{argmin}} \sum_j (Y_j - \boldsymbol{w}_j \beta)^2 + \lambda \sum_v |w_v|^p \\ \hat{Y}_j &= \boldsymbol{w}_j \beta.\end{aligned}$$

- $p = 1$ yields Lasso, $p = 2$ yields Ridge.
- λ is chosen using cross-validation.

Non-linear regression

- We are not restricted to squared error objectives.
  For instance, for binary outcomes, we could use penalized logit:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_j \frac{\exp(Y_j \boldsymbol{w}_j \beta)}{1 + \exp(\boldsymbol{w}_j \beta)} + \lambda \sum_v |w_v|^p$$

$$\hat{Y}_j = \frac{\exp(\boldsymbol{w}_j \beta)}{1 + \exp(\boldsymbol{w}_j \beta)}.$$

- Resist the temptation to give a substantive interpretation to (non-)zero coefficients for Lasso!
- Which variables end up included is very unstable when regressors are correlated (even if predictions $\hat{Y}$ are stable).
- Other prediction methods can also be used: Deep nets (coming soon), random forests...

Generative language models

- Generative models give a probability distribution over documents.
- Let us start with a very simple model.
- **Unigram** model: The words of every document are drawn independently from a single multinomial distribution.
- The probability of a document is

$$p(\boldsymbol{w}) = \prod_n p(w_n).$$

- The vector of probabilities $\beta = (p(\delta_1), \dots, p(\delta_V))$ is a point in the simplex spanned by the words δ_v.
- In the unigram model, each document is generated based on the same vector.

Mixture of unigrams

- A more complicated model is the “mixture of unigrams model.”
- This model assumes that each document has an unobserved topic z.
- Conditional on z, words are sampled from a multinomial distribution with parameter vector β_z.
- **Mixture of unigrams**: The probability of a document is

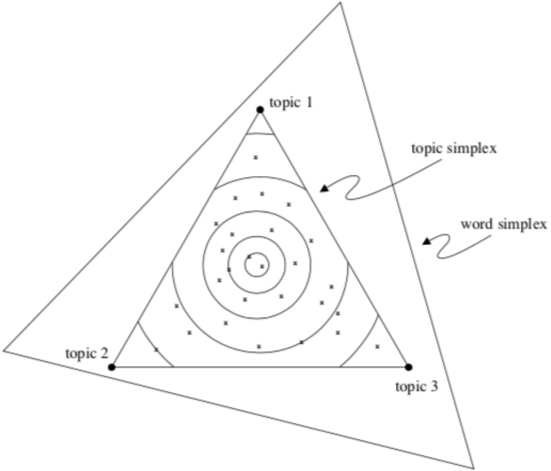
$$p(\boldsymbol{w}) = \sum_z p(z) \prod_n p(w_n|z)$$

where

$$p(w_n|z) = \beta_{z,w_n}.$$

- The vector of probabilities β_z is again a point in the simplex spanned by the words δ_v. Each topic corresponds to one point in this simplex.

Word and topic simplex

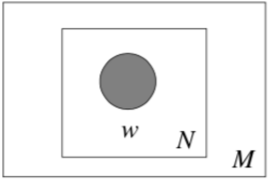


Graphical representation of hierarchical models

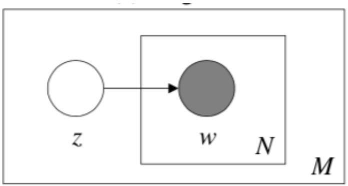
- The mixture of unigrams model is a simple case of a hierarchical model.
- Hierarchical models are defined by a sequence of conditional distributions. Not all variables in these models need to be observed.
- Hierarchical models are often represented graphically:
  - Observed variables are shaded circles, unobserved variables are empty circles.
  - Arrows represent conditional distributions.
  - Boxes are “plates” representing replicates.
    Replicates are conditionally independent repeated draws.
  - In the next slide, the outer plate represents documents.
  - The inner plate represents the repeated choice of words within a document.

Graphical representation

- Unigram:

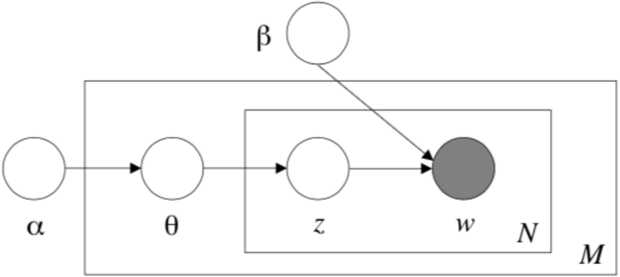


- Mixture of unigrams:



Practice problem

- Interpret the following representation of the latent Dirichlet allocation model, which we will discuss next.
- Write out its joint likelihood function.
- Write out the likelihood function of the corpus of documents ***D***.



Latent Dirichlet allocation

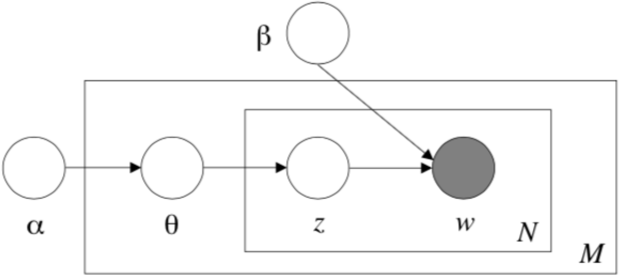
- We will now consider a very popular generative model of text.
- This is a generalization of the mixture of unigrams model.
- Introduced by Blei et al. (2003).
- For modeling text corpora and other collections of discrete data.
- Goal: Find short descriptions of the members of a collection.

"To enable efficient processing of large collections while preserving the essential statistical relationships that are useful for basic tasks such as classification, novelty detection, summarization, and similarity and relevance judgments."

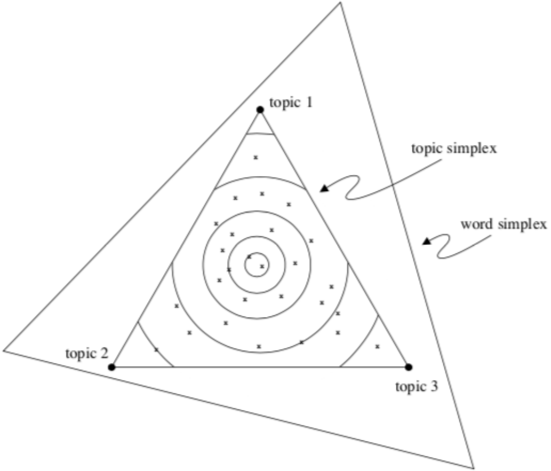
Latent Dirichlet model

1. **Exchangeability:** As before, we ignore the order of words in documents, and the order of documents. Think of this as throwing away information, not an assumption about the data generating process.
2. Condition on document lengths N.
3. For each document, draw a mixture of k topics
$$\theta \sim Dirichlet(\alpha).$$
4. Given θ, for each of the N words in the document draw a topic
$$z_n \sim Multinomial(\theta).$$
5. Given θ and z_n, draw a word w_n from the topic distribution β_{z_n},
$$w_n \sim \beta_{z_n},$$
where $\beta_{z_n,v}$ is the probability of word δ_v for topic z_n,

Graphical representation of the latent Dirichlet model



Word and topic simplex



Practice problem

What is the dimension of the parameter space for

1. The unigram model,
2. the mixture of unigrams model,
3. the latent Dirichlet allocation?

Likelihood

- Dirichlet distribution of topic-mixtures:

$$p(\theta|\alpha) = const. \cdot \prod_{j=1}^{k} \theta_j^{\alpha_j - 1}.$$

- Joint distribution of topic mixture θ, a set of N topics $\boldsymbol{z}$, and a set of N words $\boldsymbol{w}$:

$$p(\theta, \boldsymbol{z}, \boldsymbol{w}) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta).$$

Practice problem

Calculate, as explicitly as possible,

1. the probability of a given document $\boldsymbol{w}$,
2. the probability of the corpus $\boldsymbol{D}$.

Solution

- Probability of a given document $\boldsymbol{w}$:

$$p(\boldsymbol{w}|\alpha,\beta) = \int p(\theta|\alpha)\left(\prod_n \sum_{z_n} p(z_n|\theta)p(w_n|z_n,\beta)\right)d\theta$$

$$= const. \cdot \int \left(\prod_{j=1}^{k} \theta_j^{\alpha_j - 1}\right)\left(\prod_n \sum_{z_n} \theta_{z_n}\beta_{z_n,w_n}\right)d\theta$$

- Probability of the corpus $\boldsymbol{D}$:

$$p(\boldsymbol{D}|\alpha,\beta) = \prod_d \left[\int p(\theta|\alpha)\left(\prod_n \sum_{z_n} p(z_n|\theta)p(w_n|z_n,\beta)\right)d\theta\right].$$

- Note that again words $\boldsymbol{w}$, topics β_z, and mixtures of topics $\sum_z \theta_z\beta_z$ all live in the same simplex in $\mathbb{R}^V$!

Estimation

- Closed form likelihoods are not available.
- How to maximize the marginal likelihood,
  how to get the conditional expectation of θ_d?
- Blei et al. (2003): Combine
  1. variational inference (maximizing a lower bound on the likelihood),
  2. EM algorithm (alternate expectation and maximization).
- Alternative: Markov Chain Monte Carlo.
- Useful tool: **Stan**. General purpose environment for sampling from posteriors for hierarchical models. Available in R and other languages. Manual: `https://mc-stan.org/docs/2_18/bayes-stats-stan/index.html`

References

- *Gentzkow, M., Kelly, B. T., and Taddy, M. (2019). Text as data.* Journal of Economic Literature, *forthcoming*.

- *Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation.* Journal of Machine Learning Research, *3(Jan):993–1022*.