

Master's Paper of the Department of Statistics, the University of Chicago

# Hidden Structure of Text Data

Latent Topic Models and Sentiment Analysis

Nelson Auner

Advisors: Prof. Matt Taddy, Prof. Stephen Stigler

Approved \_\_\_\_\_

Date \_\_\_\_\_

## Abstract

This paper introduces a variant to existing models of multinomial regression for text analysis. Using the base model introduced by Taddy (2013), we extend the data-generating model to incorporate topics not explained by metadata. In doing so, we seek to both increase the prediction accuracy over existing techniques, bridge the gap between multinomial regression and standard topic models, and investigate methods for discovering new topics in a corpus. We explore computational aspects of our approach, provide software for parallelization of the algorithm, and conclude by proposing areas of future research.

## Contents

<b>Introduction</b>	<b>3</b>
Text Data . . . . .	3
Multinomial Model . . . . .	3
Topic Models . . . . .	4
Metadata . . . . .	4
<b>Theory and Approach</b>	<b>5</b>
Mixture models and cluster membership . . . . .	5
Model Specification . . . . .	5
Estimation of Parameters via Maximum a Posteriori . . . . .	6
Computation . . . . .	6
<b>Application</b>	<b>7</b>
Congressional Speech . . . . .	7
Evaluation of cluster-hot-starting . . . . .	7

Convergence and Stability of Clusters . . . . .	7
Interpretation of results . . . . .	8
Evaluating fit . . . . .	8
<b>Conclusion</b>	<b>8</b>
<b>Appendix</b>	<b>9</b>

# Introduction

## Text Data

Multinomial models are a common way of modeling annotated text. Typically, text information is naturally grouped by documents, and each document is represented by counts of words, or "tokens". A document might be a single written text (e.g. an academic article), or a collection of works by the same author (e.g. all of the lyrics of an album by the rolling stones) A token is often a single word (called unigram) but may also be a sequence of two or more words (e.g. bi-grams, like 'good swimmer' is a bigram, or tri-grams, like 'I eat cheese'). The word components of tokens are often reduced to a root form by removing suffixes (e.g. 'illuminated', 'illumination' and 'illuminating' all become 'illuminate').

These tokens are then aggregated by document: For  $i$  in  $i = 1, \dots, N$ , the count vector  $x_i = [x_{i1}, x_{i2}, \dots, x_{ip}]$  contains the number of occurrences of first, second,  $\dots$   $p$ th token in the  $i$ th document, where  $p$  is the total number of unique tokens in all documents. This forms the complete count matrix  $X$ , where each  $x_{ij}$  is the number of occurrences of word  $j$  in document  $i$ .

Since the number of unique words that appear in a large number of documents can be extensive, we often restrict the number of tracked tokens,  $p$  to words that occur in at least two documents. We may also remove common tokens that add little meaning and are found in all documents (i.e. 'the' or 'of').

A trivial example of such content might be student's answers to the question "What did homework assignments involve?", with the following four responses:

```
Some computation and formula proving, a lot of R code.  
Problems, computation using R  
Some computations and writing R code.  
Proofs, problems, and programming work
```

After removing common words and stemming the remaining words, we might produce the following count matrix:

Document	Some	comp	formula	prov	R	code	use	problem	writ	program	work
1	1	1	1	1	1	1	0	0	0	0	0
2	0	1	0	0	1	0	1	1	0	0	0
3	1	1	0	0	1	0	0	0	1	0	0
4	0	0	0	1	0	0	0	1	0	1	1

## Multinomial Model

We then model each document  $x_i$  as the realization of a multinomial distribution. That is,

$$x_i \sim MN(q_i, m_i)$$

Where  $q_i$  is the vector  $[q_{i1}, \dots, q_{ip}]$  of token probabilities for document  $x_i$  and  $m_i$  is  $\sum_{j=1}^p x_{ij}$ , or the total number of tokens in document  $i$

It is trivial to show that the maximum likelihood estimator of  $q_i$  is  $f_i = x_i/m_i$ , but by imposing a structure on  $q_i$ , we can model features of the data. The two most common techniques for creating structure are *topic models* and *metadata*

## Topic Models

A topic model structure assumes that each document is created from a linear combination of  $K$  topics. Each topic  $l = 1, \dots, K$  represents a distribution, or vector of probability weights  $\omega_l = [\omega_{l1}, \dots, \omega_{lp}]$ , over words. As a simple example, we can imagine a fitness store that primarily sells books on biking, running, and swimming. We can see that a probability distribution of these topics would have high probability weights on the terms (“pedal”, “helment”) for biking, (“stride”) for running, and (“breath”, “stroke”, “water”) for swimming. By denoting the proportion of topic  $l$  as  $\theta_l$ , we can imagine each document as being generated by a linear combination of topics  $\omega_1\theta_1 + \omega_2\theta_2 + \omega_3\theta_3$ , described as the following data-generating process:

1. Choose  $\theta = \theta_1, \dots, \theta_K$  the proportion of topics. (i.e., a book completely about swimming would have  $\theta = (1, 0, 0)$ , a book about triathalons might have  $\theta = (1/3, 1/3, 1/3)$ ).
2. Choose  $m_i$ , the number of words in the document
3. For each word  $j \in 1, 2, \dots, m_i$ , choose topic  $l$  with probability  $\theta_l$ . With the corresponding weighting vector  $\omega_l$ , choose a word  $x_{ij}$

## Metadata

Text data is frequently accompanied by information, or metadata, about the text itself. For example, in academic journals, metadata on an article could include the number of times the article has been cited, and the journal in which the article has been published. When this metadata is believed to be relevant to the composition of the document, we use the generic term *sentiment*. For example, given a database of written movie reviews and final rating out of five  $y \in (1, 2, 3, 4, 5)$ , we might want to model the relationship between the words used in the document and the final rating. A simple log-link model is that the contents of a document  $x$  with given metadata rating  $y$ , denoted  $x_y$ , is

$$x_y \sim MN(q_y, m_y) \text{ with } q_{yj} \sim \frac{\exp[\alpha_j + y\phi]}{\sum_{l=1}^p \exp[\alpha_l + y\phi_l]}$$

To determine the linear relationship between metadata  $y$  and count data  $x$ , we use Cook’s Inverse Regression method (2007) to reduce the dimension of  $x$  while maintaining its predictive power on  $y$ . That is, find  $\phi$  such that  $y_i - \phi'x_i \perp x_i$ . This criteria is called *sufficient reduction*.

We then take  $\phi$  and use it to predict content  $x_i$  from metadata  $y_i$ . This technique can be used to create a sparse coefficient matrix, and is, in fact, necessary to avoid over-fitting in the many cases where the number of words  $p$  is greater than the number of documents  $N$ . The inverse regression metadata approach has a computational advantage over topic models in that when creating maximum likelihood scores, the word count data  $x_{ij}$  can be collapsed by metadata label, that is  $x_y = \sum_{y_i=y} x_i$

Recent articles have proposed and implemented versions combining both metadata and topic modeling approaches.

## Theory and Approach

### Mixture models and cluster membership

We now turn our attention to the purpose of this paper, which is to implement an approach to discover and model “hidden”, or previously unspecified traits, across documents by grouping content unexplained by metadata. Initially, we will restrict our model by assuming that every document is a member of one and only one topic, or "cluster".

As a simple motivating example, we might imagine a corpus of movie reviews written by several bloggers. After accounting for text information explained by the rating (e.g. relating a 5-star rating to ‘good plot’), the remaining heterogeneity in the movie review content could be related traits of the blogger (e.g. gender, or home city) We may be interested in using predicting traits about bloggers given their movie reviews, and also in determining how movie review content changes across these traits.

### Model Specification

Denoting the word count of a document as the vector  $x_i$ , we propose that words in a document are distributed as a multinomial with a log-link to related sentiment or topics. That is:

$$x_i \sim MN(q_{ij}, m_{ij}); q_{ij} = \frac{\exp(\alpha_j + y_i \phi_j + u_i \Gamma_{kj})}{\sum_{l=1}^p \exp(\alpha_l + y_i \phi_l + u_i \Gamma_{kl})}$$

where  $y_i$ ,  $u_i$  are the metadata and cluster membership associated with document  $i$ , and  $\phi_j$  and  $\Gamma_{kj}$  are the distortion coefficients for the respective metadata and factor membership. We use the subscript  $k$  to denote that each document  $x_i$  is considered a member of  $k = 1, \dots, K$  clusters, with their own distortion vectors  $\Gamma_1, \dots, \Gamma_K$

Our focus is on predicting cluster membership  $u_i$  and the corresponding probability distortion  $\Gamma_i$ . To do so, we initialize cluster membership using one of the three following methods:

1. Random Initialization
2. K-means on the word count data  $X$
3. K-means on the residual of the word count data after incorporating metadata  $y$  (That is, give  $\hat{X}$  predicted document counts, clustering on  $X - \hat{X}$ )

## Estimation of Parameters via Maximum a Posteriori

The negative log likelihood of a multinomial distribution can be written as

$$\ell(\alpha, \phi, \Gamma, u) = \sum_{i=1}^N x_i^\top (\alpha + \phi v_i + u_i \Gamma_{kj}) - m_i \log \left( \sum_{j=1}^p \exp[\alpha + \phi v_i + u_i \Gamma_{kj}] \right) \quad (1)$$

We specify laplace priors and gamma hyperprior on coefficients, as well as a gamma lasso penalty on coefficients  $c(\Phi, \Gamma)$ . This procedure leads us to minimize

$$\ell(\alpha, \Phi, \Gamma, u) + \sum_{j=1}^p (\alpha_j / \sigma_\alpha)^2 + c(\Phi, \Gamma) \quad (2)$$

The basic algorithm we use to fit coefficients  $\alpha, \phi, \Gamma$  and cluster memberships  $u$  is two main steps iterated until convergence:

1. Determine parameters  $\alpha, \phi, \Gamma$  by fitting a multinomial regression on  $y_i | x_i, u_i$  with a gamma lasso penalty
2. For each document  $i$ , determine new cluster  $u_i$  membership as  $\operatorname{argmax}_{k=1, \dots, K} [\ell(y_i, x_i, u_k | \alpha, \phi, \Gamma)]$

By alternating between the two steps, we aim to converge to optimal parameter estimates  $\alpha, \phi, \Gamma$  as well as optimal cluster membership  $u$ .

## Computation

As noted by Taddy, multinomial regression enjoys the ability of being able to collapse observations across levels of metadata. This attractive property is preserved even in our more complex cluster membership model.

We can increase the speed of step two by only evaluating portions of the likelihood function relevant to  $u_i$  and  $\Gamma$  by eliminating first two terms from equation one:

$$L(\alpha, \phi, \Gamma, u_i) = \sum_{i=1}^N x_i^\top (u_i \Gamma_{kj}) - m_i \log \left( \sum_{j=1}^p \exp[\alpha + \phi v_i + u_i \Gamma_{kj}] \right) \quad (3)$$

In addition, the right hand side does not depend on  $x_i$  and can be precalculated for each cluster  $u_i$ . This will lead to an order-of-magnitude speed-up as long as the number of clusters is relatively small compared to the number of documents.

# Application

## Congressional Speech

We applied this method to the congressional speech data of Moskowitz and Shapiro and report our results. (Description of Data here)

## Evaluation of cluster-hot-starting

Method 2:

Clustering the congressional members by speech content led to the overwhelming (majority around 450) being placed in the same cluster, with 1, 3, and 74 persons in the other clusters. After running the algorithm, clusters seemed to be more or less random

Method 3:

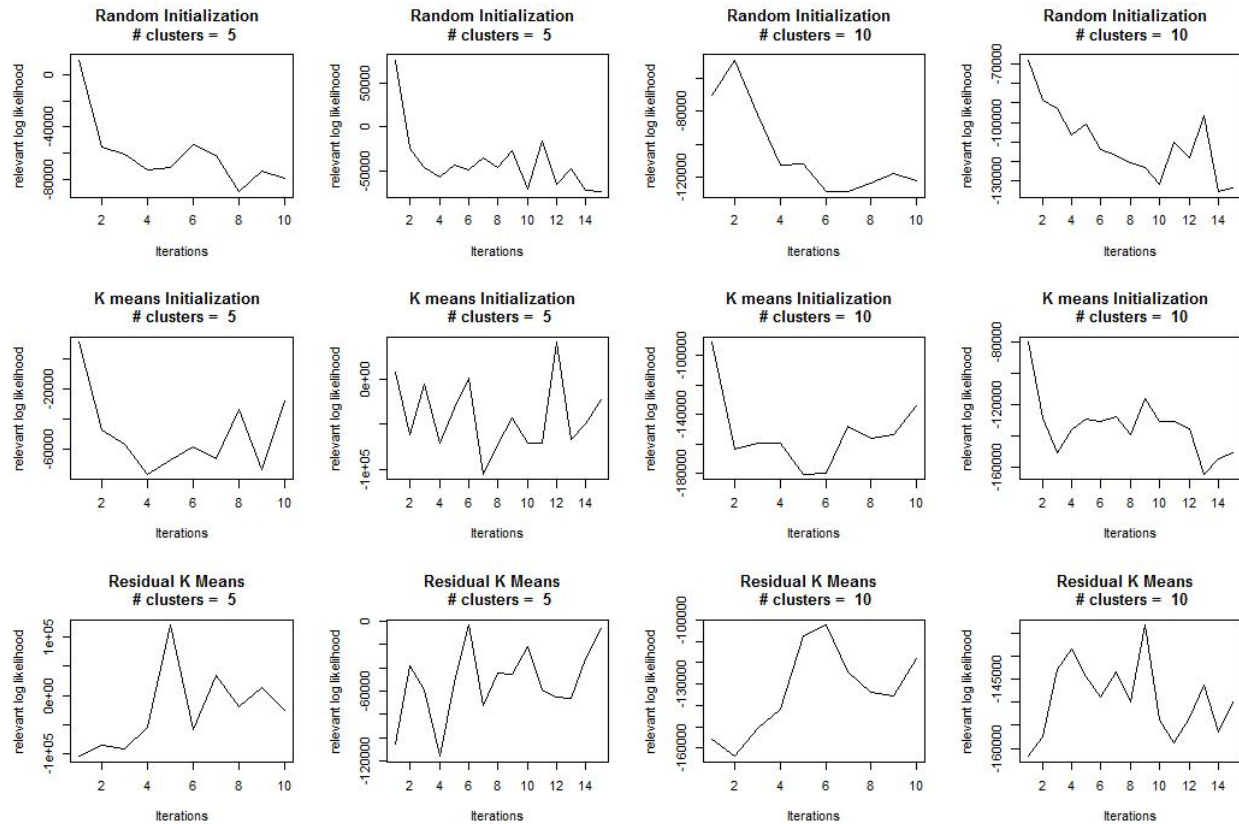
Clustering the congressional members by unexplained speech content (that is, on the residuals of the regression of party affiliation led to similar results, albeit that the size of the main cluster increased from 450 to 500 congressional members. After the algorithm,

## Convergence and Stability of Clusters

Of key interest is knowing whether or not the algorithm converges to the same clusters when run repeatedly on the same data, and also how this convergence is affected by the 3 proposed cluster initialization

At each step in classic two





## Interpretation of results

### Evaluating fit

*Graphs*

## Conclusion

# Appendix

The following may be included in an appendix:

Data

Simulation codes

Certain derivations

## References

- [1] Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London* (A) 222: 309-368.
- [2] Galton, F. (1883). *Inquiries into Human Faculty and Its Development*. London: Macmillan.