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

Combining latent topics with document attributes in text analysis

Nelson Auner

Advisors: Prof. Matt Taddy[1], Prof. Stephen Stigler[2]

Approved ____________________________________

Date ____________________________________

[1] Associate Professor of Econometrics and Statistics at Chicago Booth School of Business

[2] Ernest DeWitt Burton Distinguished Service Professor at the Department of Statistics of the University of Chicago

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 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

Introduction

Text Data

A common technique for modeling text data is the use of multinomial models on word stems derived from the original document. 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 $i = 1, ..., N$, the count vector $x_i = [x_{i1}, x_{i2}, ..., x_{ip}]$ contains the number of occurrences of first, second, … p th token in the ith 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 occurences 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:

Table 1: Example of text data from course reviews

Document	Content
1	Some computation and formula proving, a lot of R code
2	Problems, computation using R
3	Some computations and writing R code
4	Proofs, problems, and programming work

After removing common words and stemming the remaining words, we might produce the count matrix in Table 2.

Multinomial Model

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

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

Table 2: Creating a word-count matrix from text

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

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 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 $l = 1, .., K$ represents a distribution, or vector of probability weights $\omega_l = [\omega_{l1}, ..., \omega_{lp}]$, 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 θ_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 $\theta = \theta_1, ...\theta_K$ 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 $j \in 1, 2, ..., m_i$, choose topic l with probability θ_l. With the corresponding weighting vector ω_l, choose a word x_{ij}

Traditionally, the topic model proportions are given a dirchelet prior, and the model is also known as Latend Dirichlet Allocation, or LDA. For a thorough introduction to topic models,we refer the reader to Blei, Ng, and Jordan (2003).

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 metada 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.

Metadata and Unigram Models

If the support Y of metadata y takes discrete values $y^{(1)}, y^{(2)}, \dots, y^{(m)}$ with few unique observations (m small), large computational gains can be had by collapsing the token counts over levels of metadata. That is, each dataset of n ordered (text, metadata) pairs

$$\Big[(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\Big] \tag{1}$$

can be expressed as m collapsed observations:

$$\Big[(\sum_{x_i : y_i = y^{(1)}} x_i), (\sum_{x_i : y_i = y^{(2)}} x_i), \dots, (\sum_{x_i : y_i = y^{(m)}} x_i)\Big] \tag{2}$$

Then, a simple log-link model allows us to express document x with given metadata rating y, denoted x_y, as $x_y \sim MN(q_y, m_y)$, with

$$q_{yj} \sim \frac{exp[\alpha_j + y\phi_j]}{\sum_{l=1}^{p} exp[\alpha_l + y\phi_l]} \tag{3}$$

Theory and Approach

Mixture models and cluster membership

We now turn our attention to the main purpose of this paper, which is to incorporate latent topics across documents while maintaining the computational simplicity of a collapseable multinomial model. To do so we will restrict our model by assuming that every document is a member of one and only one topic. In order not to confused our approach with a traditional topic models, where each topic can take a weight $\theta \in (0, 1)$, we refer to the model in which a document can only belong to one topic as a *cluster membership model.* This also emphasizes the theoretical relationship between our model and finite mixture models.

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 and cluster membership. That is:

$$x_i \sim MN(q_{ij}, m_{ij}); \quad q_{ij} = \frac{exp(\alpha_j + y_i\phi_j + u_i\Gamma_{kj})}{\sum_{l=1}^{p} exp(\alpha_l + y_i\phi_l + u_i\Gamma_{kl})} \tag{4}$$

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

Now, the unigram collapsing can be expressed as x_{yk}, with

$$x_{yk} \sim MN(q_{yk}, m_{yk}), q_{yk} = \sum_{i:y_i=y,u_i=k} \Big[\sum_{j=1}^{p} x_{ij}\Big] \tag{5}$$

and

$$q_{yk} \sim \frac{exp[\alpha + y\phi + u_k\Gamma_k]}{\sum_{l=1}^{p} exp[\alpha_j + y\phi_j + u_k\Gamma_{kj}]} \tag{6}$$

Initializing Cluster Membership

Our focus in on predicting cluster membership u_i and the corresponding probability distortion Γ_i. Because each document can only be a member of one topic (unlike a traditional topic model), we want to investigate how important the cluster member initialization is to the final coeffecients. Forthis paper, we initiatilize 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, given predicted word count $\hat{X}$, clustering on $X - \hat{X}$

Estimation of Parameters via Maximum a Posteriori

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

$$\ell(\alpha, \phi, \Gamma, u) = \sum_{i=1}^{N} x_i^\top(\alpha + \phi v_i + u_i\Gamma_{kj}) - m_i log(\sum_{j=1}^{p} exp\Big[\alpha + \phi v_i + u_i\Gamma_{kj}\Big]) \tag{7}$$

Following previous literature on text regression (Taddy, 2013), We specify laplace priors and gamma hyperprior on coeffecients, as well as a gamma lasso penalty on coeffecients $c(\Phi, \Gamma)$. This procedure leads us to minimize

$$\ell(\alpha, \Phi, \Gamma, u) + \sum_{j=1}^{p} (\alpha_j/\sigma_\alpha)^2 + c(\Phi, \Gamma) \tag{8}$$

The procedure to fit the coeffecients of a multinomial with gamma lasso penalty are documented in Taddy (2013a). We now briefly detail the fitting of the cluster membership via Bayesian estimation:

$$\boldsymbol{u}^*_{MAP} = \arg\max_{\mathbf{u}} \;\; P(\mathbf{u}|X) \tag{9}$$

$$= \arg\max_{\mathbf{u}} \; \frac{P(\mathbf{u}|X)\, P(\boldsymbol{u})}{P(X)} \tag{10}$$

$$= \arg\max_{\mathbf{u}} P(\mathbf{u}|X)\, P(\boldsymbol{u}) \tag{11}$$

And under the assumption that the $P(\boldsymbol{u_i}) = P(\boldsymbol{u_j})$ for any two cluster vectors u_i, u_j, we have

$$\boldsymbol{u}^*_{MAP} = \arg\max_{\mathbf{u}} P(\mathbf{u}|X) \tag{12}$$

Although the likelihood function cannot be solved analytically, the discrete support for $\mathbf{u}$ makes it trivial to check the entire parameter space.

The basic algorithm we use to fit coefficients α, ϕ, Γ and cluster memberships u_i is two main steps iterated until convergence:

Algorithm for Cluster Membership Model with Gamma Lasso Penalty

1. Initialize u_i for $i = 1, \ldots, n$

2. Determine parameters α,ϕ, Γ by fitting a multinomial regression on $y_i|x_i, u_i$ with a gamma lasso penalty

3. For each document i, determine new cluster u_i membership as $argmax_{k=1,..,K}\Big[\ell(u_i|\alpha, \phi, \Gamma)\Big]$

4. Check if current cluster assignment is different from previous cluster assignment , $(\mathbf{u}^{(t)} = \mathbf{u}^{(t-1)})$.If so, return to step 2. If not, end algorithm.

By alternating between the first two steps, we aim to converge to optimal parameter estimates α, ϕ, Γ as well as optimal cluster membership u.

Computation

As noted previously, multinormal regression enjoyes the ability of being able to collapse observations across levels of metadata. This attractive property is preserved in the cluster membership model.

In step two, we can increase the speed of step two by only evaluating portions of the likelihood function relevant to u_i and Γ by eliminating first two terms from equation one:

$$L(u_i|\alpha, \phi, \Gamma) = \sum_{i=1}^{N} x_i^\top (u_i \Gamma_{kj}) - m_i log(\sum_{j=1}^{p} exp\Big[\alpha + \phi v_i + u_i \Gamma_{kj}\Big]) \tag{13}$$

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 and Evaluation of Algorithm

We applied the Cluster Membership model to two datasets; Congressional Speech records, most famously used to investigate media slant (Moskowitz and Shapiro, 2010) and a corpus of restaurant reviews called we8there.

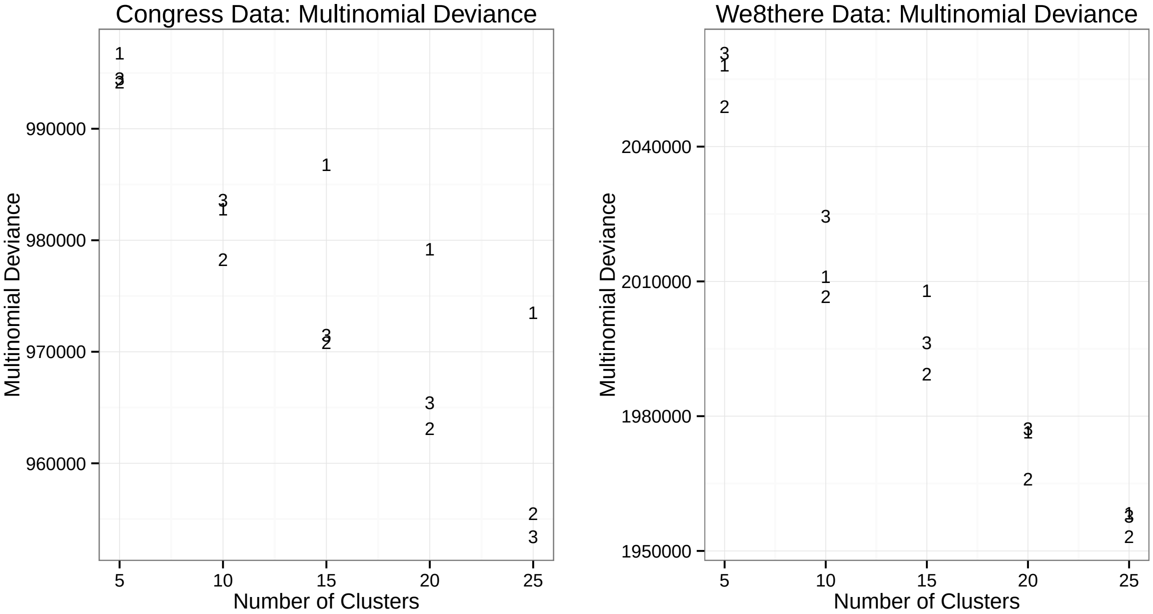
Congressional Speech

We first investigate the performance of the algorithm on the congressional speech data of. The data consists of text from 579 speeches of the members of the 109th Congress. For the analysis, party membership was regressed onto speech data. The algorithm was run for 5, 10, 15, 20, and 25 clusters, each over the 3 different initialization methods. We then report the multinomial deviance, or two times the negative log likelihood, in figure 1.

We note that, for any given number of clusters, a better-fitting model is almost always obtained by initializing the cluster memberships on the text data, compared to assigning each document a random cluster. We also note that initializing the cluster memberships to the residuals of the text data regressed on the outcome variable (in this case, GOP party membership) usually produces a worse-fitting model than simply initializing membership on the original text data.

Previous research (Taddy, 2012) has shown that the optimal number of topics for a topic model on this dataset is around 10. This fact is not shown in our data for a couple of reasons. First, we are possibly overfitting the data, since these models are not run under cross-validation. Second, due to its unigram design, the cluster membership model is much less flexible than a standard topic model, and would likely need many more clusters to model the same lexical variation.

Figure 1: Multinomial Deviance from fitted model cluster membership model



1 : random cluster initialization, 2 : K Means, 3 : K means on residuals

Interpretation of Topics

An essential aspect of topic modeling is determining the overall theme from the loadings vector. Previous research (Blei and Lafferty, 2009) shows that correctly-specified topic models allow for rich interpration of themes that can change over time, as well as model relationships between themes.

Of particular importance in our model is that our extreme simplication of topic weights does not result in "meaningless" topics. If the topics produced by our method bear no relation to the topics produced by more complex topic modeling approaches, then there is little benefit of our model, regardless of computational improvements and model simplicity.

To test topic fit, we compare topics produced from our method with the topics produced by fitting a topic-only model, via MAP, with 12 topics on Gentzkow and Shapiro's Congressional Data. Fortuneately, we find that topics from our model are, in many cases, similar to topics obtained by traditional methods. For example, the following table shows a "stem cell" topic found by our method, compared to a similar topic found using the topic-only model mentioned above.

Interpretation of results

We first evaluate our model by comparing the coeffecients predicting GOP to a gamma lasso regression without any topic models. The words with the highest loading for determing party

Table 3: Comparison of top word loadings on a stem-cell topic

Cluster Membership	Topic Model (LDA)*
umbilic.cord.blood	pluripotent.stem.cel
cord.blood.stem	national.ad.campaign
blood.stem.cel	cel.stem.cel
adult.stem.cel	stem.cel.line

*Results reported in Taddy (2012)

affiiliation are illustrated in Table 4.

Table 4: Words with highest loadings for predicting Republican-party affiliation

	Cluster Membership		Multinomial Regression	
	term	loading	term	loading
1	ready.mixed.concrete	9.25	un.official	5.47
2	driver.education	7.34	people.middle.east	5.47
3	speaker.table	7.2	speaker.table	5.47
4	medic.liability.reform	6.85	term.care.insurance	5.47
5	near.retirement.age	6.42	weapon.grade.plutonium	5.46
6	weapon.grade.plutonium	6.23	national.homeownership.month	5.46
7	death.tax.repeal	5.98	nation.oil.food	5.45
8	commonly.prescribed.drug	5.72	united.nation.oil	5.45
9	national.ad.campaign	5.69	national.heritage.corridor	5.44
10	national.homeownership.month	5.37	feder.air.marshal	5.42

*Mixed model fit with 15 topics, each topic initialized with K-means on the word count matrix

The theory behind our mixed topic model regression predicts that the topics should be able to incorporate a specific theme important to a group of individuals, leaving behind the more general predictors of party-affiliation to the regression coeffecient. We can test this prediction by examining terms that are significant for a Multinomial Regression without topics, but decrease in importance once topic models are added. For a simple illustration, we choose the phrase "nation oil food", which predicts affiliation with the republican party. The term is strongly associated with a topic we might call "domestic issues" topics, with the following high word loading terms:

We also notice that the members of this cluster (by our simplification, each observation can only be a member of one topic) are 3 republicans and 8 democrats. The ability to group observations that may have different metadata (in this case, political party) is a benefit of our mixed regression-topic model approach. Under standard multinomial regression, observations cannot be grouped by topics, whereas topic modeling does not immediately offer a computationally simple way to include influence on outside metadata.

	term	loading
1	nation.oil.food	20.09
2	united.nation.oil	12.09
3	liberty.pursuit.happiness	8.11
4	life.liberty.pursuit	8.11
5	minority.women.owned	6.73
6	universal.health	6.67
7	white.care.act	6.64
8	ryan.white.care	6.6
9	universal.health.care	5.99
10	growth.job.creation	5.39
11	drilling.arctic.national	5.3
12	tax.relief.package	5.29
13	judge.john.robert	5.26
14	fre.enterprise	5.07
15	arctic.refuge	4.93

One cluster from a model fit with 20-clusters , each having been initialized with K-means on the residuals of metadata regression

Comparison to Blei's Inverse Regression Topic Model

As mentioned in Blei (IRTM) the addition of latent topics to a model of text data with attributes is the ability to gain an intuitive concept to how metadata affects the distribution of a given topic. To demonstrate this effect, we use the graphic model presented in (Blei) to show the Democrat/GOP distortion to the "domestic issues" topic shown earlier.

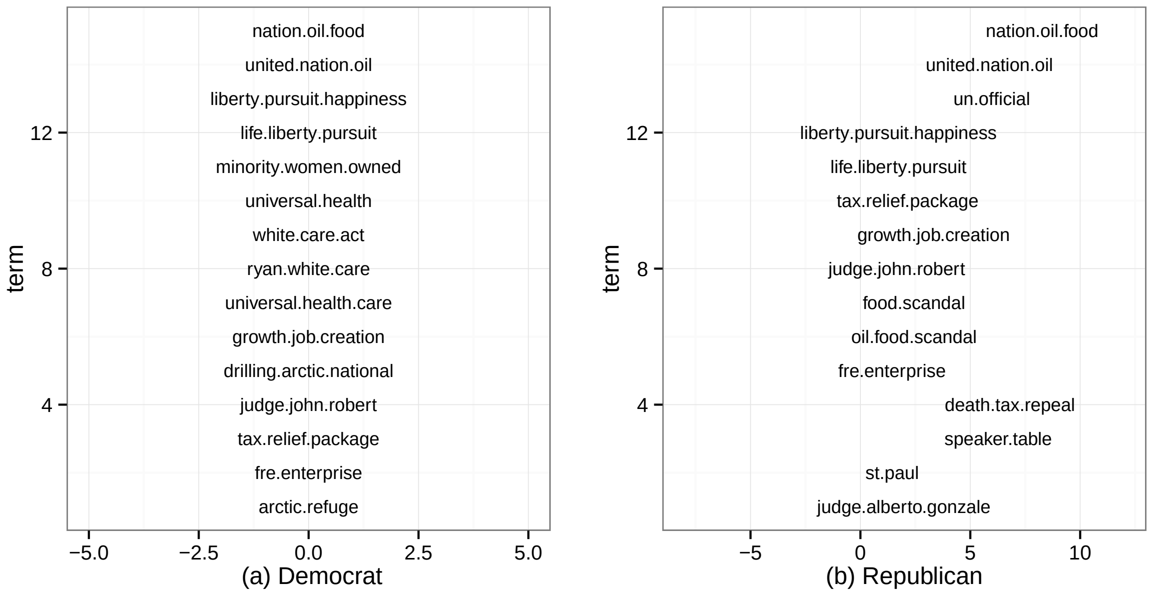
We8there Data

We briefly illustrate the results of the cluster membership model on the we8there corpus of restaurant reviews with a table of word loadings for a selected cluster, as well as a distortion graph of that topic. As with the Congress data, we note that despite the simplicity of our model, we obtain meaningful clusters. The following table illustrates a cluster we might label as "pizza diner".

As with the Congressional data, we also show the affect of metadata regression on the topic weights. However, the analysis for the food reviews data is different from the congressional data. Instead of a binary response (dummy variable for party affiliation), the covariate is the overall food review score, from 1 to 5.

The presence of the term "best mexican" in the "pizza diner" cluster distorted by a positive review illustrates one fundamental drawback of our model. Because we do not model relationship between cluster word loadings and covariate distortion word loadings, combining the two will predict collections of terms that may never appear together in the actual data.

Figure 2: Congress 109: Cluster word loadings with covariate distortion



A cluster from the Congressional data. On the left are, from highest to lowest, terms with the top Democratic word loadings. On the right are the words with the highest Republican word loadings. The horizontal position indicates the value of ϕ, or word distortion for Republicans

The shortcoming is also mentioned in other work on text analysis and metadata (Rabinovich and Blei, 2014), and possible solutions will be mentioned in the next section.

Extension

Feature Allocations

A promising extension of the cluster membership model is the generalized "feature allocation" (Broderick 2014), where each observation can be attributed multiple features. This setup is an intermediary between our cluster membership model and a traditional topic model. The algorithm provided in this paper could be extended to incorporate a feature allocation, although the order of step 3 of the algorithm increases from linear $\mathcal{O}_{cluster} = n_k$ to $\mathcal{O}_{feature} = 2^{n_k}$, where n_k refers to the number of clusters/features in the model. This increase in complexity may be reduced through changes to the algorithm, and the model would still enjoy the ability to collapse observations across unique features.

Cross Validation and Prediction

One drawback of our is that, in order to train cluster membership, the values of the metadata are required (see algorithm pseudocode in previous section). This hinder the ability to use

	term	loading
1	deep dish	7.76
2	italian beef	7.07
3	pizza like	6.85
4	style food	6.69
5	au jus	6.33
6	cut fri	6.16
7	just ok	6.01
8	great pizza	5.96
9	south side	5.94
10	pizza great	5.82
11	just over	5.75
12	took seat	5.72
13	golden brown	5.61
14	behind counter	5.58
15	got littl	5.52

One cluster from a model fit with 15 clusters , each having been initialized with K-means on the original text

the method for prediction (where, presumably, we hope to predict missing metadata from the text), and impeded our ability to perform cross validation. One possible solution for this is, after fitting a model with training data, use k-nearest neighbors or other appropriate algorithm to assign each document of the test data to a cluster. However, this approach would be difficult to combine with the feature allocation extension proposed above, since the n observations would be seperated into 2^{n_k} partitions, instead of n_k.

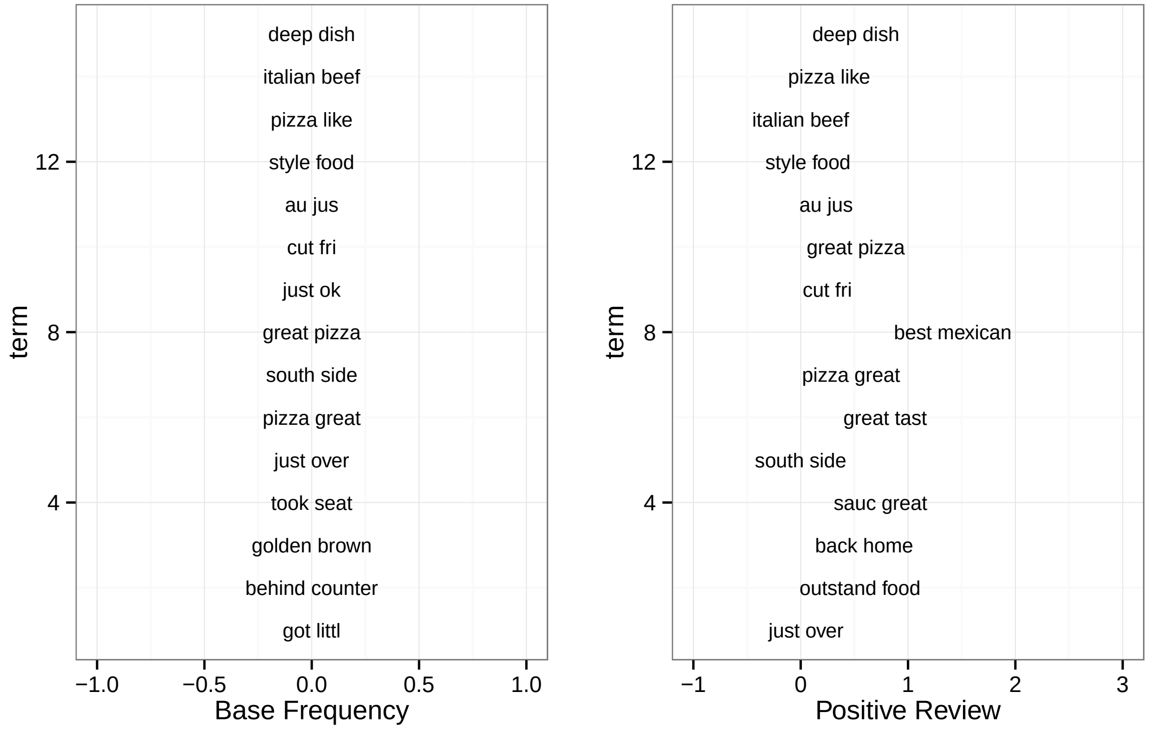
Modeling relationships between clusters and metadata

As noted in a previous section, one drawback of our model is the lack of relationship between clusters and metadata distortion vectors. Previous work on topic models allowed for correlation between topics (Blei & Lafferty 2006) and has successfully applied these models to complicated datasets (Blei & Lafferty 2007). Other authors have noted this problem (Rabinovich & Blei 2014), and possible solutions include allowing a range of metadata distortion vectors with varying relationships to each cluster or topic.

Conclusion

In this paper, we have reviewed the theory behind topic modeling and regression on metadata in text data. We introduce an algorithm that combines the metadata regression techniques developed by Taddy (2013a, 2013b) with a simple adaptation of the classic topic model. We

Figure 3: we8there: Cluster word loadings with covariate distortion



A cluster from the we8there restaurant reviews dataset. On the left are, from highest to lowest, terms with the top word loadings in this given cluster. On the right are the word loadings of the cluster altered by a positive review. The horizontal position indicates the value of ϕ, or word distortion for a positive review

then applied our algorithm to two large, sparse text datasets and report the results, noting that the simple cluster membership is able to identify similar topics found in more complex LDA models. We conclude by suggesting possible extensions to our approach to assist with generalizng and comparing the cluster membership model to alternative models for topic and metadata analysis.

Acknowledgements

I thank Professor Matt Taddy for his guidance throughout the research and writing of this paper. I also thank Professor Stephen Stigler for his feedback, the University of Chicago Research Computing Center (RCC) for computational resources, and Eric Janofsky for the helpful discussion.

References

[1] Blei, D. , Ng, A., and Jordan, M. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research (JMLR), 3:993–1022, March 2003. ISSN 1532-4435

[2] Blei, D. and Lafferty, J. (2006). Correlated topic models. Advances in Neural Information Processing Systems 18. MIT Press, Cambridge, MA.

[3] Blei, D. and Lafferty, J.(2006b). Dynamic Topic Models. Computer Science Department Paper 1038.

[4] Blei, D. and Lafferty, J. (2007). A Correlated Topic Model of Science. The Annals of Applied Statistics

[5] Broderick, T., Pitman, J., and Jordan, M. (2013). Feature allocations, probability functions, and paintboxes. Bayesian Analysis, to appear. Preprint arXiv:1301.6647, 2013

[6] Li, Cook and Tsai. Partial Inverse Regression. Biometrika (2007), 94, 3, pp. 615–625

[7] Taddy, M. (2012).On Estimation and Selection for Topic Models. Proceedings of the 15th International Conference on Artificial Intelligence and Statistics

[8] Taddy, M. (2013a) The Gamma Lasso. The gamma lasso. arXiv:1308.5623

[9] Taddy, M. (2013b). Multinomial inverse regression for text analysis. Journal of the American Statistical Association 108.

[10] Rabinovich, M. and Blei, D. (2014). The Inverse Regression Topic Model. Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32