Probabilistic modeling of tweets

Introduction

Recent years, microblogging like twitter is gaining more and more popularity over the world. People can share brief thoughts, pictures, and broadcast themselves on their smart phone wherever they go. The amount of information these bloggers post online is overwhelming and also valuable to some extent. For example on twitter, someone's interests, mood, favorite food and even which party he or she will vote for can be discovered by mining deep into their tweets. In machine learning and natural language processing, a topic model is a statistical model for uncovering the abstract "topics" that occur in a collection of documents. Here, in our final project for ST 740, we will build a Bayesian hierarchical topic model to discover hot topics posted by President Obama and discuss the goodness of fit of our model.

Methods

In year 2003, Blei, Ng, and Jordan introduced Latent Dirichlet Allocation (LDA) ([2]) which is a three-level hierarchical Bayesian model. The basic statistical assumption is "bag-of-words" - that the order of words in a document can be neglected. This is essentially an assumption of "exchangeability" for words in a document in Bayesian probability setup.
We will define the following terms following the definitions and notations in Blei, Ng, and Jordan paper.

- A word is the fundamental unit in a document. It is defined as an item from a vocabulary indexed by $1, \ldots, V$. Any word is represented using unit-basis vector of length V that has a single component with value one and others with value zero.
- A document is a sequence of N words denoted by $\mathbf{w} = (w_1, w_2, \ldots, w_N)$, where w_n is the nth word in the sequence

- A corpus is a collection of M documents denoted by $D = \{\mathbf{w.1}, \mathbf{w.2}, \ldots, \mathbf{w.M}\}$, where $\mathbf{w.m}$ is the mth document with N_m words, $\mathbf{w.m} = (w_1, w_2, \ldots, w_{N_m})$.

Given the definition of word document and corpus, Latent Dirichlet Allocation assumes that each document in a corpus D is generated as follows:

- Sample $N \sim \text{Poisson}(\xi)$.
- Sample $\theta \sim \text{Dir}(\alpha)$
- For each of the N words w_n:
  1. Sample a topic $z_n \sim \text{Multinomial}(\theta)$.
  2. Sample a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n

where β is a $k \times V$ matrix and $\beta_{ij} = p(w^j = 1|z^i = 1)$. It essentially describes the distribution of words given a topic.

To make the model simpler, we assume that the dimensionality k of the Dirichlet distribution (the number of topics present in the corpus) is known and fixed. Also, we assume that β is fixed and needs to be estimated. Note that N in the first layer of the hierarchy is independent of all the other data generating variables (θ and z), therefore, we ignore its randomness and assume the number of words in each document is known and fixed.

In this Bayesian hierarchical model, for a document with N words, we can write down the joint likelihood of topic mixture distribution $\theta = (\theta_1, \ldots, \theta_k)$, N words $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ and N topics $\mathbf{z} = (z_1, z_2, \ldots, z_N)$.

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

If we integrate over θ and sum over z, we can have the marginal distribution of a document:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) d\theta \tag{2}$$

Finally, posterior distribution:

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{3}$$

The posterior above is intractable and thus an analytic routine is not available. Here, Gibbs sampling comes to rescue ([4]).

In our model, β is assumed to be fixed but unknown, and thus needs to be estimated along the way. Here a basic Gibbs sampling scheme would be:

1. Initialize: randomly assign words to topics based on the pior, count the word frequency in each topic, initialize β from that.

2. Go through every word in each document and for each word w_n, $n = 1, 2, \ldots, N_m$ in document m:

   - For word w_n, pick a topic z_n from k topics that maximizes $p(\text{topic } z_n|\text{document } m)\times p(\text{word } w|\text{topic } z_n)$
   - $p(\text{topic } z_n|\text{document } m)$ is estimated by the proportion of topic z_n in document m
   - $p(\text{word } w|\text{topic } z_n)$ can be found by looking up the β matrix
   - After assigning a new topic to a word, update β and $p(\text{topic } z_n|\text{document } m)$

3. Iterate until the log-likelihood converges.

Analysis of real data from twitter

In this project, the most recent 5000 Tweets made by President Obama on twitter are parsed into R via a twitter API([3]). Tweets are plain text that may contain #somewords, @somebody, and http links. On twitter, People use the hash-tag symbol # before a relevant keyword in their Tweets to categorize those Tweets. We can treat the keyword after hash-tag as a user self-provided topic keyword and use that to check the accuracy of our topic model in discovering latent topics.

From these 1000 Tweets, the top 10 most frequent hash-tag keywords are shown in table (1):

Table 1: Hash-tag keywords

# Tag keyword	Count	# Tag keyword	Count
Obamacare	70	MakeCollegeAffordable	22
EnoughAlready	46	EndThisNow	19
immigration	43	GetTalking	18
ActOnReform	31	climate	16
ABetterBargain	28	ActOnClimate	15

One can see that the potential hot topics would be about Obama health-care, education, immigration reform, climate change, and middle class in America.

Then, all hash-tags are removed and with the help from a R package implementing Gibbs sampling([1]), a LDA model is built on top of these 1000 Tweets to uncover hot topics. For the Gibbs sampling, the number of topics is set to be 9 and 6000 iterations are run. The top 5 keywords in each topic are shown in table (2):

Table 2: Top 5 keywords among 9 topics

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
president	health	shutdown	watch	change	president	president	house	obama
obama	insurance	government	obama	climate	obama	obama	time	president
health	care	republicans	middle	congress	america	wage	congress	day
care	new	tell	class	heard	jobs	work	immigration	see
people	affordable	congress	live	today	country	minimum	pass	happy

The log-likelihood trace plot is shown in figure (1)

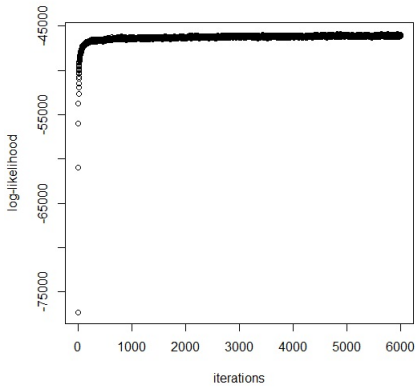

Figure 1: Log-likelihood trace plot

Conclusion

We compare the user self-provided hash-tag keywords with keywords discovered by LDA model. One can see that we can uncover almost all the topics including health care, immigration reform, climate change, and middle class in America except the education. However, each Tweet has only a few sentences and thus data from Twitter are much shorter than the typical text data analyzed by LDA. Typically LDA has very good results even with 25 iterations of Gibbs sampling. Here if we zoom in on the log-likelihood trace plot, we can see that the log-likelihood keeps increasing even after 1000 iterations. Therefore, there may be a more efficient method to analyze Tweets given their special structure (being brief). A good start would be the hash-tag keywords.

References

[1] Jonathan Chang. Package lda. *CRAN*.

[2] Michael I. Jordan David M. Blei, Andrew Y. Ng. Latent dirichelet allocation. *Joournal of Machine Learning Research*, 2003.

[3] Jeff Gentry. Package twitter. *CRAN*.

[4] T. GriNths and M. Steyvers. Finding scientific topics. *PNAS*.