Dynamic Topic Modeling : Election competition analysis using Twitter

Blerta Lindqvist Vedika Babu Shreepad Patil Harsha Tirumala
Department of Computer Science, Rutgers University
{ bl444, vb305, sp1467, hs675}@cs.rutgers.edu

Abstract

We design and implement (multiple) dynamic topic modeling schemes that leverage twitter data to identify the latent topics which multiple brands compete for. Specifically, we model the competition of multiple brands over shared topics and present a temporal evolution of this competition. We analyze the advantages/disadvantages of the candidate methods we implemented for revealing relevant topics. The competition we have monitored is the 2016 U.S. presidential elections in relation to the (major) candidates - Donald Trump and Hillary Clinton. Our work is primarily based on the ideas presented in the dynamic topic modeling project by Hao Zhang et. al. [17]

1. Introduction

Internet blogs, social networks, forums etc. provide a rich source of information online for the general public's opinions and inclinations. This information is very trustworthy since it is directly sourced from individuals - unlike the usually generic information that "samples" from surveys reveal. As a result, trends revealed from analysis on online sources (especially sites such as Twitter and Facebook) can be safely considered to be a true reflection of society. Despite surveys being more scalable, their low trustworthiness makes them a bad option for analyzing events like elections which have high personal-bias ; as opposed to, for example, product reviews which can be generalized to some extent.

We specifically address the issue of identifying the most discussed topics on Twitter in the build-up to, during and immediately after the 2016 U.S. presidential elections. The target is to build an automated system that retrieves the relevant tweets during that time, extracts the text-data and detects the most contentious topics. We further attempt to model the temporal evolution of the revealed topics.

Technically, we propose dynamic topic models that : (1) effectively represents text data, (2) models latent topics which multiple candidates compete for and (3) monitors the time evolution of these topics. We have implemented the following topic modeling schemes:
(1) LDA-HMM (LDA followed by HMM for time evolution)
(2) Dynamic LDA [3]
(3) Dynamic STC [17]
(4) a variant of the Affinity Propagation algorithm [6]

2. Prior Work

Due to the variety of brands, markets and online information sources nowadays, there has been a wealth of work in the direction of online market intelligence. Data has been leveraged from Amazon to study the relationship between product sales and review scores by [1]. Recommendation systems helping companies identify "important" users (for reviews) has been developed by [15]. In this work, we detect competitively shared election topics and monitor the temporal evolution of these topics.

Authors of [7] suggest a new dynamic topic model that tracks the temporal evolution of customers' interests and predict chances of purchasing items. Geography based topic modeling has been explored by [8] to suggest new "interesting" locations to users. Interactions between text and image data have been studied by [2] [4] and streaming algorithms for dynamic topic models developed by [5] [7]. The most relevant work to the proposed idea is the paper by Hao Zhang , Gunhee Kim and Eric P. Xing [17], on which our work is based. This work manages to handle the interdependence among the multiple brands simultaneously and is hence a better way of modeling the competitiveness over the shared topics.

The potential to improve upon this work stems from the fact that, traditionally, the Bag-of-Words representation of documents (tweets) does not exploit the semantic nature of the language - only focusing on the "counts" of words.

Word2vec [9] is a group of neural network based models used to create word embedding which converts words into vectors in a fixed dimensional space in such a way that the semantic relationships between these words are strongly captured. So, we intend to process our data using Word2Vec accordingly and hope to get an improved version of their method that outperforms its predecessor. We also apply a few different models to achieve the task and examine their suitability. We reveal our findings in sections 4 and 5.

3. Our Models

Overview

The main target of our models is to reveal the most discussed topics on Twitter prior to, during and immediately after the 2016 US presidential elections. We then analyze the temporal evolution of these topics. Below, we first present our schemes and a few important implementation details and later move on to analyzing the results.

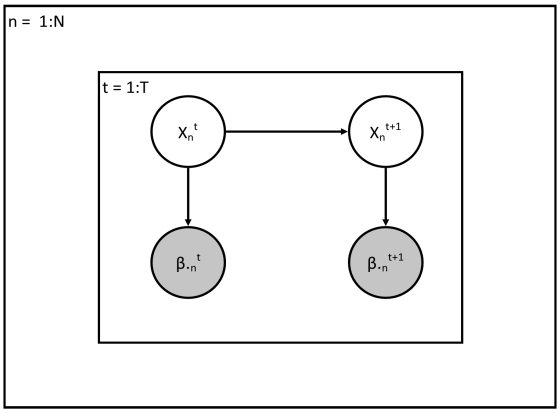
Representations

Let V be a vocabulary of N terms. In a bag-of-words model, a document d is represented as a vector $\mathbf{w}_d = \{w_{d1}, ..., w_{d_{|I_d|}}\}^{\mathcal{T}}$, where I_d is the index set of words that appears. Let B be a set of L competing brands. Let β denote a dictionary with K bases, of which each row β_k is an N dimensional basis. β_k is a topic, which is a unigram distribution over the terms in V. Here, $\beta_k \in \mathcal{P}$, where $\mathcal{P}$ is an $N-1$ simplex. $\beta_{.n} \in \mathbb{R}^K$ denoted the nth column of β.$\theta_d \in \mathbb{R}^K$ is the document code, while $z_{dn} \in \mathbb{R}^K$ is the latent representation for the nth word in document d. The vector $\mathbf{g}_d \in \mathbb{R}^L$ denotes the brands document d associates to. If two or more brands are associated with a document, we associate uniform weights.

3.1. LDA-HMM

Latent Dirichlet Allocation [2] [12] is a generative probabilistic topic modeling scheme wherein each document is modeled as a mixture over topics and each topic is modeled as a mixture over (word) probabilities. We have chosen LDA as one of the candidate methods as it is efficient and a standard for topic modeling.

In order to model the time evolution of the topics, we do the following sequentially:
(1) Apply PCA on the β(topic-word matrix) to reveal the top 20 words for each topic in each time slice.
(2) Apply Hidden Markov Modeling (HMM) to each column of this updated β to find the hidden layer (that can be used to predict future topics)



The plate diagram for the HMM on the updated β
$\beta_{.n}$ refers to the n th column of β

Implementation details

We have used the sklearn packages for applying LDA and HMM. For the HMM, we used a variant with multinomial distribution because our updated β is a binary (discrete) matrix.

3.2. Dynamic LDA (dLDA)

Dynamic LDA[3], proposed by David Blei and John Lafferty, is an extension of LDA that helps in modeling the temporal evolution of the topics of the corpus. While the order of words within a document is still insignificant, the order of the documents in the corpus is fundamental to dLDA. We chose dLDA as one of our candidate models because it seemed to be an improvement on the LDA-HMM model that we implemented.

Implementation Details

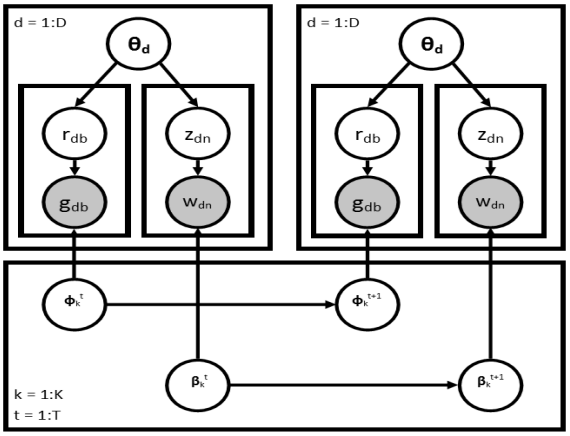
We have used the gensim package[13] to implement dLDA. In order to convert our data to the gensim corpora format, we used the Bleicorpus package in gensim.

3.3. Dynamic STC

Unlike LDA and its variants, Sparse Topical Coding [18] is a topic model framework that directly controls the posterior sparsity of the learned topics. STC learns coherent topic bases and identifies sparse topical senses of words. Since most tweets refer to very few topics (by virtue of tweets being short and sparse themselves), STC is perfectly suited to model such data.

The dynamic STC model[17] extends STC to handle streams of tweets. Twitter-Stream analysis is important to give context to the time line during which a certain topic was heavily discussed. It also addresses the problem of

detecting and tracking topics that are shared competitively by multiple brands (candidates). The Dynamic STC model, in addition, handles tweet streams. It also reveals the competitive share of each candidate over these topics.



The plate diagram for Dynamic-STC is given by the figure above.

Implementation details

The tweet streams that form our database are represented using the TF-IDF scheme[14] with the help of sklearn packages[11]. Each tweet is then labeled as belonging to one of the two candidates (Trump/Hillary); uniform weighing is used when both are mentioned. We have applied this method in two ways:
(1) On the complete dataset (3 million tweets/day)
(2) On a random sample of the data set (10000 tweets/day)

Our python code for implementing the dynamic STC is based on the works of [17] and [16].

3.4. Word2Vec-Fisher Vector-Affinity based propagation algorithm method

Affinity Propagation[6] is a clustering algorithm based on the concept of "link analysis" - similar to the Page-Rank algorithm. The idea is to estimate the exemplars (the best representatives of a cluster) for clusters based on a scoring function. Specifically, the algorithm proceeds by alternating between two steps iteratively:
(1) Updating how well-suited a candidate exemplar is
(2) Updating how appropriate is the candidate for a certain cluster

A significant advantage of this method is that the number of clusters need not be specified upfront. The algorithms stopping condition can be pre-specified. The final exemplars are those data points with a positive responsibility for themselves. Another advantage is of this method is that it considers all points as potential exemplars.

Implementation Details

In order to test how effective the Word2Vec representation [10][9](semantic word embeddings) is we transformed our corpus as follows:
(1) Use the gensim to convert the words in the corpus to word vectors
(2) Convert the obtained vectors into fisher clusters using the sklearn GMM implementation to enhance it.

Matlab implementation of the affinity propagation paper cannot handle the amount of data we have. Therefore, we used the sklearn implementation of the affinity propagation algorithm[11]. Consecutively, the preference values for each day from 11/03 to 11/12 were: 320, 225, 200, 170, 311 (9 clusters), 240, 200, 206, 320, 169. There were 10 clusters for each day except one noted above.

To impose a manageable number of clusters returned, we set the preference parameter to different values so as to get roughly 10 clusters for each day (9 for only one day). We chose K = 20 for fisher vector gmm mainly for running speed reason. Before running the affinity propagation algorithm, we run PCA so as to reduce the dimensionality of the data, while retaining at least 90% of the variance.

The affinity propagation algorithm only returns the cluster assignments. In order to define a topic out of a cluster, it would be helpful to have an ordering of the cluster words in terms, for example, of the distance from the cluster exemplar. The affinity based propagation algorithm can be changed to return the from-the-exemplar distances to enable the word ordering, using the same distance measure that is being used in the algorithm itself.

4. Results

4.1. Evaluation / Experimental Results

All the topics that the our schemes found are attached. A few of the topics have been manually annotated as well.

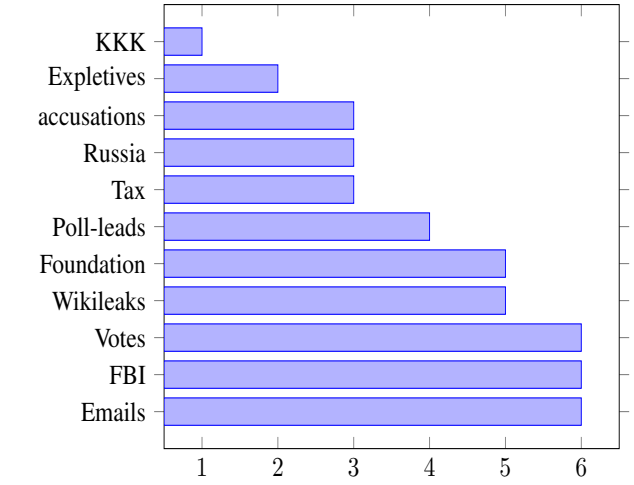
LDA , dLDA , dSTC

To check for the quality of results, we set our parameters such that each time slice(days from 11/3 to 11/12) has 10 topics. Within each topic, we printed only the top 20 words in the decreasing order of their significance.

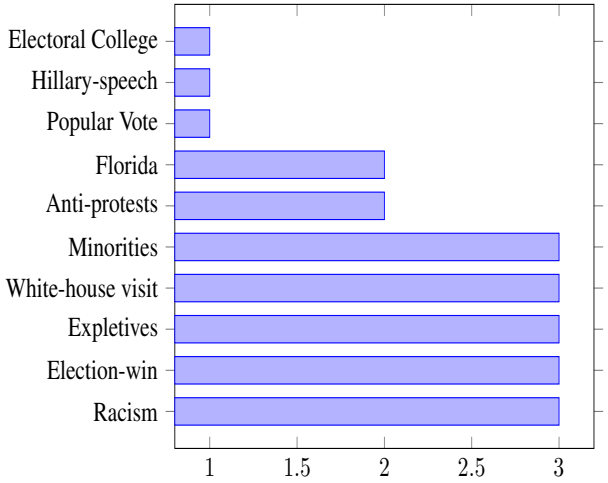
LDA-HMM

The running time of the LDA-HMM was 200 minutes (20 minutes per slice). The 10 topics for each of the days are displayed in the appendix. We tried to assign the top 20 words under each topic into election topics (like tax, expletives, FBI etc.). The mean coherence for each days' topics was about $\frac{8}{10}$.

Below we have presented the frequency of occurrences of the topics that were most discussed on Twitter in the time frame (11/3 to 12/3). The top pre-election topics (11/3 to 11/9) were markedly different from the top topics that appeared post-election (11/10 to 11/12). Most of the pre-election discussions surrounded on topics related to Hillary Clinton while all the top post-election topics were about Donald Trump.



PRE-ELECTION Topic-frequency (of 6 days)



POST-ELECTION Topic-frequency (of 4 days)

dLDA

The time taken by the dLDA scheme was about 2000 seconds (33 minutes). Very few of the topics that dLDA gave out were coherent. Initially, we ran the corpus including the urls. In this case, the urls that were deemed important by dLDA did not lead to coherent topics. Next, we ran it without the urls. In this case, the top 20 words did not yield any interesting topics.

dSTC

On the entire dataset, the dynamic STC scheme takes about 32 hours per time slice even with only 2 coordinate descent and 2 EM iterations. Since it was impossible in our time frame to complete it, we listed the topics that the first day (11/3) resulted in. Most of the topics could be annotated as the urls led to very specific issues.

We then randomly sampled 10000 tweets per day and ran the scheme with 5 coordinate descent and 5 EM iterations. It took about 2.5 hours per time slice to give results. In this run, we excluded urls. The resulting topics were not very coherent.

Affinity - Propagation

We gauged the affinity algorithm parameter so that the number of clusters returned would be 10 or close to 10 (9 in only one case). The exemplars throughout all days have been mainly short-form links - 28 out of 99. We have analyzed these links manually to get the content of the link since this was not done during the data gathering. All the content that was gathered was made lower-case from the beginning, therefore we had to google the links to find the content. Some links were no longer available. We analyzed in detail one day of the data to compare with the STC method results, for which we have the results of one day so far.

The summarized results for day 11/03 are:

11/3 Topics		
Clusters	Size	Top words
0	177	Voting, overwhelmingy for or against Trump
1	670	Trump, Melania, Wikileaks. Not very coherent topic
2	1418	Children, abuse, clinton, isis, trump. Not very coherent topic
3	1048	Clinton wikileaks email probe doj investigation
4	50	Anonymous Issues A Warning To Donald Trump! Very coherent topic
5	1411	Not clear topic
6	790	Mark Cuban, muslims, Melania
7	30	almost all are twits about Gary Johnson having no chill. Very coherent topic.
8	7167	Not clear topic
9	480	Hillary for prison. Very coherent topic

4.2. Discussion

4.2.1 LDA-HMM

LDA-HMM was able to reveal very specific topics like the Arizona scandal, Assassination attempt, Popopvich fears, NYC love march etc. One could have guessed the results of the elections by looking at the transition of the dominant brand (candidate) over the shared topics. Further, without prior knowledge of the exact election date (say only the week/month was known), it could have been easily guessed by observing the top-topics / words of top-topics.
The HMM applied on the resulting LDA topics would not yield appropriate temporal evolutions unless the topics across days are aligned to match (i.e. topic 1 of day 1 is the same as topic 1 in day 2 etc.).

Corrective Measures

The model can potentially be fixed by aligning the topics by similarity using appropriate distance function on the top words of topics.

4.2.2 dLDA

The temporal evolutions are appropriate since the model inherently tracks individual topics. However, the gensim implementation we used did not converge to coherent topics. Closer inspection and trials did not yield any better results. The anomalies with respect to the first dictionary word points to a flaw in the implementation.

Corrective Measures

Attempt to implement dLDA scheme ourselves instead of using the gensim package.

4.2.3 dSTC

The dSTC on the complete dataset resulted in very coherent topics (available from links). The dSTC also resulted in some very specific (easy-to-miss by models) topics like Democrats moving voters to Philadelphia, Grubhub resignations etc. But, only the top 5 words/links for topics were relevant to it - which we believe is due to only 2 iterations. Random initialization resulted in topics from other days showing up in inappropriate time slices - this is due to the β not fully converging.
The major drawback of the dSTC was the high run times required (about 20 times slower than the others). The random sampling experiment did not yield coherent topics. We also tried tweaking the hyper parameters. But, we could not get to any successful setting.

Corrective Measures

Attempt to find the "perfect" setting of the hyper parameters by relevant experimentation techniques. Alternatively, we could run a parallel execution of the algorithm to improve on the run time.

4.2.4 Affinity Propagation

We think the affinity based propagation algorithm needs to be tailored for the purposes of topic-based analysis so as to return an ordering of the words within the cluster based for example on the measure of distance that is being used; this would help in defining the topic better because we would know which words are closer to the exemplar and thus have more importance in defining the topic.

5. Conclusion and Future Work

We have designed and implemented multiple dynamic topic modeling schemes. We then compared the results we obtained. In the methods that we used, we noticed that the short links are present overwhelmingly as defining topics. We think this might be a result of the short links acting as succinct topics and as carrying more meaning than just words alone. So, urls cannot be discounted since the data is tweets (where most expression is through pointing to urls).
In the future, we plan on improving the model implementations by following the corrective measures we have discussed in the previous section.

References

[1] N. Archak, A. Ghose, and P. G. Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 56–65. ACM, 2007. 1

[2] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM, 2003. 1, 2

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006. 1, 2

[4] N. Chen, J. Zhu, F. Sun, and E. P. Xing. Large-margin predictive latent subspace learning for multiview data analysis. *IEEE transactions on pattern analysis and machine intelligence*, 34(12):2365–2378, 2012. 1

[5] G. Doyle and C. Elkan. Financial topic models. In *Working Notes of the NIPS-2009 Workshop on Applications for Topic Models: Text and Beyond Workshop*, 2009. 1

[6] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007. 1, 3

[7] T. Iwata, S. Watanabe, T. Yamada, and N. Ueda. Topic tracking model for analyzing consumer purchase behavior. In *IJCAI*, volume 9, pages 1427–1432, 2009. 1

[8] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 375–384. ACM, 2013. 1

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. word2vec, 2014. 2, 3

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 3

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 3

[12] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000. 2

[13] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en. 2

[14] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988. 3

[15] S. Wu, W. Rand, and L. Raschid. Recommendations in social media for brand monitoring. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 345–348. ACM, 2011. 1

[16] A. Zhang, J. Zhu, and B. Zhang. Sparse online topic models. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1489–1500. ACM, 2013. 3

[17] H. Zhang, G. Kim, and E. P. Xing. Dynamic topic modeling for monitoring market competition from online text and image data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1425–1434. ACM, 2015. 1, 2, 3

[18] J. Zhu and E. P. Xing. Sparse topical coding. In *UAI*, 2012. 2