LeadLag LDA: Estimating Topic Specific Leads and Lags of Information Outlets

Ramesh Nallapati, Xiaolin Shi, Dan McFarland, Jure Leskovec and Daniel Jurafsky
{nmramesh,shixl,mcfarla,jure,jurafsky}@stanford.edu
Stanford University, Stanford CA 94305, USA

Abstract

Identifying which outlet in social media leads the rest in disseminating novel information on specific topics is an interesting challenge for information analysts and social scientists. In this work, we hypothesize that novel ideas are disseminated through the creation and propagation of new or newly emphasized key words, and therefore lead/lag of outlets can be estimated by tracking word usage across these outlets.

First, we demonstrate the validaty of our hypothesis by showing that a simple TF-IDF based nearest-neighbors approach can recover generally accepted lead/lag behavior on the outlets pair of ACM journal articles and conference papers. Next, we build a new topic model called LeadLag LDA that estimates the lead/lag of the outlets on specific topics. We validate the topic model using the lead/lag results from the TF-IDF nearest neighbors approach. Finally, we present results from our model on two different outlet pairs of blogs vs. news media and grant proposals vs. research publications that reveal interesting patterns.

1 Introduction

The proliferation of a large number of information disseminating outlets presents several challenges to computational social scientists. One of the interesting problems is to identify which of the outlets leads the rest in dissemination of novel information. In addition, it is possible that an outlet may lead other outlets on certain topics, but may lag behind on other topics and we would like to track such topic-specific trends as well. Such analysis has several practical applications. For example knowing on what topics research funding (represented by successful grant proposals) lags behind scientific work (represented by academic publications) can help granting agencies readjust their allocation of funding to various fields of study. Knowing the topics in which blogs lead over news outlets may help information analysts track news better and faster.

2 TF-IDF Nearest Neighbors Approach

In this work, we hypothesize that novel ideas flow across communities through the creation and circulation of new or newly emphasized key words.

To test the validity of our hypothesis, we chose the simple TF-IDF based nearest neigbors approach owing to its simplicty and interpretability, as well as for its ability to capture distinguishing key words. Given two information outlets A and B, for each document d published in outlet A, the algorithm retrieves the most similar documents published in outlet B by key word usage, and compares their time stamps. If the nearest neighbors happen to be in the past with respect to document d's date of publication, it implies that it lags behind outlet B in terms of the concepts discussed in the document, and leads otherwise. The expected lag for document d is computed as a weighted average of the time differences with respect to its nearest neighbors where the weights are given by the respective similarity values. The mean lag of coprus A with respect to outlet B is then given by the average of lags of all documents in outlet A.

We also use a similarity threshold $\mathcal{T}$ below which we disregard the neighbors from lead/lag computation. This is done to avoid spurious matches since some documents published in outlet A may not have any counterparts in outlet B that discuss the same concepts. In addition, as candidates for nearest neighbors of document d in outlet A, we only consider documents from outlet B that are published within W time-units of the time of publication of d. This was done again to avoid spurious matches since our main goal is to capture temporally local propagation of novel information.

2.1 ACM Journal articles vs. Conference papers

It is widely accepted knowledge among the Computer Science (CS) research community that CS researchers typically publish novel ideas in conference proceedings. More often than not, journal articles are published either to elaborate the conference papers or to expand on the ideas of the conference papers. Note that these statements are specifically meant for Computer Science, and do not necessarily hold in other fields of study. When restricted to the publications of the same author, we can expect the journal articles to lag behind conference proceedings by around a year, since it takes roughly 3–5 months of effort to expand a conference paper and another 6–9 months to publish the journal article.

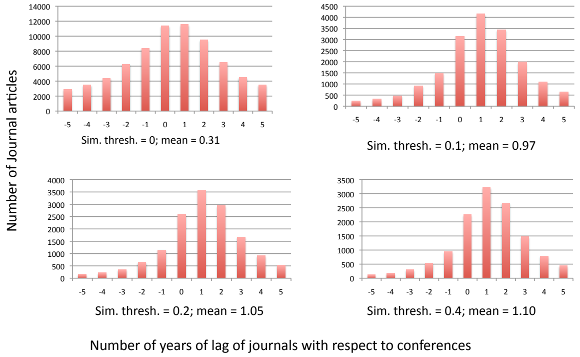
Our corpus consists of all ACM publications ranging from year 1952 to 2005[1]. In total, we have 99,677 journal publica-

[1] http://portal.acm.org/

tions and 103,191 conference papers. We used only abstracts of the papers in our experiments. Our preprocessing of the data included removing stop-words from a standard stop-words list, stemming the words using the Porter Stemmer[2], and removing the words that occur in less than 5 documents. We are finally left with 20,552 unique terms from journals alone. We discarded all terms from conferences data that do not occur in journals data.

We implemented the TF-IDF algorithm using the *Lucene* search engine[3]. The similarity is computed in terms of *Lucene's Practical Scoring Function*[4] between the TF-IDF weighted term vectors of the documents. After indexing all the conference abstracts, we converted each journal article into a query of at most 25 top TF-IDF words, and retrieved the conference abstracts that matched the query. We then scored these matches using the Lucene similarity function, and computed mean lag. In all our experiments below, we fixed the maximum number of nearest neighbors, N_{max}, to 5 and W to 5 years.

Figure 1 presents a histogram of lags of journal articles with respect to conference papers published by the same author for various values of the similarity threshold $\mathcal{T}$. The results are aggregated over multiple authors. Although there is no clear signal of lag of journal articles at $\mathcal{T} = 0$, the histogram starts shifting to the right as we increase $\mathcal{T}$ from 0 to 0.4. At a threshold of 0.4, the mean lag is approximately 1 year, which agrees quite closely with our intuition about the field of computer science research that it takes about 1 year for an author to expand a conference paper into a journal article. This result is in good agreement with the broad trends in computer science that experts agree upon, and therefore proves our hypothesis that key word usage across outlets can be used to estimate their leads and lags. In all our subsequent experiments, we set $\mathcal{T}$ to 0.4.



Sim. thresh. = 0; mean = 0.31

Sim. thresh. = 0.1; mean = 0.97

Sim. thresh. = 0.2; mean = 1.05

Sim. thresh. = 0.4; mean = 1.10

Figure 1: Lead/Lag histograms of ACM journal articles with respect to conference papers published by the same authors. The plots are for four different values of the similarity threshold $\mathcal{T}$. The histograms show that at reasonably strong thresholds, there is a clear signal that journal articles lag behind conference papers by approximately 1 year.

```
1.  For each document $d$:
2.    sample mixture over topics $\theta_d \sim \text{Dir}(\cdot|\alpha)$
3.    For each position $i$ in $1, \ldots, N_d$:
4.      sample topic $z_i \sim \text{Mult}(\cdot|\theta_d)$
5.      toss a biased coin $t_i \sim \text{Ber}(\cdot|\lambda)$
6.      if($t_i$ = 1)
7.        sample nearest neighbor $n \sim \text{Mult}(\cdot|\delta_d)$
8.        sample word $w_i$ from $\text{Mult}(\cdot|\beta_{nz_i})$
9.      else
10.       sample word $w_i \sim \text{Mult}(\cdot|\beta_{z_i})$
```

Table 1: Generative process of the Lead/Lag Topic Model. N_d is the length of document d.

3 LeadLag LDA

The TF-IDF nearest neighbor approach is effective at capturing overall lead/lag of corpora, but we are also interested in estimating lead/lag by specific *topics*. In this work, we propose a new topic model called LeadLag LDA that can capture topic specific leads and lags of knowledge outlets. The new model is a topic model analog of the nearest neighbors approach and works in the following 3 steps:

1. Learning step: In this step, we run the standard LDA model (Blei, Ng, and Jordan 2003) on all documents from outlet B, with respect to which we want to estimate lead/lag (e.g.: conferences data in the ACM outlets example), and learn the topics in the corpus in terms of the topic mixture prior α and the topic specific distributions over the vocabulary $\{\beta_1, \ldots, \beta_K\}$, and the word-to-topic assignments for all the documents in the corpus, where K is the number of topics.

2. Nearest neighbors step: For each document in outlet A, whose lead/lag we want to estimate (e.g.: journals data in the ACM outlets example), we identify its nearest neighbors in outlet B using the TF-IDF approach.

3. Inference using LeadLag LDA: Using the nearest neighbors for each document and the learned values of the LDA model on outlet B as the input, we perform inference on documents from outlet A using LeadLag LDA outlined in Table 1.

The new LeadLag model is similar to LDA in that it generates a topic assignment z_i for each word-position i in the document d from the document's mixture over topics given by θ_d (steps 1 through 4 in Table 1). However unlike LDA which samples the word w_i from the topic specific distribution β_{z_i}, the LeadLag model performs a biased coin toss with a probability λ (step 5 in the table). If the coin shows heads, it draws a neighbor n from a multinomial distribution δ_d over its neighbors and then draws a word from the neighbor using the probability β_{nz_iw} (steps 7 and 8 in the table) which is given by:

$$\beta_{nz_iw} = \kappa \frac{c_n(w|z_i)}{c_n(z_i)} + (1-\kappa)\beta_{z_iw} \qquad (1)$$

where $c_n(w|z_i)$ is the number of times that the word w is assigned to topic z_i in the neighbor document n, $c(z_i)$ is

[2] http://tartarus.org/∼martin/PorterStemmer/

[3] http://lucene.apache.org/java/docs/index.html

[4] http://lucene.apache.org/java/3_0_1/api/core/org/apache/lucene/search/Similarity.html

the document's total count of assignments of topic z_i, and κ is a smoothing parameter that is set to 0.9. Therefore, the model highly encourages the document to borrow topic specific language from one of its nearest neighbors. The probability δ_{dn}, which is topic independent, represents the likelihood that the document d used the same language as that in n. To complete the generative story, if the biased coin shows tails, the model reverts to the original generative process of LDA, in which the word is sampled from the learned distribution over the vocabulary β_{z_i} (step 10 in Table 1). The coin toss probability λ is a tunable parameter, which we set to 0.9 to encourage each document to reuse vocabulary from its neighbors as much as possible.

We estimate the parameters of the model using variational EM (Blei, Ng, and Jordan 2003), the details of which we skip owing to space constraints. Once we estimate δ_d, the expected lag of document d with respect to outlet B is given by:

$$Lag(d) = \sum_{n \in N(d)} \delta_{dn}(T(n) - T(d)) \tag{2}$$

The mean lag is then given by averaging the lags of all documents in outlet A. In addition to the mean lag which is topic independent, one could also compute topic-specific lags of the outlet A using the following equation:

$$Lag(A; k) = \frac{\sum_{(d \in A; (\theta_{dk} N_d) > 4)} Lag(d)\theta_{dk}}{\sum_{(d \in A; (\theta_{dk} N_d) > 4)} (\theta_{dk})} \tag{3}$$

where N_d is the length of document d. In other words, the topic-specific lag of outlet A on topic k is estimated simply as the weighted average of lags of all documents in A, where the weights are the relevance of the corresponding documents to the topic k. In the above equation, for each topic k, we only considered those documents for lead/lag estimation that have at least 4 words assigned to that topic in expectation.

We implemented LeadLag LDA by extending and modifying David Blei's LDA code in C[5]. We also built a multithreaded implementation of this code that allows us to scale the model to the large corpora we used in our experiments.

3.1 Evaluation of LeadLag LDA

Since we do not have any ground truth labeled data in terms of lead/lag for topics, we use the results from the TF-IDF nearest neighbors model as the ground truth for evaluation purposes. This is a reasonable approximation, since we already validated the TF-IDF nearest neighbors model on the ACM Journals vs. Conferences data. The LeadLag model can estimate topic-specific lead/lags using Eq. 3, but one could also compute topic-independent lead/lag by using Eq. 2, which could be compared with the values generated by TF-IDF nearest neighbors approach.

Figure 2 shows the lag estimates of the TF-IDF model and LeadLag LDA at various values of the similarity threshold $\mathcal{T}$. Both the curves are in good alignment, validating LeadLag LDA as an accurate model for lead/lag analysis.

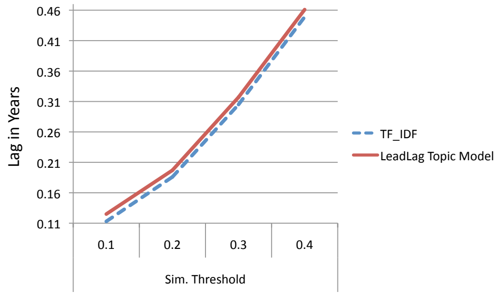


Figure 2: Comparison of lag estimated by LeadLag LDA with that of TF-IDF model as a function of threshold. Estimates of LeadLag LDA align very closely with those of TF-IDF.

We also compared the log-likelihood estimates of LeadLag LDA with those of LDA on the ACM journals data. We found that LeadLag LDA is able to outperform LDA at $\mathcal{T} > 0.2$ because it is able to learn better from the additional information of high quality neighbors. At lower thresholds, the nearest neighbor matches are more noisy, resulting in poorer predictive power of the model. Also, for higher number of topics, the LeadLag LDA suffers from sparsity of topic-specific information, and is therefore unable to outperform LDA until relatively higher thresholds are reached.

4 Results and Discussion

4.1 CS Grants vs. Science

We ran a 50 topic LeadLag LDA on the twin outlets of grants and science. The grants outlet is represented by successful NSF grant proposals[6] , while 'science' is approximated by all publications from the ISI dataset[7]. We focused our analysis in the area of Computer Science. From the ISI dataset consisting of most academic journal publications since 1960's, we extracted abstracts from Computer Science publications based on the "Field" labels, which resulted in 471,553 documents. A vast majority of the these documents are uniformly distributed in the timespan between 1991 and 2008. We also have successful grant proposals data from NSF whose awards are mostly from year 1990 to 2009. We extracted all Computer Science abstracts from this dataset using the NSF program names, which resulted in 12,388 abstracts. After stopping and stemming the NSF corpus, and removing words that occur in less than 5 documents, we ended up with a vocabulary of 8,326 unique terms. We used the same vocabulary in the ISI corpus as well, and discarded any terms unseen in the NSF corpus.

Figure 3 shows the lag of grants with respect to science in Computer Science on various topics. The plot shows that NSF grants lag behind science on topics in Computer Science such as 'Information Retrieval', 'Computer Aided Health Care', 'Mobile Networks', and 'Network Security', by around a year, but has a slight lead on other topics such as 'Databases' and 'Algorithms'.

[5] http://www.cs.princeton.edu/ blei/lda-c/

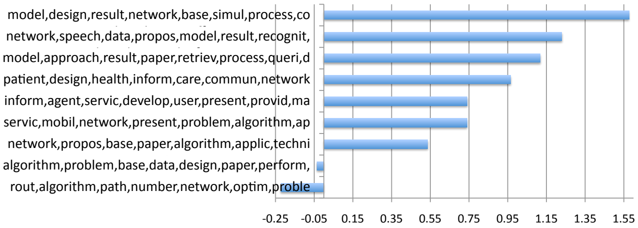


Figure 3: Number of years by which NSF grants lag behind ISI papers resolved by topics in Computer Science. Topics are described in terms of their top 10 most likely terms. NSF leads on 'Databases' and 'Algorithms' (last two bars in the histogram) but lags behind on other topics of Computer Science such as 'Speech Recognition' (second bar from top), 'Computed Aided Health Care' (fourth bar fron

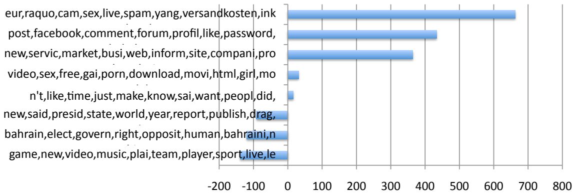


Figure 4: Number of seconds by which news lags behind blogs resolved by topics. Topics are described in terms of their top 10 most likely terms. News leads blogs on topics such as 'Sports' (bottom most bar), 'Politics' (second bar from bottom), but lags behind blogs on topics such as 'Adult content' (first and fourth entries from top), and 'Business' (third bar from top).

4.2 News vs. Blogs

For this run, we used a subset of the Spinn3r index[8] that consists of all entries from the entire day of 2010-10-22. The dataset has a variety of outlet types including blogs, news, *Twitter* feeds and *Facebook* postings. We identified news stories by matching the URLs of the posts with a list of 18,615 URLs we indexed from *Google News*[9]. All other postings that do not have the words 'twitter' or 'facebook' in their URLs are treated as blog entries. After removing posts that are less than 10 words long, we are left with 1,769,228 blog postings and 247,543 news stories. We pruned the vocabulary using the standard procedure on the news corpus, which gave us 233,442 unique words. We trained a 25 topic LDA (to minimize the computational effort and to reduce sparsity for topical lag computation) on the blogs corpus and ran inference using LeadLag LDA on the news corpus. Figure 4 lists the topic-specific lags of news for a few selected topics. The results show that while the news outlets lead on traditional news topics such as 'Sports' and 'Politics', blogs lead on 'Business' and 'Adult Content'.

5 Related Work and Conclusion

Recently (Ramage, Manning, and McFarland 2010) presented LDA based techniques to model which universities lead the rest using similarity in the topic space. Their approach, though highly related in intent to ours, does not model lead/lag by specific topics. In the topic modeling family, the work of (Gerrish and Blei 2010) comes closest to our work. In this paper, the topic specific impact of a document is modeled in terms of the document's 'language' that is reused by other documents published in its future. However, the goal of their work is modeling impact of individual documents, while we are interested in modeling the lead/lag of outlets in disseminating novel information.

LeadLag LDA is related to the Citation Influence model (Dietz, Bickel, and Scheffer 2007) in terms of its broad design. The goal of the Citation Influence model is to capture the relative importance of a document's citations in influencing the document's content. Their approach was to allow each document to 'copy' topics for its word generation from the cited documents. The Lead/Lag topic model has an analogous goal of capturing the relative influence of its nearest neighbors on its own content. However, we address the problem differently, by requiring the document to 'copy' from its neighbors topic specific words explicitly, rather than topics themselves. This constrains the model to capture similarity of documents in terms of topic-specific word usage, rather than the looser requirement of topical similarity.

In this work, we empirically validated our hypothesis that lead/lag of outlets can be captured by tracking usage of key words by testing a simple text based TF-IDF model on ACM Journals vs. Conferences outlet pairs where general agreement on lead/lag behavior exists. We also built a new LeadLag topic model that can compute lead/lag by topics, and validated it against the TF-IDF model. The output of the model on three different outlet pairs presents interesting insights into their topic-specific lead/lag behavior.

Although there has been considerable work in the past in terms of modeling influence using network information, research on using textual data to model diffusion of information is only beginning to emerge. We hope that this work encourages other researchers to pursue this promising line of research more vigorously.

Acknowledgments

This research was supported by NSF grant NSF-0835614 CDI-Type II: What drives the dynamic creation of science? We wish to thank our anonymous reviewers for their deeply insightful comments and feedback.

References

Blei, D.; Ng, A.; and Jordan, M. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:9931022.

Dietz, L.; Bickel, S.; and Scheffer, T. 2007. Unsupervised prediction of citation influences. In *Proceedings of the 24th International Conference on Machine Learning*.

Gerrish, S., and Blei, D. 2010. A language-based approach to measuring scholarly impact. In *International Conference on Machine Learning*.

Ramage, D.; Manning, C. D.; and McFarland, D. A. 2010. Which universities lead and lag? toward university rankings based on scholarly output. In *NIPS Workshop on Computational Social Science and the Wisdom of the Crowds*.

[6] http://www.nsf.gov

[7] http://www.isiknowledge.com

[8] http://www.icwsm.org/data/; http://www.spinn3r.com/

[9] http://news.google.com