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 validity 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. Besides, such a study could also help social scientists in analyzing how information spreads across communities and outlets.

In this work, we hypothesize that novel ideas flow across communities through the creation and circulation of new or newly emphasized key words. If this hypothesis holds, then tracking the usage of key words across outlets can help us detect the proponents of novel information. Our contributions in this paper are three fold:

1. We demonstrate the validity of the ‘novel information through novel key words’ hypothesis using a simple TF-IDF nearest neighbors approach that recovers the generally accepted lead/lag behavior between journal articles and conference papers in Computer Science.

2. We propose a new topic model for estimating topic-specific lead/lag of outlets and validate the model using the TF-IDF output as the ground truth.

3. We present and analyze the results of our model on two different pairs of outlets, namely, NSF successful grant proposals vs. ISI research publications and news wire vs. blogs.

2 Word usage can reveal lead/lag patterns
In this section, we test the validity of our assumption that word usage can be used to estimate the lead/lag of information outlets. For this purpose, we need a pair of outlets between which we know the ground truth of lead/lag behavior. After considerable search, we narrowed down on the Computer Science outlets of journals and conferences.

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. 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 get the paper published in a journal. When we compare related journal articles and conference papers, we often find that the journal articles are published either to elaborate the conference papers or to expand on the ideas of the conference papers. 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 get the paper published in a journal. We compare related journal articles and conference papers based on independent conversations with top CS academics XXX, YYY and ZZZ (names withheld for anonymity). Note that these statements are specifically meant for Computer Science, and do not necessarily hold in other fields of study.

Due to the relative novelty of this work, we have found it very hard to find a corpus with known lead/lag behavior of outlets.

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
ence papers without the same-authorship restriction, we still expect the journal articles to lag behind on an average, but by a smaller period. This is because of the interleaving effect, where the conference paper of one author may be inspired by the journal article of another author, thereby reducing the mean lead of conference papers.

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

The data comes with information on the authors of the publications but without resolving them into unique IDs. We implemented a simple but effective entity resolution algorithm that takes into account variations in naming conventions as well as co-occurrences of authors. Based on manual inspection of known entities, we found that our algorithm gave us high precision as well as very good recall. The authorship information will play an important role in our experiments, as we describe in section 2.2 below.

### 2.2 TF-IDF nearest neighbors approach

To test the validity of our hypothesis, we chose the simple TF-IDF based nearest neighbors approach owing to its simplicity and interpretability, as well as for its ability to capture distinguishing key words. For each journal article, the algorithm retrieves the most similar conference papers by key word usage, and compares their time stamps. The similarity is computed in terms of Lucene’s Practical Scoring Function between the TF-IDF weighted term vectors of the journal article and the conference paper (Salton and Buckley 1988). If the nearest conference papers happen to be in the past with respect to the journal article’s date of publication, it implies that the journal article lags behind the conference papers in terms of the concepts discussed in the article, and leads otherwise. The expected lag for each journal article 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 journal articles with respect to conference papers is then given by the average of lags of all journal articles. The approach is presented more formally in Table 1.

Note that we also use a similarity threshold $\mathcal{T}$ below which we disregard the neighbors from lead/lag computation, as shown in step 1 of Table 1. This is done to avoid spurious matches since some journal articles may not have any counterparts in the conference proceedings that discuss the same concepts. In addition, we only consider conference papers published within $W$ time-units of the time of a journal article’s publication. This was done again to avoid spurious matches since our main goal is to capture temporally local propagation of novel information.

We implemented the algorithm using the Lucene search engine. 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 as described in Table 1. In all our experiments below, we fixed $N_{\max}$ to 5 and $W$ to 5 years.

#### Author-specific Lead/Lag

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.

#### General Lead/Lag

The histograms of lead/lag of journal articles unconstrained by authorship information are shown in Figure 2. The pattern looks very similar to the author-constrained analysis above, but the value of mean lag is only half of what we see in the author-constrained data. This again is in line with our intuition that the general lag of journals should be lower than the author-specific lag due to the interleaving effect.

The results from our twin experiments above are in good agreement with the broad trends in computer science that experts agree upon, and therefore prove our hypothesis that key word usage across outlets can be used to estimate their leads and lags.

---

4. [http://tartarus.org/~martin/PorterStemmer/](http://tartarus.org/~martin/PorterStemmer/)
1. For each document $d$ in outlet $A$:

2. compute $N(d)$ nearest neighbors from outlet $B$ s.t. $\forall n=1^N(d)$ $(sim(d, n) \geq T$ & $|T(n) - T(d)| < W$) & $N(d) \leq N_{max}$,

3. if $N(d) > 0$, estimate the lag of the journal article in terms of its neighbors as: $\text{Lag}(d) = \frac{\sum_{n=1}^{N(d)} (T(d) - T(n)) \times \text{sim}(d, n)}{\sum_{n=1}^{N(d)} \text{sim}(d, n)}$.

4. Estimate Mean Lag = $\frac{\sum_{d \in N(d) > 0} \text{Lag}(d)}{\sum_{d \in N(d) > 0}}$.

Table 1: TF-IDF nearest neighbors approach for estimating lag of outlet $A$ with respect to outlet $B$. The similarity $sim(d, n)$ is computed in terms of the cosine of the angle between the TF-IDF weighted word vectors of the documents from outlet $A$ and outlet $B$. $T$ is the similarity threshold which is a tunable parameter of the model, while $T(n)$ is the time-stamp of the document $d$ from outlet $A$ and $T(n)$ is the time-stamp of its nearest neighbor from outlet $B$. Only the documents in outlet $B$ published $W$ time-units before or after the time of publication of each document in outlet $A$ are considered for its nearest neighbor analysis. Finally, $N_{max}$ is the maximum number of nearest neighbors allowed per document.

Figure 2: Unconstrained lead/lag histograms of ACM journal articles with respect to conference papers. The plots are for four different values of similarity threshold $T$. The histograms show that even when unrestricted by authorship, journal articles still lag behind conference papers, but by half the time-period as the lag of journal articles with respect to conference papers published by the same author.

Choosing optimal parameters of the TF-IDF model

Choosing Optimal $T$: From our experiments, it appears that the optimal value of $T$ should be around $0.3$–$0.4$ since those are the values that generate the commonly accepted values of mean lags. Since this only offers a ballpark value, we first manually inspected the abstracts of nearest neighbor pairs at various threshold values. We found that as the similarity value gets higher, we get more accurate matches.

At a threshold of $0.40$, we start seeing neighbor pairs that are highly likely to discuss the same concepts. However, there is a downside to choosing higher thresholds: firstly, the data becomes sparser at higher thresholds. Secondly, as the plot in Figure 3 demonstrates, even in the case of author-independent lead/lag analysis, an increasing proportion of the neighbor pairs end up being the works of the same author. In other words, at high thresholds, we will capture average trends of author-specific behavior and not general outlet behavior. We decided to choose $0.4$ as the optimal value for the similarity threshold $T$, since it not only captures neighbor pairs that discuss similar concepts with high accuracy, but also limits the proportion of neighbor pairs with same-authorship at $25\%$, allowing us to capture general outlet level trends. In addition, at a threshold of $0.4$, the mean lag of $1$ year for author-specific analysis remains our general agreement of lag of journals with respect to conferences.

Further, it would be reasonable to assume that the similarity threshold transfers well across corpora because Lucene’s Practical Scoring Function that our similarity is based upon, is normalized for several variations across corpora such as document length, query length, etc.

Choosing the optimal number for $N_{max}$: In Figure 4, we plotted the average lags as a function of the threshold for various numbers of nearest neighbors $N_{max}$. As indicated by the plot, at higher number of neighbors, the lag estimates are consistently smaller across all thresholds, since the lags are ‘smoothed’ by more neighbors. However, the difference in lags for various values of $N_{max}$ reduces asymptotically at higher thresholds, especially at thresholds $\geq 0.4$. Since we chose $0.4$ as the optimal threshold, the mean value of lag is approximately independent of $N_{max}$. We chose $5$ as the best value of number of nearest neighbors for computational efficiency.

Figure 3: General lead/Lag analysis: plot of mean lag in years (solid curve) and proportion of nearest neighbor pairs above the threshold that share the same author (broken curve) as a function of similarity threshold $T$.

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.
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 \(\text{Mult}(\cdot|\beta_{nz_i})\)
      else
         9. sample word \(w_i \sim \text{Mult}(\cdot|\beta_{zi})\)

Table 2: Generative process of the Lead/Lag Topic Model. \(N_d\) is the length of document \(d\).

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 \(\alpha\) 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 outlined in Table 1.

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 3 and graphically displayed in Figure 5.

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 \(\theta_d\) (steps 1 through 4 in Table 3). However unlike LDA which samples the word \(w_i\) from the topic specific distribution \(\beta_{zi}\), the LeadLag model performs a biased coin toss with a probability \(\lambda\) (step 5 in the table). If the coin shows heads, it draws a neighbor \(n\) from a multinomial distribution \(\delta_d\) over its neighbors and then draws a word from the neighbor using the probability \(\beta_{nz_iw}\) (steps 7 and 8 in the table) which is given by:

\[
\beta_{nz_iw} = \frac{c_n(w|z_i)}{c_n(z_i)} + (1 - \kappa)\beta_{ziw}
\]

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 the document’s total count of assignments of topic \(z_i\), and \(\kappa\) is a smoothing parameter that is set to 0.9. In other words, to generate a word conditioned on topic \(z_i\) in document \(d\), the model picks one of its neighbors \(n\) with probability \(\delta_{dn}\) and then samples one of the words in the neighbor that has been assigned the same topic \(z_i\). Therefore, the model highly encourages the document to borrow topic specific language from one of its nearest neighbors. The probability \(\delta_{dn}\), 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 \(\beta_{zi}\) (step 10 in Table 3). The coin toss probability \(\lambda\) is a tunable parameter, which we set to 0.9 to encourage each document to reuse vocabulary from its neighbors as much as possible.

Given the model’s parameters, the log-likelihood of the observed data in corpus \(A\) is given by:

\[
\log P(w) = \sum_{d=1}^{M} \sum_{i=1}^{N_d} \sum_{k=1}^{K} \sum_{n \in N(d)} \delta_{dn} \beta_{nkwi} + \sum_{d=1}^{M} \sum_{i=1}^{N_d} \log \theta_d + \sum_{d=1}^{M} \sum_{i=1}^{N_d} \log (\alpha)
\]
\[ (1 - \lambda)\beta_{kw_i} + \lambda \beta_{kw} \]  
\[ \frac{P(\theta_d|\alpha)}{\theta_d} \]  
\[ (2) \]
Since the estimation of the parameters is intractable by exact inference, we use variational approximations following standard procedure described in (Blei, Ng, and Jordan 2003). Accordingly, the approximate variational posterior is defined as:
\[ P(z|w) = \prod_{d=1}^{M} Q(\theta_d|\gamma_d) \prod_{i=1}^{N} (\lambda_i + (1 - \lambda_i)) \left( \sum_{n \in N(d)} \delta_{din} \right) \phi_{diz}, \]  
\[ (3) \]
where \( \lambda_i \) is the posterior probability for document \( d \) of sampling words from one of its nearest neighbors, \( \delta_{din} \) is the posterior probability that the document \( d \) picks neighbor \( n \) to draw a word at position \( i \), \( \phi_{dik} \) is the probability of topic \( k \) at position \( i \), and \( Q(\theta_d|\gamma_d) \) is the posterior Dirichlet distribution over the topic mixture for document \( d \). In the above equation. Although the terms \( (\lambda_i + 1 - \lambda_i) \) and \( (\sum_n \delta_{din}) \) trivially sum to 1, they do play an important role in approximating the original log-likelihood.

Using variational EM (Blei, Ng, and Jordan 2003) results in the following update equations:
\[ \phi_{dik} \propto \exp(E_Q(\log \beta_{dk}) + \lambda_i \sum_n \delta_{in} \log \beta_{nkw_i}) \]  
\[ + (1 - \lambda_i) \log \beta_{kw} \]  
\[ (4) \]
\[ \delta_{din} \propto \delta_{dn} \exp(\sum_{k=1}^{K} \phi_{dik} \log \beta_{nkw}) \]  
\[ (5) \]
\[ \lambda_d = \log(\log \lambda \frac{1}{1 - \lambda} + \frac{1}{N_d} \sum_i \delta_{in} \log \delta_{dn} - \log \delta_{din}) \]  
\[ + \sum_{i=1}^{N_d} \phi_{dik} \sum_n \delta_{din} \log \beta_{nkw_i} - \log \beta_{kw_i}) \]  
\[ (6) \]
\[ \delta_{dn} \propto \sum_{i=1}^{N_d} \delta_{din} \]  
\[ (7) \]
Once we estimate \( \delta_{d} \) using equation 6, the expected lag of document \( d \) with respect to corpus \( A \) is given by:
\[ \text{Lag}(d) = \sum_{n \in N(d)} \delta_{dn}(T(n) - T(d)) \]  
\[ (8) \]
The mean lag is then given by averaging the lags of all documents in outlet \( A \). Note the analogy of this equation with the equation in step 3 of Table 1.

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:
\[ \text{Lag}(A; k) = \frac{\sum_{d \in A; (\theta_{dk}, N_d) > 4} \text{Lag}(d; \theta_{dk})}{\sum_{d \in A; (\theta_{dk}, N_d) > 4} \theta_{dk}} \]  
\[ (9) \]
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. From our manual inspection, we found that any document with less than 4 words assigned to a given topic does not adequately represent that topic and is therefore not a reliable estimator of lag for that topic.

We implemented LeadLag LDA by extending and modifying David Blei’s LDA code in C\(^7\). We also built a multi-threaded 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. 9, but one could also compute topic-independent lead/lag by using Eq. 8, which could be compared with the values generated by TF-IDF nearest neighbors approach.

Since LeadLag LDA uses the nearest neighbors identified by the TF-IDF model as input, one could argue that the evaluation is biased in favor of LeadLag LDA. However, there is still no reason for LeadLag LDA to outperform the same lead/lag values as TF-IDF since the model uses the learned values of \( \delta_{d} \) in estimating the lag for each document (see Eq. 8) and is completely unaware of the TF-IDF similarity values\(^8\). Figure 6 shows the lag estimates of the TF-IDF model and

![Figure 6: 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.](http://www.cs.princeton.edu/blei/lda-c/)

\(^7\)http://www.cs.princeton.edu/blei/lda-c/

\(^8\)In the trivial case where each document has only one neighbor, LeadLag LDA will produce exactly the same value of lag as the TF-IDF approach since there is nothing to learn in \( \delta_{d} \); it would be equal to 1 for the lone neighbor. But we empirically observed in the ACM data at \( T = 0.4 \) that 50% of the documents that had neighbors had strictly more than 1 neighbor.
We also compared the log-likelihood estimates of LeadLag LDA with those of LDA on the ACM journals data. Table 3 shows the log-likelihood values for various values of similarity thresholds and for two different numbers of topics. We only aggregated the log-likelihood values for journal articles that had neighbors at a given value of similarity threshold, since LeadLag LDA reduces to LDA for documents with no neighbors. This explains why LDA has different values at different similarity thresholds for a given number of topics, although LDA computes likelihood purely based on the document’s content.

<table>
<thead>
<tr>
<th>Sim. Threshold</th>
<th>50 Topics</th>
<th>100 Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LeadLag LDA</td>
<td>LDA</td>
</tr>
<tr>
<td>0.1</td>
<td>-8.76E+06</td>
<td>-8.71E+06</td>
</tr>
<tr>
<td>0.2</td>
<td>-3.40E+07</td>
<td>-3.38E+07</td>
</tr>
<tr>
<td>0.3</td>
<td>-1.80E+07</td>
<td>-1.83E+07</td>
</tr>
<tr>
<td>0.4</td>
<td>-9.80E+06</td>
<td>-1.03E+07</td>
</tr>
<tr>
<td>0.5</td>
<td>-5.96E+06</td>
<td>-6.52E+06</td>
</tr>
</tbody>
</table>

Table 3: Comparison of Log Likelihood of basic LDA and LeadLag LDA on ACM Journals data for 50 and 100 topics. Bold faced entries correspond to the better model under the respective number of topics.

The table reveals interesting information about the models. LeadLag LDA is able to outperform LDA at higher thresholds 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 higher thresholds are reached, where the quality of neighbors is much better.

4 Results and Discussion

4.1 CS Journals vs. Conferences

In this subsection, we present topic specific lead/lag results between journals and conferences using the same dataset we used in our validation experiments in section 2.1. Although we ran topic models with various number of topics ranging from 50 through 200, in practice, we found that 50 topics works best for this corpus. The reason is that with increasing number of topics, documents that pass the condition $N_{d\theta}k_{di} > 4$ in Eq. 9 become scarcer, and estimates of topic-specific lag begin to exhibit higher variance, and are therefore less reliable. Figure 7 displays the lag values for journal articles for several randomly selected topics. The plot shows that even when resolved by topics, journals tend to lag behind conferences by around a year.

4.2 CS Grants vs. Science

We also ran a 50 topic LeadLag LDA on the twin outlets of grants and science. The grants outlet is represented by successful NSF grant proposals9, while ‘science’ is approximated by all publications from the ISI dataset10. 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 8 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 most modern 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 traditional topics such as ‘Databases’ and ‘Algorithms’.

4.3 News vs. Blogs

For this run, we used a subset of the Spinn3r index11 (Leskovec, Backstrom, and Kleinberg 2009) 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 News12. 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 9 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

Although there has been very little work that explicitly models lead/lag of knowledge outlets in disseminating novel in-

---

9 http://www.nsf.gov
10 http://www.isiknowledge.com
12 http://news.google.com
formation, there has been some interesting work in tracking trends and influences across outlets. For example, (Leskovec, Backstrom, and Kleinberg 2009) track short distinctive phrases called ‘memes’ across various outlets such as blogs, news media and social media and plots interesting visualizations of the rise and fall of popularity of the memes. In a related paper, (Yang and Leskovec 2010) model the influence of outlets based on the rate of information diffusion through the network. Using the model, they plot the influence of each outlet such as news, television and blogs in disseminating specific memes as a function of time. The memes are in turn classified into various topics such as Politics, Entertainment, Business, Sports and Technology, which allows the user to see the influence of outlets on each topic separately.

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.

The main novelty of our work lies in two aspects:
1. Network independent modeling: The models we presented can estimate the lead/lag of outlets completely independent of the network information, and purely based on text and time-stamp information. Network information, although useful, is often missing from many datasets. For example in the grants vs. science data, grant proposals have no incoming links because they are not evidence of scientific accomplishments. In other datasets such as blogs, and twitter feeds, network information may often be missing or noisy. We believe there are several insights one could draw from tracking word usage alone, and our work is a step in this direction.
2. Automatic topic discovery: LeadLag LDA automatically estimates topic-specific leads and lags but does not need any labeled data for topics. This is in contrast to the approaches mentioned above that either perform topic independent analysis or rely on manually tagged data to identify the 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.

6 Conclusion

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.

6.1 Future Work

In this work, we presented methods to estimate lead/lag of only a pair of outlets with respect to each other. In the future, we would like to extend this approach to settings where there are more than two outlets. The LeadLag LDA model we proposed uses the neighbors computed by the TF-IDF model as input. This design was primarily chosen for reasons of computational efficiency and scalability. In the future, we would like to relax this assumption, and factor-in neighbor detection as part of the model.

We also made the simplistic assumption that the lead/lag of one outlet with respect to the other is static. In practice, it is likely that the lead/lag behavior is time-dependent. We would like to extend our approach to capture such dynamic lead/lag patterns.

References


Figure 7: Number of years by which ACM journals lag behind conferences resolved by topics. Topics are described in terms of the top 10 most likely terms. On all topics journals uniformly lag behind conferences.

Figure 8: 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 traditional topics like ‘Databases’ and ‘Algorithms’ (last two bars in the histogram) but lags behind on more modern topics of Computer Science such as ‘Speech Recognition’ (second bar from top), ‘Computed Aided Health Care’ (fourth bar from top), ‘Mobile Networks’ (sixth bar from top), etc.

Figure 9: 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).