SANDIA REPORT

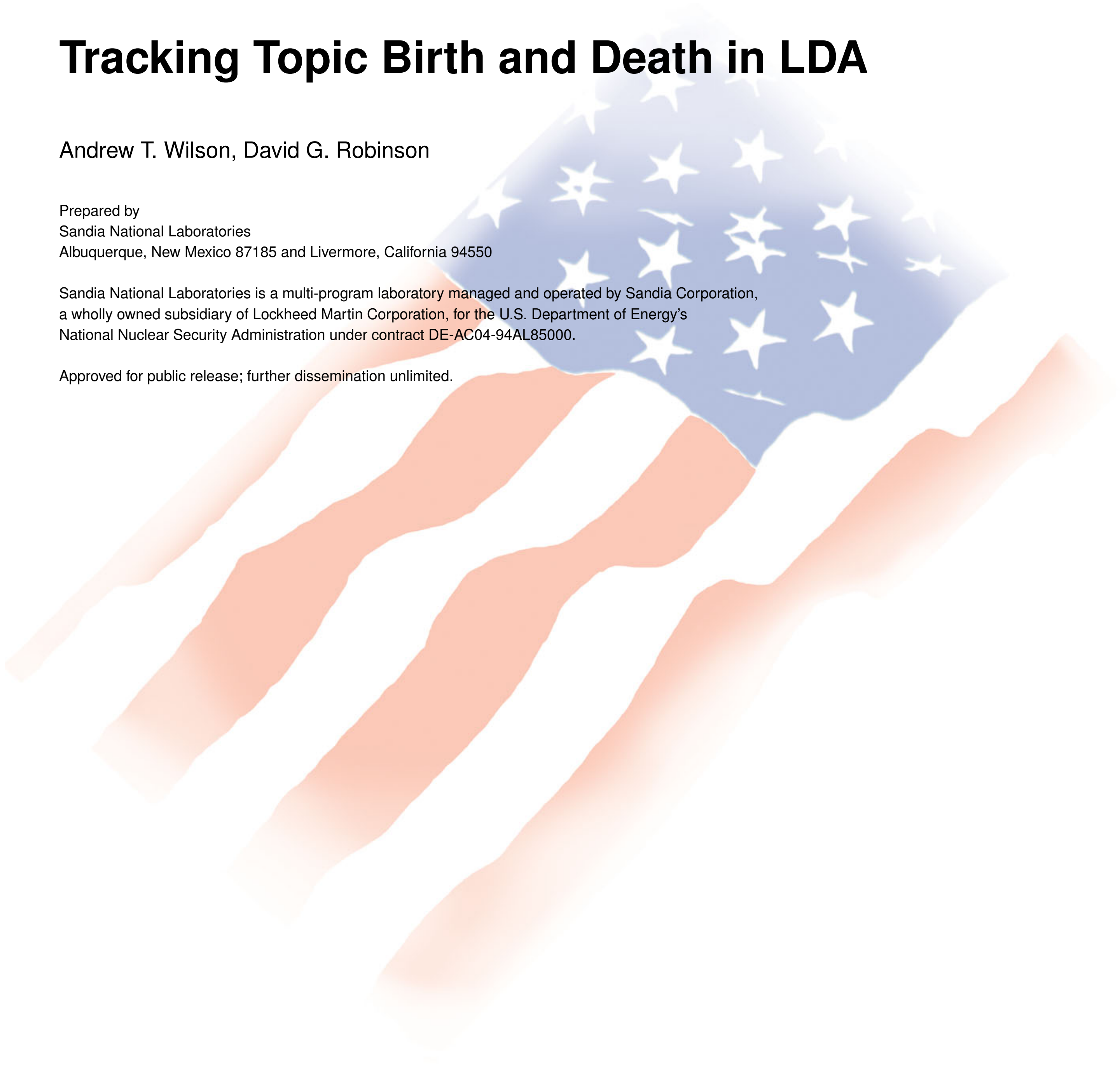
SAND2011-6927
Unlimited Release
Printed September 2011

Tracking Topic Birth and Death in LDA

Andrew T. Wilson, David G. Robinson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation,
a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's
National Nuclear Security Administration under contract DE-AC04-94AL85000.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2011-6927
Unlimited Release
Printed September 2011

Tracking Topic Birth and Death in LDA

Andrew T. Wilson
Sandia National Laboratories
P.O. Box 5800
M/S 1323
Albuquerque, NM 87185-1323
atwilso@sandia.gov

David G. Robinson
Sandia National Laboratories
P.O. Box 5800
M/S 1323
Albuquerque, NM 87185-1323
drobin@sandia.gov

Abstract

Most topic modeling algorithms that address the evolution of documents over time use the same number of topics at all times. This obscures the common occurrence in the data where new subjects arise and old ones diminish or disappear entirely. We propose an algorithm to model the birth and death of topics within an LDA-like framework. The user selects an initial number of topics, after which new topics are created and retired without further supervision. Our approach also accommodates many of the acceleration and parallelization schemes developed in recent years for standard LDA.

Contents

List of Figures

Chapter 1

Introduction

In recent years, topic modeling algorithms such as latent semantic analysis (LSA)[17], latent Dirichlet allocation (LDA)[10] and their descendants have offered a powerful way to explore and interrogate corpora far too large for any human to grasp without assistance. Using such algorithms we are able to search for similar documents, model and track the volume of topics over time, search for correlated topics or model them with a hierarchy.

Most of these algorithms are intended for use with static corpora where the number of documents and the size of the vocabulary are known in advance. Moreover, almost all current topic modeling algorithms fix the number of topics as one of the input parameters and keep it fixed across the entire corpus. While this is appropriate for static corpora, it becomes a serious handicap when analyzing time-varying data sets where topics come and go as a matter of course. This is doubly true for online algorithms that may not have the option of revising earlier results in light of new data. To be sure, these algorithms will account for changing data one way or another, but without the ability to adapt to structural changes such as entirely new topics they may do so in counterintuitive ways. See Figure 1.2 for an example where a major change in content (discussion of Hurricane Katrina) appears in a wholly non-obvious place in the data (a topic about communication).

1.1 Motivation

Understanding the changes in a data set over time is essential in narrative formation and is a natural, ubiquitous part of any sort of analysis. For an everyday example, see Figure 1.1, an excerpt from a chart that traces the lineage of pop and rock-and-roll music starting in the 1950s. This chart was constructed manually by an expert in the field and surveys hundreds of artists (whose works, in some sense, can be considered "documents") to illustrate the birth and death[1] of genres ("topics") within popular music. However, this period in popular music comprises a very small amount of data compared to other domains of current and ongoing interest. The following two examples involve far more data than any expert could hope to grasp in a lifetime of study – and that data is constantly growing. Moreover, large amounts of money, security, and even lives can depend upon the ability to quickly grasp and respond to changes in their structure.

[1] In some cases the disappearance of a type of music may be a greater cultural contribution than its birth.

Portfolio Analysis: Given a set of technical papers, patents and press releases covering a field, what are the major topics of discussion and what entities are discussing them? When does discussion of a particular subject cease? Where should research funds or venture capital be invested for greatest return?

Cybersecurity: Traffic on a network can be tokenized in many ways, including some as simple as grouping on source and target address and TCP port. This makes it accessible to topic modeling algorithms. One of the most difficult and enduring tasks in cybersecurity is characterizing "normal" traffic for use as a baseline for detecting abnormal and potentially interesting events. An algorithm that detects new "topics" in such traffic can present them to an analyst for investigation and either highlight them in future or incorporate them into the evolving baseline.

We aim to close the gap between what current algorithms can do (Figure 1.2) and what we would like to do (Figure 1.1). We present an algorithm to enable the discovery of new topics in time-varying document sets. We use regular LDA with collapsed Gibbs sampling as a basis in order to preserve the applicability of many of the optimizations and extensions developed in recent years.

1.2 Algorithm Sketch

The intuition behind our approach is as follows. As described by Blei et al. [10], the latent Dirichlet allocation algorithm models a document as a mixture of *topics*[2]. Each topic is a probability distribution over some vocabulary V. LDA uses Bayesian inference to learn both the mixture components ϕ and the mixing proportions θ that best represent a set of input documents D. The mixture components, which are our topics, can then be used to infer mixing proportions for a new set of documents D'.

If we allow the topics ϕ to change as we learn mixing proportions for D', we obtain a new matrix ϕ' whose contents are similar to the original ϕ. Intuitively, the topics change slightly to better approximate the combination of D and D'. We call these changes *topic drift*. Our hypothesis is tha while a small amount of drift is normal, a large drift indicates the emergence of a new topic. Since the distance between two probability distributions is a well-studied concept, we can measure this drift and operate on its values.

In the next section we provide a brief survey of related work before moving on to the details of our algorithm and its implementation. We present test results on a real-world data set composed of articles posted on BoingBoing (http://boingboing.net), a blog concentrating on themes of intellectual property, popular culture and science fiction. We conclude with a discussion of our algorithm's advantages and drawbacks and a few avenues for future work.

[2] A topic in the LDA sense is not the same as a topic in the linguistic sense. However, for the purposes of this paper we use the two interchangeably, as LDA topics are often comprehensible as linguistic topics.

Figure 1.1. An excerpt from the chart entitled “Genealogy of Pop/Rock Music” by Reebee Garofalo, cited in Edward Tufte’s “Visual Explanations” [51]. This is a hand-drawn visualization of the emergence, development and disappearance of sub-genres of music across nearly 30 years of history. Our ultimate goal is to generate charts like this automatically.

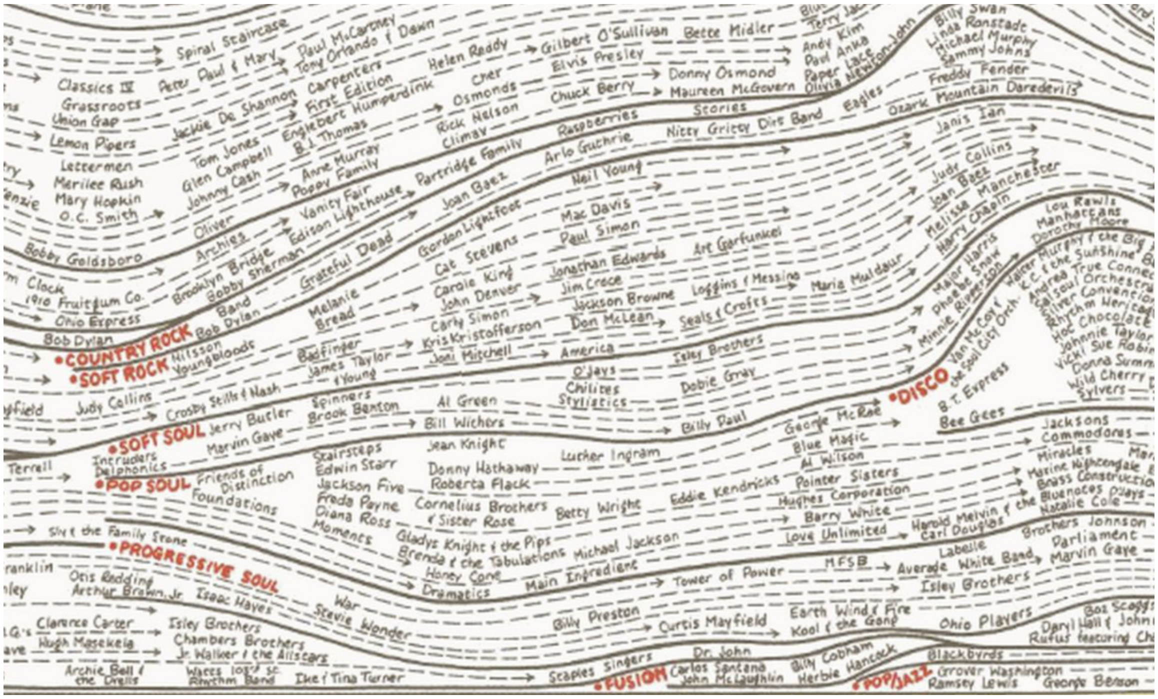
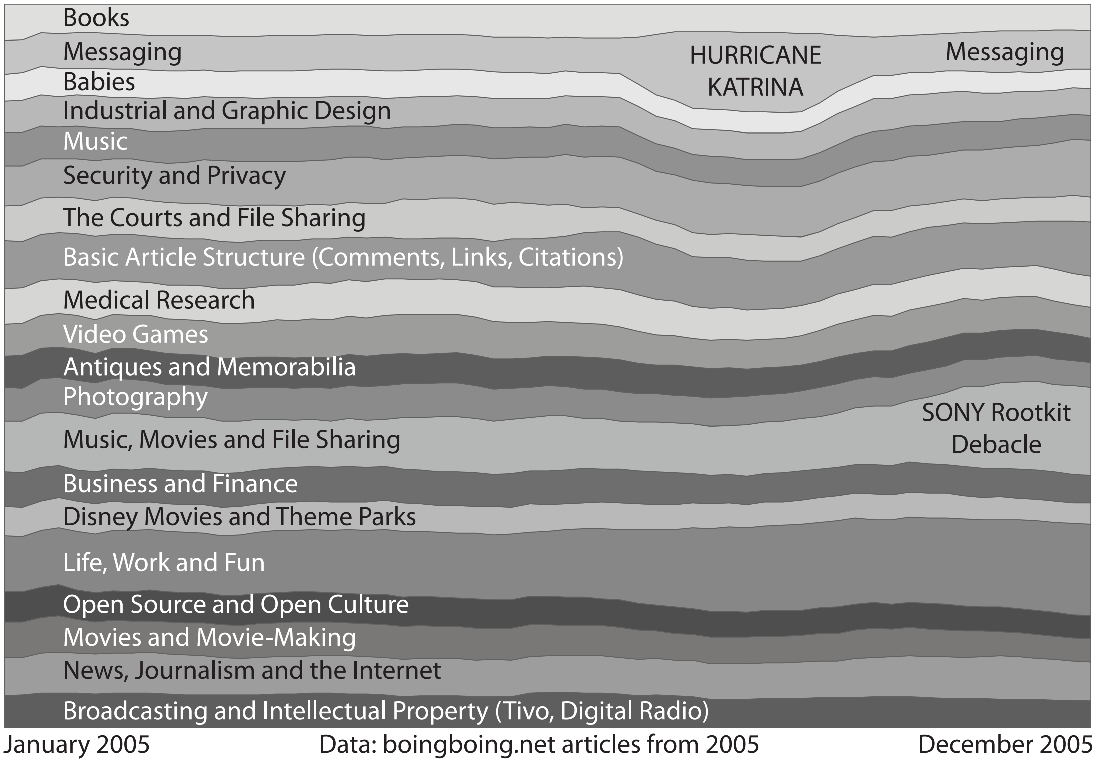


Figure 1.2. Latent Dirichlet Allocation (LDA) applied to 7,025 articles from the Boing Boing web site (`http://boingboing.net`) spanning all of 2005. We fixed the number of topics at 20 to produce the chart here. Since LDA cannot change the number of topics during execution, major developments such as Hurricane Katrina are subsumed in existing topics and are difficult to distinguish.



Chapter 2

Related Work

There are two broad classes of dynamic topic models: those that rely on discretization of time, e.g. an underlying Markov model, and those that are based on time as a continuous variable within the analysis. Discrete time models inherently assume that topics arise at specific, predetermined points in time. Effectively, this is similar to evaluating a sequence of static LDA models on each interval in isolation and assessing the the change in the topic distribution from one interval to the next. Computationally this is cumbersome and can lead to missing the emergence of topics.

For example, in Topics Over Time[53] the topics are held constant and the time information within the model is treated as a variable and used to discover these hidden topics. A change in words coupled with a discrete change in time is used to detect a change in topic patterns. Conversely, Timeline[2] uses a hierarchical Dirichlet process[50] to learn topics within within each group of documents and introduces temporal dependence to connect topics from different intervals. The mathematics involved in introducing this dependence do not lend themselves to a simple implementation.

Other methods [e.g. 3] rely on the KL divergence[33] to detect a change in topic distribution. The intent of the KL metric is to provide a measure of the information gained or lost as an indication of the emergence or decay of topics. The lack of symmetry of the KL metric presents the possibility of mis-characterizing the change in topic structure. There are a number of alternative valid distance metrics that can be used to detect the topic dynamics; our choice here is the symmetric Jensen-Shannon information metric[36]. The symmetric nature of the metric allows us to use a z-test[54] to statistically identify a change in topics as a function of time.

Another approach to allowing a model to change over time arises from sequential importance sampling, also known as particle filtering. Doucet et al. [19] provide an overview. The intuition behind particle filtering is that instead of spending all available computation on tracking the single best estimate of a set of parameters, one can, in a sense, try all possibilities and let the statistics sort it out. Canini et al. [14] describe an implementation of online LDA in the framework of particle filters that yields results with higher likelihood than the common approach of running regular LDA run multiple times and keeping the best result.

Chapter 3

Algorithm

In this section we describe the details of our algorithm. Our goal is to track the divergence of topics between epochs to identify newly-emerged topics and the change in volume and content to identify topics that have run their course and can be terminated.

3.1 Data Organization and Parameters

We begin with the assumption that the documents are ordered in ascending order according to their timestamps. The user chooses the size of one *epoch*, the unit of time within which topics will remain constant, at whatever granularity is desired. For a corpus where several documents arrive each day, a week-long epoch may be most appropriate. For a corpus spanning decades, a month- or year-long epoch may be best. This choice is informed more by the user's needs than by any algorithmic requirement.

The user must select values for the following parameters. We discuss suggested rule-of-thumb values in Section 4.3.

- k_0, the initial number of topics
- α and β, the standard LDA hyperparameters ($\frac{50}{k_0}$ and 0.1 are common choices)
- z_{split}, the required "outlierness" to declare a new topic
- t_{immune}, the number of epochs that must pass after a split before a recently-split or recently-created topic is eligible for bifurcation
- v_{min}, the minimum volume below which a topic is eligible for termination
- t_{ending}, the number of epochs of "probation" before a topic with volume below v_{min} will be terminated

Once we have chosen values for all the parameters, we set aside the first several epochs' worth of documents as a training set. In our experiments we used 10. There is no specific required value for this number except that it should contain several times as many tokens as the densest epoch. We call this training set e_0.

We begin by fitting a standard LDA model to the documents in e_0 with the user's values for k_0, α and β. Using collapsed Gibbs sampling, we learn z_0, an assignment of topics to tokens. We run the Gibbs sampler to allow it to burn in and then use a single sample from the posterior distribution to estimate ϕ_0 and θ_0.

3.2 Tracking Topic Drift

After processing the training data in epoch 0 we proceed as indicated in Algorithm 1. To compute the drift of any topic ϕ_t^k with respect to its counterpart ϕ_{t-1}^k we note that each topic is a probability distribution. This allows us to choose any convenient divergence measure such as Hellinger distance, Hellinger coefficient or Jensen-Shannon divergence. Jensen-Shannon divergence is attractive because it handles naturally the case where the two distributions being compared have different support.

After computing the drift d_t^k for all topics in epoch t we can determine whether any of them have changed enough to indicate that it has changed enough to constitute a new topic.

3.3 Identifying New Topics

Many factors, from the choice of hyperparameters and k_0 to the distribution of types in the data itself, can influence the exact values of the drift measures from epoch to epoch. For this reason we do not use the drift values directly but instead consider their properties as an ensemble and look for outliers.

Since statistics such as the mean and variance are themselves susceptible to the influence of outliers, we use the *modified Z score*[21] to identify the central tendency. The modified Z score is similar in spirit to the standard Z score but uses the median and median absolute deviation (MAD) instead of the mean and standard deviation. The median and MAD are far more resistant to the influence of outliers. [1]

Once we have the Z score for each topic the parameter z_{split} (one of the supplied inputs to the algorithm) allows us to identify which topics have drifted too far and need to be split. Algorithm 1 describes this process. We note that the reassigned tokens are precisely those that caused the excessive drift and thus form a natural foundation for the new topic. Finally, once a new topic

[1] The median and MAD are based on a loose assumption that the data being characterized are normally distributed. We observe that the distribution of drift values is unimodal but not necessarily normal according to an Anderson-Darling test. Neither are they conclusively *not* normal, although they are often skewed toward higher values. We might achieve better results by fitting a gamma distribution to the data and using that to estimate the likelihood of each drift value.

Algorithm 1 Overall algorithm for LDA with topic birth/death. The notation LDA($D,\mathbf{z}$) indicates the use of LDA to learn topic assignments for all tokens in the supplied documents D using topics from the already-learned model $\mathbf{z}$. Details of the method for detecting and closing low-volume topics are omitted here for clarity but are discussed further in Section 3.4.

```
$K_0 \leftarrow K_{initial}$
$E_0 \leftarrow$ training documents
$E_1, \ldots, E_N \leftarrow$ documents divided into epochs
$(\mathbf{z}, \phi_0) \leftarrow$ LDA($E_0$, $\emptyset$) {train initial model}
**for** $i = 1$ **to** $N$ **do**
    Mark low-volume topics as closed
    $(\mathbf{z}, \phi_i) \leftarrow$ LDA($E_i$, $z$) {learn assignments for latest epoch}
    **for** $k = 1$ **to** $K_{i-1}$ **do**
        $d_i \leftarrow$JS$(\phi^k_{i-1}, \phi^k_i)$ {compute topic drift since last epoch}
    **end for**
    $(m, s) \leftarrow$ median and MAD of $d_1 \ldots d_K$
    **for** $k = 1$ **to** $K_{i-1}$ **do**
        $a_k \leftarrow \frac{d_k - m}{s}$
        **if** $a_k > z_{split}$ **then**
            $k_{new} \leftarrow K_i + 1$ {start a new topic}
            $K_i \leftarrow K_i + 1$
            Reassign tokens in $\mathbf{z}$ from $k$ to $k_{new}$
        **end if**
    **end for**
    Remove tokens for oldest $|E_{i-1}|$ documents from $z_i$
**end for**
```

has been created, both it and its parent are immune from further splits and termination for t_{immune} epochs.

3.4 Closing Old Topics

Just as new topics of discussion arise over time in bodies of text, old ones will often fade away and be subsumed into some larger discussion or else dropped entirely. We approximate this effect by measuring the number of tokens assigned to each topic k as a fraction of all documents in the current window. If the total number of tokens in the window after epoch e is n_e, a topic must hold at least $\frac{n_e \cdot v_{min}}{k_e}$ tokens to remain a going concern. Topics with fewer tokens are placed on *probation*. If a topic stays on probation for more than t_{ending} epochs then it is marked as *closed*. Tokens in new documents cannot be assigned to closed topics.

Chapter 4

Implementation and Results

4.1 Implementation Notes

We implemented our algorithm primarily in Python with a few speed-critical components in a C++ module. Our LDA implementation relies on collapsed Gibbs sampling with a symmetric Dirichlet prior as described by Griffiths and Steyvers [25]. We use NLTK [8] to load documents from disk and remove common stop words using its built-in "english" stop list. We do not otherwise filter the tokens to remove either low- or high-frequency terms from the data.

In many LDA implementations using Gibbs sampling, the ϕ and θ count arrays account for the majority of the runtime memory requirements. The ϕ arrays grow particularly large and sparse when the number of terms used in the corpus grows without bound, as is frequently the case with technical literature. We take advantage of this sparsity by implementing ϕ and θ as hash tables instead of flat arrays. Moreover, we move these hash tables along with the topic sampler itself into a C++ module callable from Python for speed. Since these are the most frequently called parts of the code we realize more than 10x acceleration in the system as a whole.

4.2 Handling Changing Vocabulary

The size of the vocabulary and the number of topics are both involved in the update equation for topic assignments in the Gibbs sampler. Since both of these parameters change naturally as new documents arrive and old ones leave, we must account for their changing values. In our system we recalculate α, K and W (see Algorithm 1) at the beginning of each epoch according to the terms and topics currently in use. This suffices because the collapsed Gibbs sampler does not explicitly store estimates of θ and ϕ.

4.3 Parameter Values

Our hope was that it would be possible to start the algorithm off with a parameter count reflecting a certain desired level of abstraction for the results. For example, an initial topic count of 20 for a corpus of thousands of documents should produce very high-level topics whereas a topic count of 100-200 should produce far more detailed results. To test this, we ran with the following parameter settings, all determined empirically.

- $K_0 = 20$ (fairly abstract topics)
- $\alpha = \frac{50}{K}$ (standard value in the LDA literature)
- $\beta = 0.1$ (another standard value in the LDA literature)
- $z_{split} = 5$
- $t_{immune} = 5$ (determined by inspecting new topics to see when their volumes stabilize)
- $v_{min} = 0.25$ (see Section 5.2.2 for discussion)
- $t_{ending} = 5$

4.4 Test Data and Environment

We tested our algorithm using a set of 63,999 articles from Boing Boing (`http://boingboing.net`), a blog covering issues in popular culture with a loose focus on technology, intellectual property and politics. In late January 2011 they made all the articles published on their web site available for download as XML [42]. We retrieved these articles and stripped out the HTML formatting, leaving just plain text. The entire data set contains 9.48 million tokens.

We ran our tests using one core of a dual-processor Mac Pro desktop with 32GB of memory. Even when running the entire data set at once with full history data in memory (including all documents) the Python process never required more than 3.2GB of memory. We anticipate that this will go down still further with an all-C++ implementation.

4.5 Results: Finding Katrina

For clarity of illustration, we present here the results of a run over the subset of the data spanning calendar year 2005. This year is a good test case because of one particularly notable event. Hurricane Katrina struck the city of New Orleans in late August, causing widespread devastation and sparking a number of crises whose effects and resolution were topics in the media for many months afterward. We used this event as a benchmark to judge whether our algorithm was doing anything at all reasonable.

For ease of comparison, we reproduce as Figure 4.5 the 20-topic chart of 2005 first seen in the introduction. It is followed by our results on the same data in Figure 4.5. Topics that were automatically created by our algorithm are highlighted in green and blue.

Figure 4.1. Latent Dirichlet Allocation (LDA) applied to 7,025 articles from the BoingBoing web site (`http://boingboing.net`) spanning all of 2005. We fixed the number of topics at 20 to produce the chart here. Since LDA cannot change the number of topics during execution, major developments such as Hurricane Katrina are subsumed in existing topics and are difficult to distinguish. The labels on each topic were created manually by inspecting the most highly-weighted terms in the LDA results.

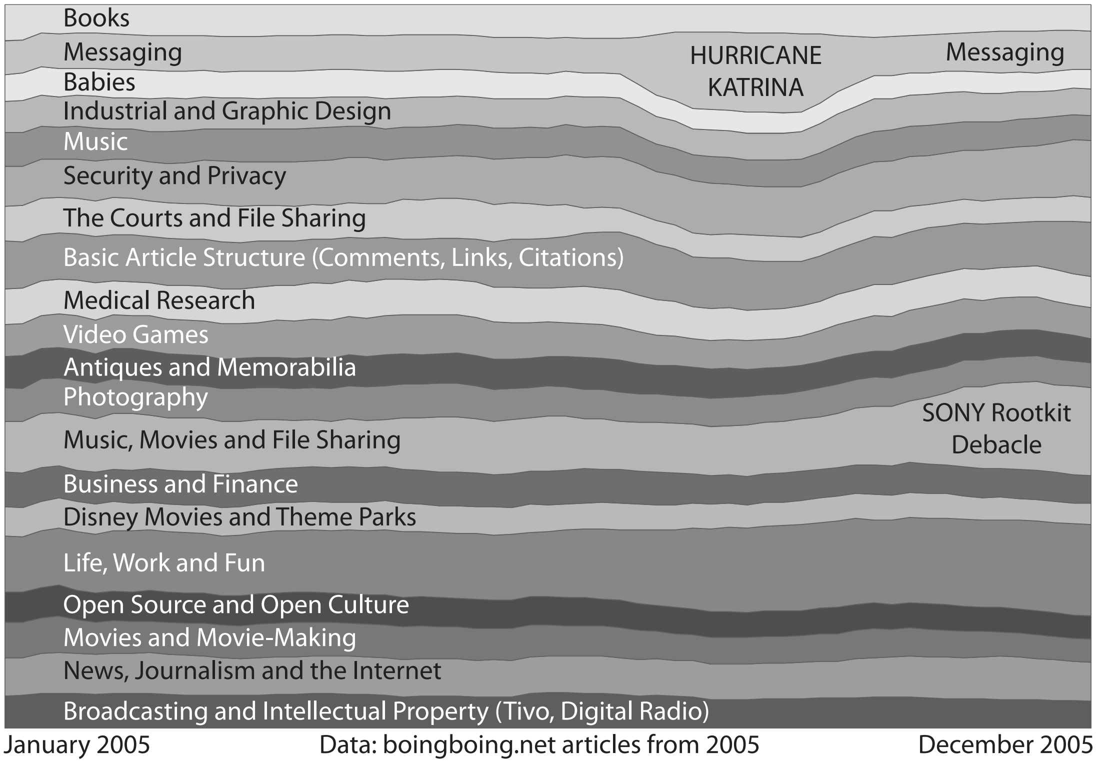
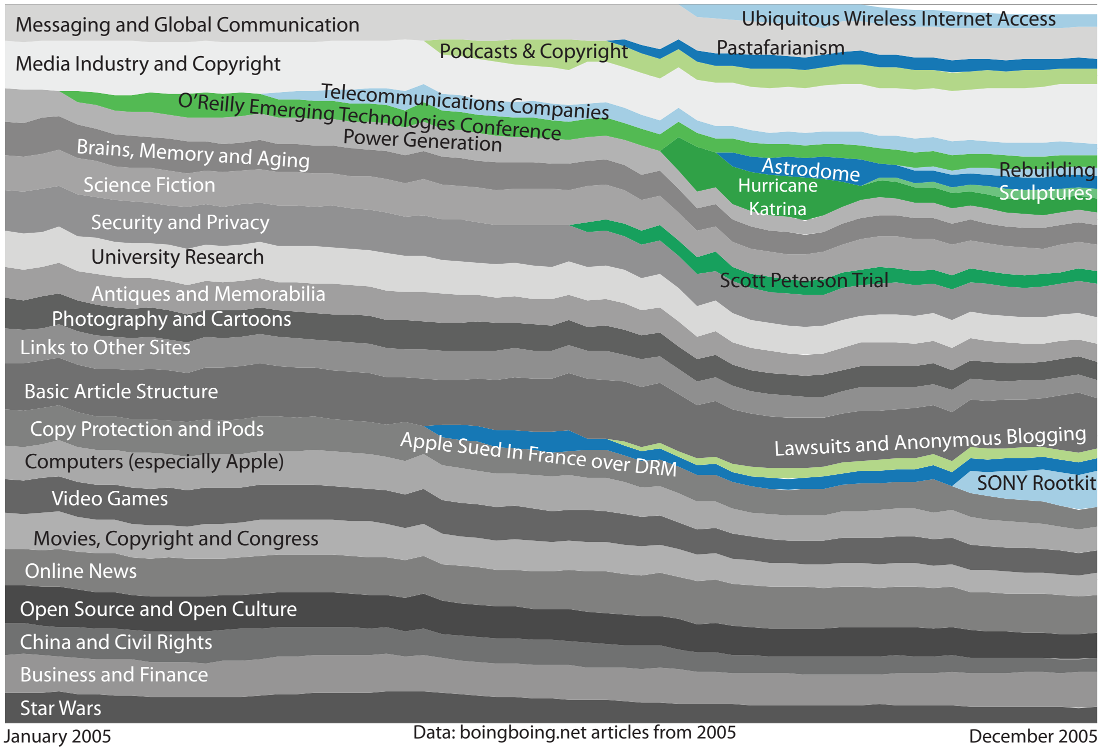


Figure 4.2. LDA with topic birth/death applied to the same 7,025 articles from BoingBoing. The original 20 topics are drawn in grayscale. Topics created by our algorithm are shown in blue and green and appear directly above the topic from which they separated. For example, the topic "Apple Sued In France over DRM" emerged from the "Copy Protection and iPods" topic. Note that not only did our algorithm separate Hurricane Katrina as a distinct topic, it identified further splits such as conditions in the Astrodome evacuation shelter and the beginnings of the process of rebuilding.



Chapter 5

Discussion

We developed our algorithm as an extension to LDA with the intent that existing acceleration schemes such as Fast LDA [41], Sparse LDA [57] and Approximate Distributed LDA [40] should still be applicable after our modifications. We achieved this by leaving the collapsed Gibbs sampler unchanged and instead modifying the number of topics and the token labels between iterations. We can do this with exactly the same operations that the sampler itself uses to label tokens with topics and track the counts of how many tokens in each document belong to each topic.

5.1 Planning for Parallelism

Our approach lends itself to a parallel implementation. The most data-intensive part is the computation of topic divergence between the current and previous epoch. In this stage we compute k Jensen-Shannon divergences over distributions of length V. Once all the drift values are available, the simplest (and fastest) approach is to compute and broadcast the modified z-scores, then allow each node to independently modify topic labels for its share of the latest epoch. No complex communication is required.

5.2 Shortcomings

We find two principal shortcomings in the algorithm as presented in this report. Both relate to the behavior of created topics over time as it relates to already-existing topics. As a result, our algorithm performs very well over smaller numbers of epochs (50-100) but begins to diverge beyond that.

5.2.1 Rich Topics Get Richer

We observe a "rich get richer" phenomenon among the top few topics when we run our algorithm for more than about 100 epochs. That is, the heaviest topics from the training set and the first few epochs will be used more and more frequently to label tokens as time goes on. At the same time, newly created topics rarely grow beyond a certain point. For example, when we ran the entire BoingBoing corpus with 50 initial topics, we reached a point (midway through 2007) where the algorithm held a total of 360 topics. Fully 25% of the tokens were labeled with the most heavily-weighted topic. This topic was used to label over 100 times as many tokens as the topic with median weight. Moreover, the top 10 topics accounted for just over half of the token labels. Under such circumstances we believe that the most frequently appearing topics will continue to grow.

We speculate on the causes of this problem. As we noted in Section 4.5, newly created topics tend to be very specific (e.g. Hurricane Katrina, the Astrodome, or SONY's distribution of a particular copy protection method) while earlier topics tend to represent broader categories such as video games, intellectual property and science fiction. It may be that general categories are "close enough" to attract tokens that might be better suited to a more specific topic.

One possible solution to this problem is to use more tokens to seed a new topic. Instead of using only the tokens from the latest epoch, we could reach back into the document window and reassign other tokens from earlier documents. The parent topic is an obvious source for these tokens. This is unlikely to be sufficient since the new topic will still be limited by the contents of its parent. Up to a certain point, we are aided by the fact that the Gibbs sampler uses proportions (i.e. $\mathbf{p}(k|d) \propto \mathbf{p}(t|k)\mathbf{p}(k|d)$ for a topic k, term t and document d) without reference to the overall number of tokens belonging to each topic, but we do not believe this will be enough. We discuss other possible solutions in Section 6.2

5.2.2 Topics Never Die

To our surprise, we have not observed a single instance of a topic being terminated when running on real data. Instead, topics will settle down to some minimal-but-stable volume and stay there indefinitely. We conjecture that this may be due to the influence of the Dirichlet prior as a smoothing factor in the selection of topics for tokens. Since an open topic will always have some nonzero probability of being chosen for any token, it stands to reason that in each epoch there will be some nonzero number of tokens with any given label. It may also be the case that these "ghost" topics are overfitting some aspect of the data. Given their generally small size, it is also possible that these ghost topics accumulate random drift over time that keeps them too large to qualify for closure.

Chapter 6

Conclusions

6.1 Alternative Distance Metrics

While the algorithm performs smoothly with the Jensen-Shannon distance metric, it is not clear that JS will generalize to other data sets. A number of alternative measures have been used successfully on similar efforts. These include:

Pearson Correlation Coefficient (Pearson, 1896)

$$\sigma_p = \frac{\sum_i^C (f_{U_i} - \bar{f}_U)(f_{V_i} - \bar{f}_V)}{\sqrt{\sum_i^C (f_{U_i} - \bar{f}_U)^2}\sqrt{\sum_i^C (f_{V_i} - \bar{f}_V)^2}}$$

where $\bar{f}_U = 1/n \sum_i^C f_{U_i} = 1/C$

Hellinger Coefficient (Hellinger, 1907)

$$\sigma_H = \sum_i^C \sqrt{f_{U_i} f_{V_i}}$$

Proportional Similarity (Renkonen, 1938)

$$\sigma_{ps}(U,V) = 1 - \frac{\sum_i^C |f_{U_i} - f_{V_i}|}{2} = \sum_i^C \min(f_{U_i}, f_{V_i})$$

Of these, the Hellinger Coefficient and the Proportional Similarity have the most promise for increased topic resolution while being robust to data stream characteristics.

6.2 Topic Decay

As mentioned in Section 5.2.2, in our tests the topics do not noticeably decay. This likely results from a number of factors including, among other items, too strong a prior for topic generation (large values of α) or possibly from long term instability of the Gibbs sampler. Reducing α is not a preferred solution, since this will negatively impact the emergence of topics as well. One solution might be to periodically, *reset* the analysis at some logical point in the data stream. While this raises the question of correspondence between topics before and after this reset, it would allow the sampler to learn a new, more balanced distribution going forward. A second, significantly more computationally involved solution, is the relaxation of the number of topics at each epoch; allow the number of topics to increase or decrease with each generation. This approach is well suited for a particle filter implementation similar to the one described by Canini et al. [14].

6.3 Clustering in Topic Space

One possibility that has been raised in the literature as well as in discussion is to identify thematically related groups of documents by performing an additional clustering step in the space of mixing weights (probability distributions) resulting from LDA. Here each document would be regarded as a point on a k-simplex whose coordinates are defined by that document's mixing weights. It is possible that this would sidestep the topic decay problem. We could either use an algorithm such as k-means that takes the number of clusters as an input parameter or else opt for a clustering method such as the one described by Schubert and Sidenbladh [48] that learns the "best" number of clusters during execution.

6.4 Final Thoughts

We have demonstrated that topic drift in LDA is an indicator of the emergence of a new subject within a corpus of non-technical news articles. In tests, our method successfully identified major topic shifts within our test corpus. While we have not yet tested our approach on a wider array of data types, we believe that this same method will generalize well. Our algorithm is structured to maintain compatibility with many of the acceleration schemes and enhancements made to LDA in recent years. Although topic decay remains a significant problem when processing larger data sets, we have approaches in mind to ameliorate or avoid this phenomenon.

References

[1] *Modeling Science*, April 2008.

[2] Amr Ahmed and E P Xing. *Timeline: A Dynamic Hierarchical Dirichlet Process Model for Recovering Birth/Death and Evolution of Topics in Text Stream*. Citeseer, 2010.

[3] L AlSumait, D Barbará, and C Domeniconi. On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 3–12, 2008.

[4] F Bacchus and YW Teh. Making forward chaining relevant. *Proceedings of the Fourth International Conference on AI Planning Systems*, pages 54–61, 1998.

[5] A Banerjee and S Basu. Topic models over text streams: A study of batch and online unsupervised learning. *SIAM Data Mining*, 2007.

[6] N Bartlett, D Pfau, and F Wood. Forgetting Counts: Constant Memory Inference for a Dependent Hierarchical Pitman-Yor Process. *to appear) ICML*, 2010.

[7] M Bautin, L Vijayarenu, and S Skiena. International sentiment analysis for news and blogs. *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2008.

[8] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing, 2009.

[9] David M Blei and John D Lafferty. A Correlated Topic model of Science: Supplement. *Annals of Applied Statistics*, 1(1):17–35, 2007.

[10] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research 3*, pages 993–1022, 2003.

[11] DM Blei and JD Lafferty. Dynamic topic models. *Proceedings of the 23rd international conference on Machine learning*, page 120, 2006.

[12] II Bogen. Application of kalman filters to identify unexpected change in blogs. *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 305–312, 2008.

[13] L Bolelli, Ş Ertekin, and CL Giles. Topic and Trend Detection in Text Collections Using Latent Dirichlet Allocation. *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 776–780, 2009.

[14] KR Canini, L Shi, and TL Griffiths. Online inference of topics with latent Dirichlet allocation. *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 5, 2009.

[15] B Carpenter. Integrating Out Multinomial Parameters in Latent Dirichlet Allocation and Naive Bayes for Collapsed Gibbs Sampling. pages 1–10, September 2010.

[16] Sourav Chatterji and Lior Pachter. Multiple organism gene finding by collapsed Gibbs sampling. In *RECOMB '04: Proceedings of the eighth annual international conference on Research in computational molecular biology*, pages 187–193, New York, NY, USA, 2004. ACM.

[17] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[18] Yi-qun Ding, Shan-ping Li, Zhen Zhang, and Bin Shen. Hierarchical topic modeling with nested hierarchical Dirichlet process. *Journal of Zhejiang University - Science A*, 10(6):858–867, June 2009.

[19] A Doucet, N de Freitas, and N Gordon. An Introduction to Sequential Monte Carlo Methods. pages 1–12, January 2011.

[20] A e Gohr, A Hinneburg, R e Schult, and M Spiliopoulou. Topic evolution in a stream of documents. *SIAM*.

[21] Agata Fallon and Christine Spada, 1997. [Online; accessed 22-Sep-2011].

[22] W Fu and EP Xing. Dynamic mixed membership blockmodel for evolving networks. *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[23] D Görür and Y W Teh. An efficient sequential Monte Carlo algorithm for coalescent clustering. *NIPS 2008*, 2008.

[24] B Graf. Semi-Supervised Learning of Bayesian Language Models with Pitman-Yor Priors.

[25] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences USA*, volume 101, pages 5228–5235, 2004.

[26] Dan He and D Parker. Topic dynamics: an alternative model of bursts in streams of topics. *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, July 2010.

[27] Q He, B Chen, J Pei, B Qiu, and P Mitra. Detecting topic evolution in scientific literature: how can citations help? *Proceeding of the 18th . . .* , 2009.

[28] M Hoffman, D Blei, and F Bach. Online Learning for Latent Dirichlet Allocation. *NIPS 2010*, 2010.

[29] B Hsu and J Glass. Style & topic language model adaptation using HMM-LDA. *Proc. of EMNLP*, 2006.

[30] MI Jordan, Z Ghahramani, TS Jaakkola, and LK Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[31] Ata Kabán and Mark Girolami. A Dynamic Probabilistic Model to Visualise Topic Evolution in Text Streams. *Journal of Intelligent Information Systems*, 18(2):107–125, March 2002.

[32] M Kolar, AA Le Song, and EP Xing. Estimating Time-Varying Networks. *Arxiv preprint arXiv:0812.5087*, 2008.

[33] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86, 1951.

[34] MK Le Song and EP Xing. Time-Varying Dynamic Bayesian Networks. *Advances in Neural Information Processing Systems*, 22:1732–1740.

[35] CX Lin, B Zhao, Q Mei, and J Han. PET: a statistical model for popular events tracking in social communities. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 929–938, 2010.

[36] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, August 2002.

[37] A McCallum, X Wang, and A Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 2007.

[38] Q Mei, X Shen, and CX Zhai. Automatic labeling of multinomial topic models. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 499, 2007.

[39] Q Mei and CX Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 198–207, 2005.

[40] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed inference for latent dirichlet allocation. In *Advances in Neural Information Processing Systems 20: Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems (NIPS 2007)*, 2008.

[41] Ian Porteous, Arthur Asuncion, David Newman, Padhraic Smyth, Alexander Ihler, and Max Welling. Fast collapsed Gibbs sampling for latent dirichlet allocation. In *In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 569–577, 2008.

[42] Dean Putney. ”Eleven years' worth of Boing Boing posts in one file!”. `http://boingboing.net/2011/01/25/eleven-years-worth-o.html`, 2011. [Accessed on 22-Sep-2011].

[43] L Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989.

[44] Daniel Ramage, Paul Heymann, Christopher D. Manning, and Hector Garcia-Molina. Clustering the tagged web. In *Second ACM International Conference on Web Search and Data Mining (WSDM 2009)*, November 2008.

[45] L Ren, DB Dunson, and L Carin. The dynamic hierarchical Dirichlet process. *Proceedings of the 25th international conference on machine learning*, pages 824–831, 2008.

[46] Lu Ren. *Modeling Temporal and Spatial Data Dependence with Bayesian Nonparametrics*. PhD thesis, April 2010.

[47] Issei Sato and Hiroshi Nakagawa. Topic models with power-law using Pitman-Yor process. In *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Request Permissions, July 2010.

[48] J Schubert and H Sidenbladh. Sequential clustering with particle filters-Estimating the number of clusters from data. *Information Fusion, 2005 8th International Conference on*, 1:8, 2006.

[49] A Strehl, J Ghosh, and R Mooney. Impact of similarity measures on web-page clustering. *AAAI Tech Report WS-00-01. Workshop on Artificial Intelligence for Web*, 2000.

[50] Y W Teh, M I Jordan, M J Beal, and D M Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[51] Edward Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, 1997.

[52] C Wang, D Blei, and D Heckerman. Continuous time dynamic topic models. *The 23rd Conference on Uncertainty in Artificial Intelligence*, 2008.

[53] X Wang and A McCallum. Topics over time: a non-Markov continuous-time model of topical trends. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, 2006.

[54] Wikipedia. ”z-test — Wikipedia, the free encyclopedia”, 2011. [Online; accessed 22-Sep-2011].

[55] D Wilkinson. *Parallel Bayesian Computation*. University of Newcastle, 2005.

[56] B Witten and C Nachenberg. Malware Evolution: A Snapshot of Threats and Countermeasures in 2005. *Malware Detection*, 2007.

[57] Han Xiao and Thomas Stibor. Efficient collapsed Gibbs sampling for latent dirichlet allocation. In *Asian Conference on Machine Learning (ACML)*, 2010.

[58] EP Xing, W Fu, and L Song. A state-space mixed membership blockmodel for dynamic network tomography. *Annals of Applied Statistics*, 4(2):535–566, 2010.

[59] T Xu, Z Zhang, P Yu, and B Long. Dirichlet Process Based Evolutionary Clustering. *Proceedings of the 2008 Eighth IEEE International Conference . . .* , 2008.

[60] T Xu, ZM Zhang, PS Yu, and B Long. Dirichlet process based evolutionary clustering. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 648–657, 2008.

[61] T Xu, ZM Zhang, PS Yu, and B Long. Evolutionary clustering by hierarchical dirichlet process with hidden markov state. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 658–667, 2008.

[62] CX Zhai, A Velivelli, and B Yu. A cross-collection mixture model for comparative text mining. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, page 748, 2004.

DISTRIBUTION:

1	MS 0899	Technical Library, 9536 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911

Sandia National Laboratories