Clustered Latent Dirichlet Allocation for Scientific Discovery

1st Christopher Gropp
Clemson University
Clemson, SC, United States
cgropp@g.clemson.edu

2nd Alexander Herzog
Clemson University
Clemson, SC, United States
aherzog@clemson.edu

3rd Ilya Safro
Clemson University
Clemson, SC, United States
isafro@clemson.edu

4th Paul W. Wilson
Clemson University
Clemson, SC, United States
pww@clemson.edu

5th Amy W. Apon
Clemson University
Clemson, SC, United States
aapon@clemson.edu

***Abstract*—Topic modeling, a method for extracting the underlying themes from a collection of documents, is an increasingly important component of the design of intelligent systems enabling the sense-making of highly dynamic and diverse streams of text data related but not limited to scientific discovery. Traditional methods such as Dynamic Topic Modeling (DTM) do not lend themselves well to direct parallelization because of dependencies from one time step to another. In this paper, we introduce and empirically analyze Clustered Latent Dirichlet Allocation (CLDA), a method for extracting dynamic latent topics from a collection of documents. Our approach is based on data decomposition in which the data is partitioned into segments, followed by topic modeling on the individual segments. The resulting local models are then combined into a global solution using clustering. The decomposition and resulting parallelization leads to very fast runtime even on very large datasets. Our approach furthermore provides insight into how the composition of topics changes over time and can also be applied using other data partitioning strategies over any discrete features of the data, such as geographic features or classes of users. In this paper CLDA is applied successfully to seventeen years of NIPS conference papers (2,484 documents and 3,280,697 words), seventeen years of computer science journal abstracts (533,588 documents and 46,446,184 words), and to forty years of the PubMed corpus (4,025,976 documents and 386,847,695 words). On the PubMed corpus, we demonstrate the versatility of CLDA by segmenting the data by both time and by journal. Our runtime on this corpus demonstrates an ability to function on very large scale datasets.**

***Index Terms*—topic modeling, big data, scientific discovery**

Introduction

Topic modeling, a method for extracting the underlying themes from a collection of documents, is an increasingly important component of the design of intelligent systems enabling the sense-making of highly dynamic and diverse streams of text data [1], [2]. One of the most common models used in practice is Latent Dirichlet Allocation (LDA) [3]. With LDA, documents are assumed to be randomly generated from one or more topics, each of which is a distribution of words. The topics are viewed as latent variables, and LDA executes by inferring the topics from the documents via a Dirichlet process. The algorithm repeatedly samples the documents and modifies the topics to better fit them until reaching a specified convergence. LDA has a number of assumptions, including that both words and documents are unordered and that all documents are generated in the same timeframe.

Of interest in streaming Big Data analytics is the modeling of topics found in a dynamic stream of data, for example, a social media data stream that changes quickly in time [4], a massive collection of publications that have been produced over long time steps [5], or discretized sensor data [6]. Dynamic Topic Modeling (DTM) [5] relaxes the assumption of LDA that all documents are generated in the same time step. The corpus is divided into a sequence of time steps. A fixed count of topics is estimated by the DTM method and the set of topics spans all time steps, but the most important words extracted in each time step for a particular topic are allowed to change through time. The estimation of the topics in a time step is dependent on the estimation from the previous time step. DTM enables observation of how the language of a topic changes over time, and also how well represented a topic is at any given point in time.

The DTM algorithm is, in general, much slower than that of LDA. Because of the dependencies, the application of the DTM algorithm and implementation developed by Blei and Gerrish [7] is limited to modest-sized datasets. DTM, while mathematically elegant, does not lend itself well to direct parallelization because of dependencies from one time step to another, though a few recent attempts have made progress in this area [8].

Our Contribution: We propose an alternative approach to modeling topic dynamics, namely, Clustered Latent Dirichlet Allocation (CLDA). Our approach is based on data decomposition in which the data are partitioned into segments, followed by topic modeling on the individual

This work has been supported by HPCC Systems, LexisNexis Risk Solutions, RELX Group, Elsevier Scopus, Department of Education GAANN award P200A150310, and National Science Foundation awards #1228312 and #1405767.

segments. The resulting local models are then combined into a global solution using clustering. We implement this approach using a fast, parallel algorithm for LDA [9] for inferring the local models, and k-means clustering [10] for combining the results. Our approach has several advantages. The decomposition and resulting parallelization leads to very fast runtime even on very large datasets. The clustering of local topics provides additional insights into topic dynamics; for example, a topic can emerge, die off, or split into multiple local topics, while preserving a global view of topic representation. We compare CLDA to DTM and LDA using metrics of runtime performance, perplexity, and the similarity of topics produced. We report strong results on all of these metrics. The implementation of CLDA is available at [11].

An overview of the steps of CLDA is as follows. First, the data are discretized into segments using time steps or other criteria such as geographic location or data source. Each of these segments is a sub-corpus that is used as the input to a separate run of PLDA [12], a highly parallelized implementation of LDA. Since processing each segment is an independent task, the runs of PLDA on these several segments can also be performed in parallel. The output for this step is a set of topics for every segment. The full list of these topics is passed to a parallelized implementation of k-means clustering [10], producing a set of topics representative of the full set. Because each step of the method uses highly parallelized code, and because the estimation of local topics can be further independently parallelized, CLDA is highly scalable and fast on even very large datasets. As a result of clustering, each original topic is a member of a particular global topic cluster, which is represented by its centroid.

Mathematical analysis of CLDA as compared to DTM is intractable. In this paper we empirically compare CLDA to the original DTM implementation and PLDA with respect to runtime, the quality of topics, and the similarity of the topics that are produced. We apply CLDA successfully to seventeen years of NIPS conference papers (2,484 documents and 3,280,697 words), seventeen years of computer science journal abstracts (533,560 documents and 40,002,197 words), and to forty years of the PubMed corpus (4,025,978 documents and 273,853,980 words). Our experiments show that CLDA executes two orders of magnitude faster than the original DTM implementation and, as a result, can be practically used for processing much larger datasets than is possible with DTM.

We evaluate CLDA on the quality of the topic models that are produced. Perplexity is traditionally used as the measure of the quality of a topic model, and describes how close the model fit is to a held-out dataset. Our results show that the perplexity of CLDA is comparable to that of DTM and LDA on the same dataset and using the same global topic parameter. We also find that CLDA is capable of much stronger perplexity results when utilizing the ability to examine different segmentation approaches.

A key challenge we face in this research is the identification of a robust measure for comparing the similarity of two different topic models. Perplexity does not measure how similar the topics are to each other for different models. For the goal of comparing the actual topics produced by the DTM and CLDA modeling approaches we need an additional metric. For this last evaluation we choose to apply set-based measures, including the Sørensen-Dice coefficient and the Jaccard index, to compare the top most frequently occurring subset of words of the topic. Our results show that topics generated by CLDA are similar, using set-based metrics, to those generated by DTM and LDA.

CLDA has a number of promising characteristics. Unlike DTM, which requires that all time steps have the same number of topics, CLDA allows for the birth and death of topics between time steps. CLDA also facilitates in-depth exploration of topics within time segments without sacrificing information about global trends. CLDA scales favorably with the number of processors, the size of the document corpus, and the number of topics across even very large datasets. In addition, analysis can be performed on the composition of each global topic in each segment, allowing a better fit for individual time steps than DTM. Matching the original topic mixtures to their representative centroids also enables comparison across time of global topics and analysis of how the global topics change over time.

Background and Related Work

Topic Modeling

Clustered Latent Dirichlet Allocation is presented as an extension of LDA for analyzing large corpora that can be partitioned into segments [3]. LDA incorporates a number of assumptions. These include assumptions that words are unordered, topics are distributions of words, and multiple topics can contribute to a document that is a mixture of topics. Additionally, there is an assumption that the prior distribution of each topic is a Dirichlet distribution, which distinguishes it from more generalized methods. Only the output of the model (i.e., documents) can be observed directly. The topics and topic mixtures are latent variables that must be inferred. Detailed relevant background on LDA including implementation details of PLDA [12] and PLDA+ [9] can be found in a long version of this paper [13].

Dynamic Topic Modeling

One of the assumptions of LDA is that every document is equally important, but when evaluating documents over a long span of time this is problematic. For example, since language changes over time, the classification of a document written in 2000 should be based more on how it compares with documents written in the 1990s than in the 1900s. This problem can be partially sidestepped by considering blocks of time as separate collections, and performing LDA on each of them independently. This has the advantage of reducing the size of the corpus being used on any given task, which makes the method faster. But, without further processing, it has the disadvantage that it loses the information about how a topic evolves over time. CLDA addresses this problem directly.

DTM is one approach to the time dependency problem [5]. Documents are sorted into discrete time steps, each containing a sizable corpus of its own. Each time step has its own topics, multiplying the size of the output by the number of time steps. We refer to the set of topics linked to each other over time as a *global topic*, where its representation at a given time step is a *local topic*. During each iteration, topics are updated by repeated inference on documents in their own time step, and also by consideration of the topic's form in the preceding time step. Detailed relevant background on DTM and related approaches can be found in a long version of this paper [13]. Investigation of dynamic topic modeling approaches that improve these weaknesses and increase performance is an active area of research [14]–[22].Table I provides an overview of some of the most important developments and how they compare to CLDA. Note that while a parallel DTM [8] is known, their implementation and test data are not available, so we cannot make direct comparisons.

METHOD DESCRIPTION

CLDA uses a data decomposition parallelization strategy. The data are split into multiple segments and LDA is applied to estimate local topics in each segment in parallel. Then, local topics are merged and clustering is used to calculate global topics on the merged local topics. Algorithm 1 provides the pseudocode and Figure 1 provides the flowchart for CLDA. We describe the steps of CLDA in detail.

Input : Number of segments S, number of local topics L, number of global topics K
Output: $L \times S$ local topics, grouped into K clusters
`SPLIT` text corpus into S segments ;
for *all segments* $s \in \{1, ..., S\}$ **do** `// runs in parallel`
  `APPLY LDA` to estimate local topics $\{t_s^i\}_{i=1}^L$;
end
`MERGE` ($\{t_s^i\}_{i=1}^L$ for $s \in \{1, ..., S\}) \to U$; `// runs in parallel, see Algorithm 2` ;
`CLUSTER` U into K global topics ;

Algorithm 1: Pseudocode for Clustered Latent Dirichlet Allocation (CLDA)

STEP 1: SPLIT text corpus into S segments

First the corpus is divided into S segments on which LDA will be applied. In our application we divide the data according to the naturally occurring disjoint segments of time steps (yearly data for each corpus) and journal publication (for the Pubmed corpus). Other applications might divide data by geographical location or data source. The division of the overall corpus into individual segments can be performed as a serial task or in parallel. The vocabulary is distributed to all tasks prior to the LDA computation, and the remaining data manipulation before each LDA executes independently on the individual, smaller, segments. The smaller the segments, the more efficiently this approach can utilize parallel resources, as segments can be processed in parallel to each other and each individual segment will be processed more quickly due to the reduced data size. However, segments that are too small risk compromising result quality, as discussed later.

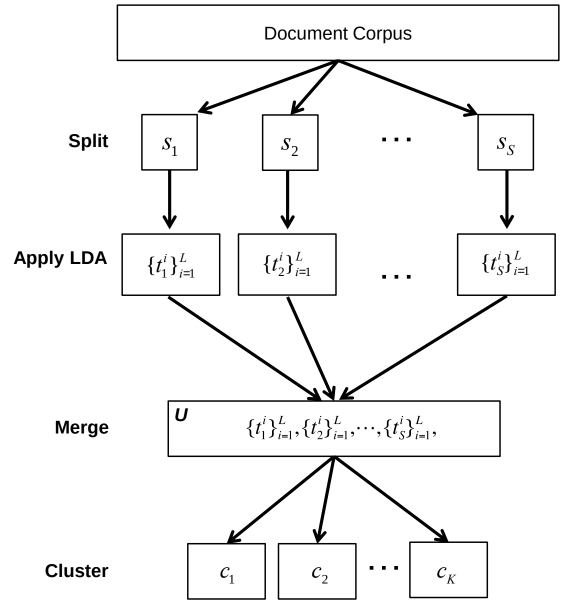

Fig. 1. **Flowchart of Algorithm 1.** This figure shows how topics are generated and combined in parallel. Details for each step are described below.

LDA requires that the number of estimated topics, L, be selected *a priori*. The number of local topics L can be larger or smaller than the number of global topics K. We have found that often better results are obtained when the number of local topics L is larger than what may be expected for global topics. The larger number of topics at the local level allows for small topics to be discovered, and for greater breadth of a topic that is unusually well represented. If many topics represent the same subject at any given segment, these are clustered together.

In this paper we describe the case where L is constant for each segment. Extensions are possible where a different L is set for each segment, either for domain-specific reasons or after determining the locally optimal number of topics through cross-validation.

STEP 2: APPLY LDA to estimate local topics

In the second step the documents in each segment are analyzed with LDA. LDA can run concurrently on separate processors (or groups of processors, if using parallel implementations of LDA such as PLDA in our experiments) for nearly perfect parallelism. This step results in a collection of L topics $\{t_s^i\}_{i=1}^L$ at each segment $s \in \{1, ..., S\}$, for a total of $S \cdot L$ local topics (whose merged union is denoted by U in Algorithm 1) that are clustered in the next stage.

STEP 3: MERGE local topics

The third step is to merge the emitted topics into a single collection U which can be input to the clustering method. At

TABLE I
Comparison of existing topic modeling approaches and CLDA

	CLDA	LDA	PLDA+	TOT	HDP	parallel HDP	DTM	iDTM	parallel DTM
Reference		[3]	[9]	[23]	[24]	[25]	[5]	[14]	[8]
Parallelized	✓	-	✓	-	-	✓	-	-	✓
Includes time component	✓	-	-	✓	(✓)†	(✓)†	✓	✓	✓
Evolution of topics	✓	-	-	✓	-	-	✓	✓	✓
Allows for birth/death of topics	✓	-	-	✓	-	-	-	✓	-
Unlimited number of segments	✓	-	-	✓	✓	✓	✓	✓	✓
Multiple segmentation options	✓	-	-	-	✓	✓	-	-	-

Notes:† HDP was built for nested data. Similar to the modeling approach presented in this paper, HDP could be applied to time-segmented data to estimate changes in topics over time.

the conceptual level this requires concatenating the emitted topics into a single list, but in practice this step is more involved. The individual outputs $\{t_s^i\}_{i=1}^L$ have indexing entries that must be removed before they can be concatenated. The entries are then re-indexed to match the input requirements of the chosen implementation of k-means (here [10]). It is also necessary to ensure that the generated topics are comparable. LDA acts on a vocabulary consisting of everything that appears in its source documents, and produces topics with a value for each element in the vocabulary. If a word appears in one document collection but not another, the resulting topics are not directly comparable. As such, if any of the segments does not contain the full vocabulary, it is necessary at this stage to add the missing entries to the topics, as shown in Algorithm 2. The entries are added with zero contribution to the topic. Depending on application, it may be valuable to instead set these values to some small value ϵ, or set $\{t_s^i(w)\}_{i=1}^L \Leftarrow \{t_s^i(w)\}_{i=1}^L + \epsilon$ for all topic entries. Either adjustment can be performed in this stage with minimal performance cost, but our implementation leaves the topics untouched beyond the addition of zeros for missing words.

Input : Number of segments S, full vocabulary W, local vocabularies W_s, local topics $\{t_s^i\}_{i=1}^L$
Output: Topic set U, containing all local topics in a shared vocabulary space
for *all segments* $s \in \{1, ..., S\}$ **do** `// runs in parallel`
  for *all words* $w \in W$ **do**
    if $w \notin W_s$ **then**
      add w to W_s ;
      for *all local topics* $\{t_s^i\}_{i=1}^L$ **do**
        $t_s^i(w) \leftarrow 0$;
      end
    end
  end
end
$U \leftarrow \bigcup_{s=1}^S \{t_s^i\}_{i=1}^L$;

Algorithm 2: Pseudocode for MERGE step

In addition to ensuring the local topics are comparable in dimension, they must be comparable in scale. Some LDA implementations, including PLDA, provide varying magnitudes for topic vectors based on their occurrence in the data. The goal of CLDA is to cluster the local topics based on the meaning, and we assume that all local topics are equally weighted. As such, the topics are normalized before clustering them. This operation is straightforward and has no dependence on other topics or other segments, and can thus be done independently before the merge, or all at once afterwards. Our implementation performs this normalization before the merge, but there is no difference in the results either way.

STEP 4: CLUSTER local topics

The fourth step is to combine local topics into global topics. The k-means clustering requires that the number of global topics, which we denote as K, is set *a priori*.

In the extreme cases, $K = 1$ defines a single cluster containing all local topics, and $K = (S \cdot L)$ defines a cluster for every topic individually. If $K > L$, not every global topic will have a representation at each segment, which means global topics will disappear and/or reappear. If $K \leq L$, topics *may* disappear and reappear at individual segments, depending on the results of the clustering. We consider this to be an advantage of the method over alternative implementations, such as DTM, which assumes that topics are universally represented over the entire time period.

k-means clustering is sensitive to its initialization. In CLDA, this may result in different topics. There are ways to evaluate the output of k-means across different initial values, including inter-class sum of squares, but generating initial values that are sufficiently different from each other is still a data-dependent challenge. Running LDA on the entire corpus provides a set of topics that make an intuitive set of initial values for clustering, regardless of data properties. While this can be done concurrently with running LDA on individual segments, it takes longer to complete than individual segments due to the larger corpus. To avoid the performance impact, we can use fewer iterations on the full corpus than we do on the local segments. Alternatively, instead of running LDA on the entire corpus, choose K random topics from U as the initial values. In our implementation, we support both options.

STEP 5: OUTPUT local topics and global topic assignments

Once clustering is complete there are two important outputs. The first is the centroids themselves, each usable as a topic, and the second is the assignment of the original topics to their corresponding global clusters. The centroids are useful on their own, and provide summary information DTM does not. DTM does not provide a general vision of a given dynamic topic, only its local topics at each time step. CLDA provides both a segment-agnostic version of a topic and a varying number of local topics at each segment, including potentially none at all, which would indicate that the topic was not meaningfully present in that segment.

EXPERIMENTAL VALIDATION

We evaluate our method on three different data sets, which are summarized in Table II. Our first data set is a collection of all NIPS papers from 1987 to 2003 [26], which we selected because it is a widely used data source for evaluating the quality and performance of topic models. The NIPS data contains 2,484 documents (about 150 documents per time segment), 14,036 unique words, and 3,280,697 tokens. Our second data source is a collection of abstracts from published articles in computer science provided by Elsevier and pre-processed using the open source HPCC Systems platform by LexisNexis. This data set covers the same number of time segments as the NIPS data, but includes a much larger number of documents ($N = 533,588$) with about 31,000 documents per time step. The computer science abstracts data contains 17,998 unique words and 46,446,184 tokens after our pre-processing. We remove stopwords using the NLTK stopword definition, as well as any word that appears fewer than 100 times or in fewer than 10 unique documents. We also remove single-character words and words containing only symbols or numbers. This corpus is much broader in scope and hence requires a greater number of topics to describe the documents. Our third corpus is a forty year collection of article abstracts from PubMed for the time period from 1976–2015, which contains 4,025,976 documents after we removed non-English abstracts and all articles published in journals with less than 10,000 total number of articles. The PubMed corpus contains 4,025,976 documents, 69,742 unique words and 386,847,695 tokens after being pre-processed the same way as the computer science abstracts data. We use this dataset to demonstrate the scalability of the approach, and the use of alternate segmentations of the same data. *Having a scalable implementation capable of producing dynamic topics is extremely important in the analysis of biomedical literature to accelerate scientific discovery.* Examples include analysis of scientific trends [27], [28] and topic modeling based hypothesis generation [29] which makes the ABC model [30], [31] more interpretable. Non-scalable topic modeling often represents a major bottleneck in qualitative scientific literature analysis [32].

All experiments were performed on Clemson University's Palmetto Cluster, using Intel Xeon Gold 6148 nodes with 40 cores, 748 GB of RAM, and 56 Gbps interconnects. Our implementation of CLDA for these experiments utilizes PLDA [9] for the LDA stage, and k-means [10] for the clustering stage. The data manipulation code connecting the stages is written in Python 2.7 and Python 3.6, and can be found here: [11]. Future work will explore alternative implementations of the various CLDA components.

TABLE II
Overview of data used for evaluation

	NIPS	Computer Science Abstracts	PubMed
Time period	1987–2003	1996–2012	1976–2015
No. of time segments	17	17	40
No. of journal segments	N/A	N/A	208
No. of documents	2,484	533,588	4,025,976
Vocabulary size	14,036	17,998	69,742
Total word tokens	3,280,697	46,446,184	386,847,695

Performance

We first compare CLDA's runtime with the DTM implementation by Blei and Gerrish [7] on the computer science abstract data. We execute the DTM model on 20 topics, and the CLDA model using 20 global topics and 20 local topics per segment. Blei and Gerrish's DTM implementation is not parallelized, and PLDA does not run in serial, so we are unable to perfectly match the resources used. All timing experiments were run on the same hardware, only varying processor count.

The results shown in Table III demonstrate that the algorithm is orders of magnitude faster than the original implementation of DTM. This is unsurprising; the primary operation of consequence is the LDA phase of the algorithm, where our implementation utilizes the highly optimized PLDA. The other operations largely consist of data manipulation to normalize or rotate files, and the clustering step. However, the clustering input is small compared to the size of the input data, so in our experiments k-means converged in seconds.

TABLE III
Runtime results on computer science abstracts.
CLDA operations can be completed in parallel; the maximum runtime of any CLDA operation was 3 minutes, but the sum of all CLDA operations is presented for fairer comparison of resource utilization.

	# of Cores	Iterations	Sum of Walltime (minutes)
DTM	1	100	3497
PLDA	12	1,000	41
CLDA	12	1,000	33

We next evaluate CLDA on the PubMed data, which is an order of magnitude larger than the computer science abstracts data. Additionally, we have enough data for enough different journals to examine the PubMed data using each of time segmentation and journal segmentation. *The ability to examine alternative data segmentations is a key contribution of CLDA.* The LDA phase of the algorithm concluded in 22 minutes on this dataset using 1,000 iterations and 12 cores per segment, a total of 480 cores overall. This performance takes advantage of multiple levels of parallelism, as CLDA is able to process each segment simultaneously and PLDA can leverage distributed computing within a segment. DTM applied to the same data with the same number of iterations would take approximately 29 weeks to complete given our earlier findings.

Quality

In order to be useful, the topics produced by the algorithm must be either very similar to those produced by DTM, or superior to them. Measuring the quality of a topic model is an open question, but a standard approximation is the perplexity metric. This metric evaluates how likely the topic model is to generate a set of provided documents. A lower perplexity indicates a model more closely fits the documents. As perplexity is a function of probabilities rather than direct model parameters, it can be used to compare different models over the same input.

Perplexity is calculated using

$$\text{perplexity} = \exp\left(-\frac{\sum\limits_{d\in D}\sum\limits_{w\in d}\log P(w|d)}{\sum\limits_{d\in D} N_d}\right) \tag{1}$$

where d denotes a document in the corpus D, w denotes a word, and N_d denotes the number of tokens. We use a hold-out set to evaluate perplexity, executing the model on 90% of the data and testing it on the remaining 10%. To evaluate the probability of generating a word $P(w|d)$, it is necessary to generate topic mixtures for held-out documents. We use the code provided with PLDA for this task [12], but a more thorough study of this problem can be found in Wallach et al. [33].

Note that in the case of CLDA, documents can be evaluated in the context of both local topics and global topics. We believe that evaluating document level perplexity using the local topics better represents the power of CLDA, and is comparable to how perplexity must be evaluated on DTM, since DTM lacks global topics.

Table IV provides perplexity results for CLDA, DTM, and PLDA estimated on the full computer science abstract data. The DTM model was executed for 58 hours using 20 topics, while the CLDA models were executed in 10 minutes using $K = 20$ global topics and $L = 20$ local topics. PLDA estimated on the full data using 20 topics completed in 17 minutes. The results show that CLDA has a comparable perplexity to DTM and PLDA for this data set.

TABLE IV
Perplexity results on computer science abstracts.
These results are the averages of 10-fold cross validation. Lower scores indicate better predictive power.

	DTM	PLDA	CLDA
Perplexity	1,950	1885	1871

Similarity

The previous results indicate that our system is both very fast and has competitive perplexity to other methods. We wish to know how similar the generated topics are to those generated by DTM or LDA.

Topics are probability mass functions represented by vectors, but this is not how humans interpret them [34], [35]. Rather than look holistically at the entire vector, a human will examine the most heavily weighted words in a topic; for example, the top five. These words will provide insight as to the conceptual meaning of a topic. In order to compare the insights gleaned from a set of topics, we thus need to compare what a human compares; the words most strongly tied to a topic. For a word-wise comparison of topics as sets of important words we will use the Sørensen-Dice coefficient

$$S(A,B) = \frac{2 * |A \cap B|}{|A| + |B|} \tag{2}$$

and the Jaccard index

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

where A and B are sets; in our context, A and B are representative sets of two topics being compared. Specifically, we use the top 20 most commonly generated words in a topic as its representative set. Chang et al. [34] used the top 5 words as the core of a topic for their intruder experiment, but they were using humans to detect outliers instead of searching for broad similarity. We chose this value as it is low enough to be human-readable, but high enough to dampen the impact of minor value differences on ordering. However, this value is still arbitrary. Future work will explore other means of transforming topics into sets.

We compared the systems using both measures. We compared the global topics to each other by comparing their means. For our system, these are emitted by clustering, but for DTM we averaged the local topics together. Both the Sørensen-Dice coefficient and the Jaccard index compare single sets to each other. Comparing the outputs of DTM and CLDA requires assigning a one-to-one matching between the two collections. If the topics generated by DTM and CLDA both include a topic describing the same concepts, these two topics will match more closely than they match other topics. If this is not the case, then the topics are not similar and a low value will be obtained regardless of the optimality of matching. Our experiment utilizes this assumption by greedily matching the pair of unassigned topics that are closest to each other under the Jaccard index out of all possible pairings, repeating the process until all topics are assigned. The Jaccard index and Sørensen-Dice coefficient are calculated for each match.

The values of the global topic matches are shown in Figure 2 sorted from best to worst. We compare CLDA's global topics to those of both PLDA and DTM, and also compare PLDA to DTM directly as a reference point. We find that these comparisons all produce similar results; the global topics from each approach are all roughly the same distance from one another.

Multiple segmentations

Unlike other topic models, CLDA can be used on varying data segmentations rather than just one. In particular, we examine the PubMed data set segmented by each of time and journal. Note that while the full PubMed corpus contains articles from journals with only a comparatively small number of articles each, we have trimmed our corpus to only include

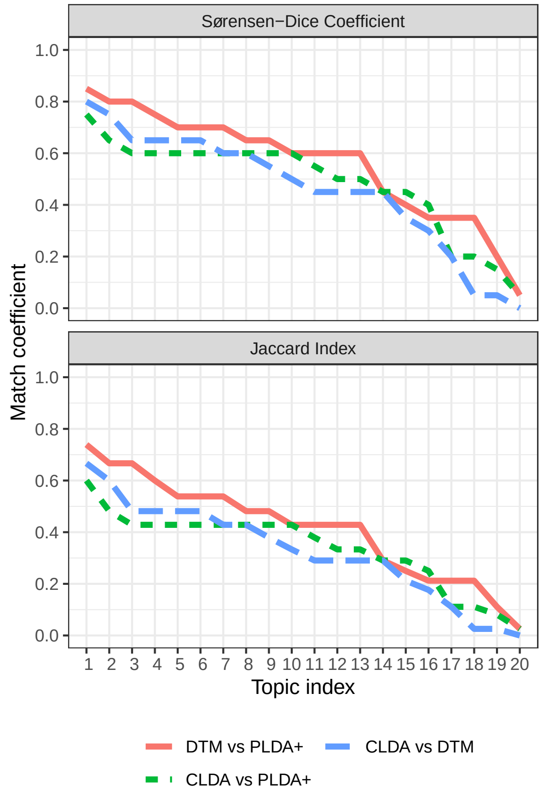

Fig. 2. Similarity of global topic centroids between DTM (estimated with 20 topics), PLDA (estimated with 20 topics), and CLDA (estimated with 20 global topics and 20 local topics) applied to the computer science abstracts data as measured by Sørensen-Dice Coefficient and Jaccard Index.

abstracts from journals with at least 10,000 documents in the collection. This way, we ensure that each journal segment contains at least 10,000 documents, rather than having many segments with only a few documents.

We compare the results of these two segmentations on several axes. We examine their runtime and perplexity, using the methods described in our above experiments. The runtime and perplexity results are summarized in Table V.

TABLE V
Runtime and Perplexity Results on PubMed corpus

	PLDA	CLDA (years)	CLDA (journals)
Walltime (minutes, max)	121	9	11
Walltime (minutes, sum)	121	143	308
Perplexity	2657	2356	1392

We find that CLDA allows for much better exploitation of parallel resources, with a total walltime an order of magnitude faster than that of PLDA. However, we do find that there is some cost to total resource utilization, especially when dealing with a large number of segments. We hypothesize that this is due to static initialization times, since the effect appears greatly magnified in the journal segmentation, which has 208 segments compared to the 40 of the year segmentation. Interestingly, this did not occur in the computer science abstract data, which had only 17 segments.

We also find that the global topics of the journal segmentation differ from those of the year segmentation. In particular, the topics generated by PLDA are substantially closer to those of the year segmentation than they are to those of the journal segmentation. We also find that while the perplexity of the year segmentation is better than that of PLDA, the perplexity of the journal segmentation is vastly superior to that; we hypothesize that CLDA is taking full advantage of the smaller and more thematically cohesive segments to produce very specific local topics.

Global and local topic dynamics

LDA can be used to capture change in topic proportions over time, by executing the model over a whole corpus and then evaluating segments of it. This does not capture any change in topic language over time, forcing each segment to use the same topics. DTM relaxes this constraint by allowing the topics to vary over time. DTM produces both a version of each topic at each segment, as well as the relative proportion of each topic at each segment, demonstrating how both language and representation change over time [5]. However, DTM fixes the number of topics across time, with each overall topic having one representative per segment. CLDA relaxes this further, allowing a global topic to have any number of local representatives at each segment, including zero. In addition to allowing for topics to branch out, better fitting their local data, this also allows for global topics to appear and disappear entirely.

The strength of DTM is the variation of topics over time, taking on forms better suited to their local data while remaining tied together by a common theme. Blei et al. [5] demonstrate this by examining the changing form of a topic at several time steps, as well as their changing proportions over time. CLDA produces output to provide this same type of insight into a corpus.

We show the changing topic proportions for selected topics in both the NIPS data and computer science abstract data in Figure 3. Like DTM, CLDA provides insight into the rising and falling predominance of various topics in a corpus. Unlike DTM, CLDA global topics need not be composed of exactly one topic at each segment. Figure 4 shows how a changing number of local topics represent a global topic we identify as "Computer Networks" for six selected time segments from the computer science abstract data. While these topics are all clustered together, they represent distinct ideas within the overall concept of "Computer Networks". One may focus on software defined networking, while another may focus on the communication between remote sensors. While this distinction is useful to examine, treating these as fully separate topics does not produce an accurate picture of how prevalent computer networks research is in the corpus as a whole. Clustering these topics together provides both the global insight of overall representation and local insight into a research area's subdomains.

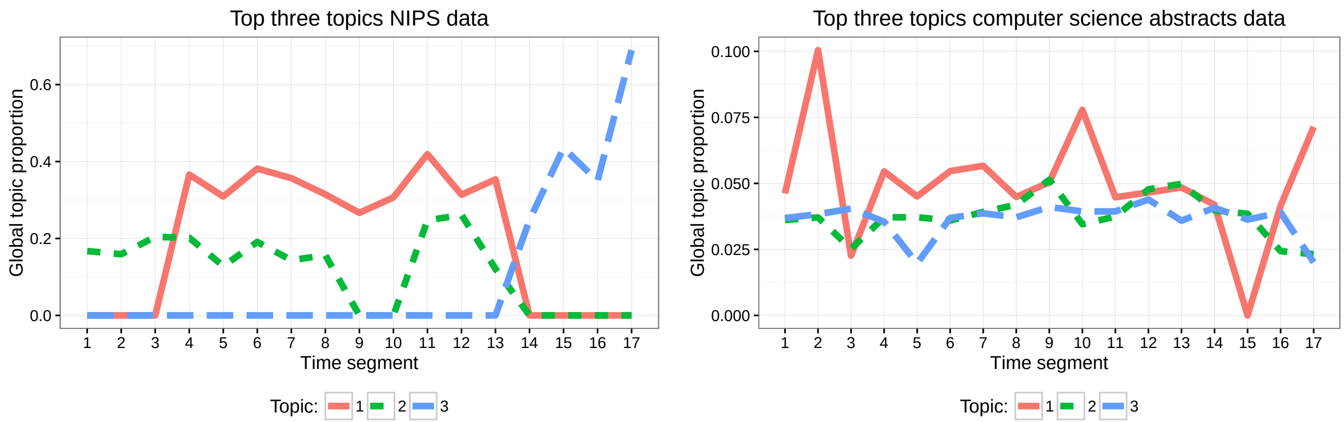

Fig. 3. Evolution of three largest global topics for the NIPS data (left panel) and computer science abstracts data (right panel).

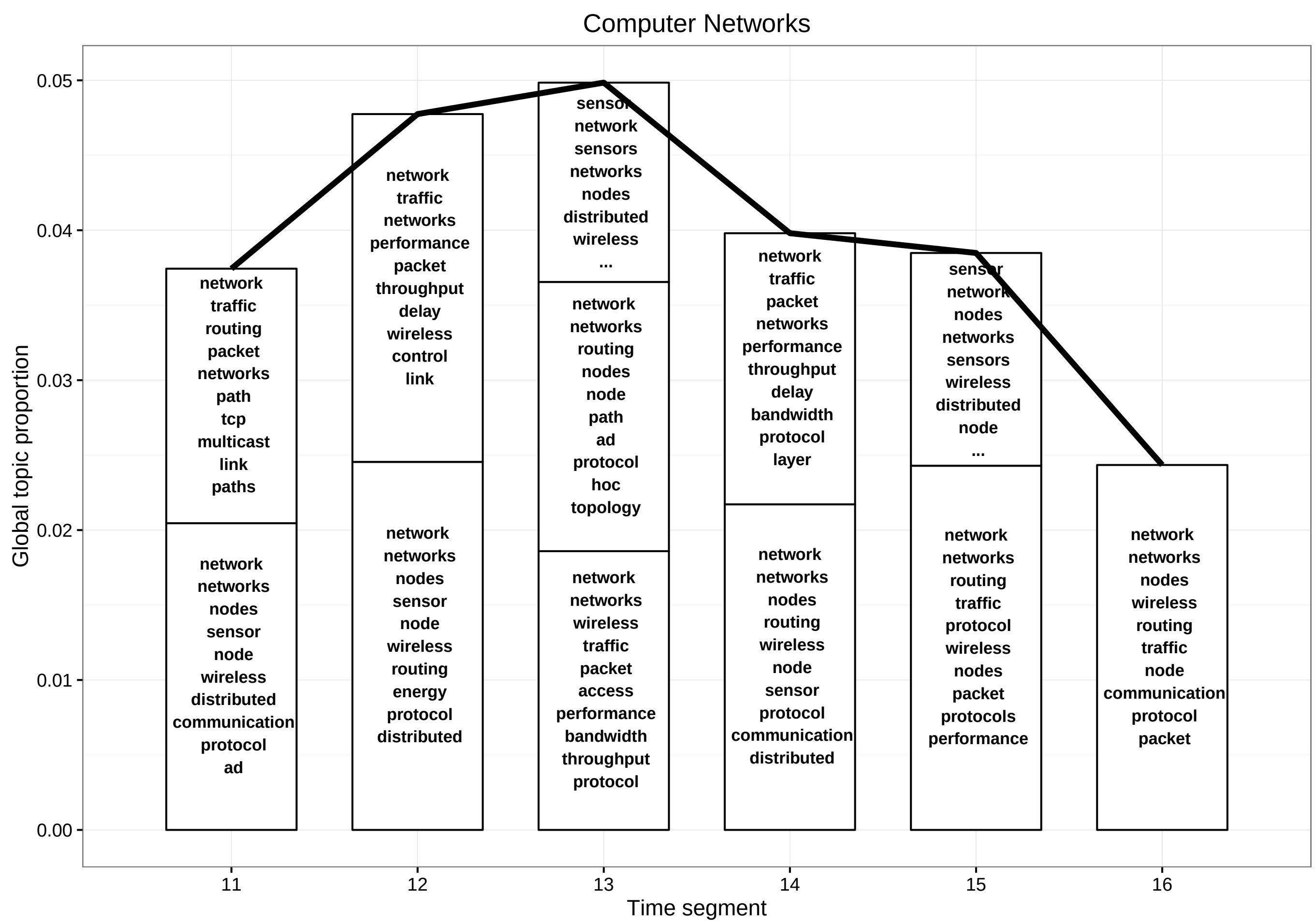

Fig. 4. Local topics for selected time segments corresponding to global topic "Computer Networks" from the computer science abstracts data using 62 global topics and 50 local topics in each segment. Each bar lists the top words in each local topic. The height of each bar corresponds to the proportion a local topic contributes to the global topic.

Conclusion

We have constructed and evaluated CLDA, a method for analyzing topic dynamics in text data. This algorithm leverages existing parallel components to increase speed and facilitate the use of large corpora. It begins by discretizing the data into disjoint segments, and applying Latent Dirichlet Allocation on each segment in parallel. The resulting local topics are merged and then k-means clustering is applied, producing a number of global topics. Each global topic is composed of a number of local topics in each segment, and provides a summary of the cohesive theme across segments. Our system is built using PLDA [12] and parallel k-means clustering [10].

We find that our system performs faster than the original implementation of DTM by two orders of magnitude. CLDA also has perplexity comparable to that of DTM and PLDA. The topics generated by CLDA, DTM, and PLDA are all broadly similar to each other. CLDA shows a more detailed composition of local topics than is possible with DTM, and enables global topics to emerge and disappear over the time span. We also demonstrate CLDA's application to examining multiple segmentations of the same data, a capability not possessed by other models we are aware of. Taken together, these results show that CLDA is a promising approach for modeling dynamics in topics estimated from textual data. The implementation of CLDA is available at [11].

REFERENCES

[1] Z. Li, J. Tang, X. Wang, J. Liu, and H. Lu, "Multimedia news summarization in search," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 3, pp. 33:1–33:20, Feb. 2016. [Online]. Available: http://doi.acm.org/10.1145/2822907

[2] Z. Yin, L. Cao, Q. Gu, and J. Han, "Latent community topic analysis: Integration of community discovery with topic modeling," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 4, pp. 63:1–63:21, Sep. 2012. [Online]. Available: http://doi.acm.org/10.1145/2337542.2337548

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[4] M. Cataldi, L. D. Caro, and C. Schifanella, "Personalized emerging topic detection based on a term aging model," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 7:1–7:27, Jan. 2014. [Online]. Available: http://doi.acm.org/10.1145/2542182.2542189

[5] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 113–120.

[6] K. Farrahi and D. Gatica-Perez, "Discovering routines from large-scale human locations using probabilistic topic models," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 1, pp. 3:1–3:27, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1889681.1889684

[7] D. M. Blei and S. M. Gerrish, "Code for estimating dynamic topic models," 2011. [Online]. Available: https://github.com/blei-lab/dtm

[8] A. Bhadury, J. Chen, J. Zhu, and S. Liu, "Scaling up dynamic topic models," *arXiv preprint arXiv:1602.06049*, 2016.

[9] Z. Liu, Y. Zhang, E. Y. Chang, and M. Sun, "Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 26, 2011.

[10] W.-K. Liao, "Parallel k-means data clustering," Northwestern University, 2013. [Online]. Available: http://users.eecs.northwestern.edu/ wkliao/Kmeans/

[11] "Clustered LDA: implementation," https://github.com/groppcw/CLDA, 2017.

[12] Y. Wang, H. Bai, M. Stanton, W.-Y. Chen, and E. Y. Chang, "Plda: Parallel latent dirichlet allocation for large-scale applications," in *Algorithmic Aspects in Information and Management*. Springer, 2009, pp. 301–314.

[13] C. Gropp, A. Herzog, I. Safro, P. W. Wilson, and A. W. Apon, "Scalable dynamic topic modeling with clustered latent dirichlet allocation (clda)," *arXiv preprint arXiv:1610.07703*, 2016.

[14] A. Ahmed and E. P. Xing, "Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream," in *Proceedings of the 26th International Conference on Conference on Uncertainty in Artificial Intelligence*, 2010.

[15] Q. Diao, J. Jiang, F. Zhu, and E.-P. Lim, "Finding bursty topics from microblogs," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 536–544.

[16] A. Dubey, A. Hefny, S. Williamson, and E. P. Xing, "A nonparametric mixture model for topic modeling over time." in *SDM*. SIAM, 2013, pp. 530–538.

[17] C. Chen, N. Ding, and W. Buntine, "Dependent hierarchical normalized random measures for dynamic topic modeling," *arXiv preprint arXiv:1206.4671*, 2012.

[18] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and L. Giles, "Detecting topic evolution in scientific literature: how can citations help?" in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 957–966.

[19] Y. Tu, N. Johri, D. Roth, and J. Hockenmaier, "Citation author topic model in expert search," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 2010, pp. 1265–1273.

[20] S. Xu, Q. Shi, X. Qiao, L. Zhu, H. Jung, S. Lee, and S.-P. Choi, "Author-topic over time (atot): a dynamic users' interest model," in *Mobile, Ubiquitous, and Intelligent Computing*. Springer, 2014, pp. 239–245.

[21] K. W. Lim and W. L. Buntine, "Bibliographic analysis with the citation network topic model." in *ACML*, 2014.

[22] Y. He, C. Lin, W. Gao, and K.-F. Wong, "Dynamic joint sentiment-topic model," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 6:1–6:21, Jan. 2014. [Online]. Available: http://doi.acm.org/10.1145/2542182.2542188

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

[24] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the american statistical association*, 2012.

[25] D. Cheng and Y. Liu, "Parallel gibbs sampling for hierarchical dirichlet processes via gamma processes equivalence," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 562–571.

[26] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean Embedding of Co-occurrence Data," *The Journal of Machine Learning Research*, vol. 8, pp. 2265–2295, 2007.

[27] C. Spreckelsen and K. Kabino, "How is my field evolving?-network based analysis of biomedical scientific discourse." *Studies in health technology and informatics*, vol. 223, pp. 199–206, 2016.

[28] D.-C. Li, M. Rastegar-Mojarad, J. Okamoto, H. Liu, and S. Leichow, "A bibliometric analysis on cancer population science with topic modeling," *AMIA Summits on Translational Science Proceedings*, vol. 2015, p. 102, 2015.

[29] J. Sybrandt, M. Shtutman, and I. Safro, "Moliere: Automatic biomedical hypothesis generation system," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1633–1642.

[30] D. R. Swanson and N. R. Smalheiser, "An interactive system for finding complementary literatures: a stimulus to scientific discovery," *Artificial intelligence*, vol. 91, no. 2, pp. 183–203, 1997.

[31] N. R. Smalheiser and D. R. Swanson, "Using arrowsmith: a computer-assisted approach to formulating and assessing scientific hypotheses," *Computer methods and programs in biomedicine*, vol. 57, no. 3, pp. 149–153, 1998.

[32] J. Sybrandt, A. Carrabba, A. Herzog, and I. Safro, "Are abstracts enough for hypothesis generation?" *arXiv preprint arXiv:1804.05942*, 2018.

[33] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation methods for topic models," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1105–1112.

[34] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Advances in neural information processing systems*, 2009, pp. 288–296.

[35] W. B. Towne, C. P. Rosé, and J. D. Herbsleb, "Measuring similarity similarly: Lda and human perception," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 1, pp. 7:1–7:28, Sep. 2016. [Online]. Available: http://doi.acm.org/10.1145/2890510