A Gensim-based approach to continuous-time infinite dynamic topic modeling

by

Timothy Darryl Tucker

B.S., Kansas State University, 2021

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Approved by:

Major Professor
Dr. William Hsu

Copyright

Abstract

This report focuses on the problem of dynamic topic modeling for a document stream with a variable number of topics in continuous time. Previous solution approaches include Dynamic Latent Dirichlet Allocation (D-LDA) and online Hierarchical Dirichlet Processes (oHDP). However, models like D-LDA face the issue of requiring a fixed number of topics over time while models like oHDP can only make inferences in discrete steps of time over a minimum time quantum to allow synchronous updates to the number of topics. The dissertation of Elshamy (2012) introduced a hybrid approach consisting of two topic models: a top-level model whose hyperparameters govern the number of extant topics, and the other, a model that uses the number of topics to inform the evolution of topics in continuous time. This report consists of a new implementation of that model using the *Gensim* library – the purpose of which is to facilitate empirical exploration of DTM design aspects such as hyperparameter updating and parameter interpretation for visualization of topics over time using timestamped text corpora. The resultant implementation is evaluated using a full-text benchmark corpora that are commonly used and consist of timestamped documents. Our objective being improving the log likelihood (and perplexity) on validation documents through an understanding of hyperparameter effects on topic generation and modification. The results of comparing a prior baseline DTM to the hybrid DTM are reported along with basic visualization output for the topics found.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unlike Transformer or LLM-based approaches to topic modeling, it is known that dynamic topic models directly encode our assumptions about the importance of sequencing in collections of documents that describe an evolving body of knowledge.

As documents are *created* over time, their content changes. Changes to the content require that topics adapt (or evolve) to accommodate the evolving body of knowledge. Models like Latent Dirichlet Allocation[1] and Hierarchical Dirichlet Processes[2] do not encode encode this assumption. Instead, they treat documents as being exchangeable. *Exchangeability*, in this context, refers to an object's ability to be swapped with any other object in a stream of information without affecting any inferential procedures. While exchangeability is very significant to words in topic modeling, it is not always extendable to whole documents. For example, documents generated at different times are not considered exchangeable (see Section 2.2 for further details). Dynamic topic models (DTM) are seen as a method of working with documents that are exchangeable in only the partitions of time that they belong to (e.g., two documents from 2004 are exchangeable, documents from 2004 and 2005, respectively, are not exchangeable). The ciDTM[3] is an example of a DTM implemented with these assumptions.

This report focuses on the re-implementation of a continuous-time infinite dynamic topic

model (ciDTM) by Elshamy and Hsu. The original ciDTM model was implemented in the C++ language. However, most ML work is performed in the Python language. Therefore, in this work, I do not focus on a one-to-one re-implementation of the original ciDTM model. Instead, I choose to create a close approximation of the original model in Python. This approximating implementation is created via models already existing in software libraries.

In this work, I show that a ciDTM can be mostly or fully recreated so long as there are two models that satisfy the properties described by Elshamy and Hsu – one model per property. Following recreation of the model, I seek to also address a secondary research question. Say that I have a ciDTM (original or re-implemented). I want to improve the log-likelihood over a set of documents via modifications to an infinite dynamic topic model's (iDTMs) hyperparameters. If I assume that the approximate number of topics at any time step T affects the log-likelihood of the data, then how can I modify the hyperparameters of an iDTM such that the topic count (more or less) predictably changes?

The remainder of this paper is as follows: Chapter 2 expands on the concept of topic modeling, dynamic topic modeling, and the original ciDTM formulation. Chapter 3 follows with an explanation of the *Dim Sum Process*, a process crucial to the development of the original ciDTM. I then give an explanation of how the ciDTM may be reconstructed through the Dim Sum Process and *Gensim*[4]. Chapter 4 will outline the experimentation plan. Chapter 5 provides a textual and visual description and interpretation of the results obtained from the experiments described in the previous chapter. Finally, Chapter 6 closes the paper with a brief discussion about the strengths and weaknesses of the implementation, as well as some future work.

Chapter 2

Background

In this section we cover some of the background knowledge of topic modeling, dynamic topic modeling, and the continuous-infinite dynamic topic model.

2.1 Topic Models

Topic modeling describes a set of unsupervised learning models that create groupings (clusters) of documents based on their underlying thematic structure.

2.1.1 A Probabilistic Approach to Modeling Topics

According to Churchill and Singh, topic modeling is a means of summarizing a collection of documents into groups that are represented by words found to commonly co-occur. In topic modeling, word co-occurrence is significant because, often times, words that co-occur likely derive from the same concept or topic. For example, picture someone reading a document and observing the terms: epinephrine, phenergan, and promethazine. Assuming that they know that these are names belonging to medicines, a natural assumption would be that the document they are reading is likely related to the field of medicine (pharmacology, specifically).

The word co-occurrence assumption for topics, while a very simplifying assumption for how to categorize words by topic, is fundamental in topic modeling. It provides us with a statistical basis for inferring how each word in a document relates to its topical structure.

If we break down documents into their principle components, we see that they are composed of words. While various other structures may be present in a document, words are the only perceptible unit of knowledge. Each word in a document has some relationship to the topical structure of the document. However, this relationship is not explicitly known. So, given the words of a document, we would like to infer how each word in the document is related to its topical structure. The original process for deciding how a word relates to a topic could be complex. Thus, we rely on a simplifying assumption for how that relationship was generated – word co-occurrence.

It is worth noting that word co-occurrence is, generally, not enough to determine the topical structure of a document. Words that co-occur may belong to the same topic. However, it is rare that a word is isolated to a single topic. Similarly, it is rare that most literature is confined to a singular topic. Returning to the medicine example from before, the names of those medicines could have appeared in a paper that was about analyzing and comparing the chemical composition of those medicines. This would align more with the topic of Chemistry rather than Medicine. However, words from both topics compose the document. In essence, documents may contain a mixture of topics, with one topic dominating most of a document. Thus, when considering topic proportions and how words relate to topics, there is a degree of uncertainty about the topics said to exist in a document or set of documents. We not only want to reason through our uncertainty about what topics exist, but also how much of each topic composes a document. Topic models utilize probability theory to reflect these uncertainties over a set of topics said to describe a set of documents.

2.1.2 Modeling Topics as a Generative Task

Understanding the inherent uncertainty regarding topics and topic proportions in documents, Blei et al. adopted a generative perspective for topic modeling. In the generative perspective, it is assumed that for some set of documents, some process that allowed for the exact creation of the whole set. This, of course, includes the topics mixtures from which the set of documents (and thus, the words composing them) were derived. To re-frame the problem of finding the set of topics and topic mixtures into the generative perspective, Blei et al. had to make some fundamental assumptions. First, words are the value that we can observe. Any other values such as topic-term weights describe latent structures. Lastly, all values, even for latent parameters, are sampled from some probability distribution. Words, for example, are sampled from a multinomial distribution. Following this simple set of assumptions, Blei et al. proceed to lay the foundation for the first generative probabilistic topic model: Latent Dirichlet Allocation (LDA). The process that they describe can be summarized as the following: for every document, there are N words that are distributed according to a Poisson distribution. Topic proportions θ for a document are sampled from a Dirichlet distribution over the $K-1$ simplex of K topics. Each word in a document is assigned a topic z_n from the topic mixture θ. The probability of a word w_n is conditioned on the choice of z_n and the topic-term matrix β that describes the contribution of each word to describing a topic. So, by reformulating the original problem of finding topics and topic proportions into a generative problem, the goal became one where the model must find values for the latent parameters such that these values maximize the likelihood that the selected topic mixtures selected allowed the set of documents to be created.

While the generative process for LDA carries many simplifying assumptions, it is also the basis for many other topic models in the domain of topic modeling. For example, the Hierarchical Dirichlet Process by Teh et al. is an extension of the generative process for LDA. The difference between the two models being that the hierarchical process assumes that there exists a global set of topics from which all document-level topic mixtures derive.

As the authors note, this is equivalent to saying that the topics at the document-level form a subset of the global set of topics.

2.2 Dynamic Topic Models

Generative probabilistic topic models are segmented into two distinct categories: atemporal and temporal. Atemporal topic models describe a group of topic models that assume all documents were generated at the same time T. Periodical news reports are a good example of both atemporal and temporal processes. While the articles in one publication of a periodical may have been created at different times, to an outside observer, each publication was created at the same time – that time being whatever the date is on the publication. This is the type of assumption that atemporal models have for all documents. Now, suppose that we do not limit our examination of a news periodical to just one publication. Instead, we examine the periodical across some time interval. In such a case, each publication belonging to a periodical was created at different points of time $T = (t_1, t_2, t_3, ...)$. The ordering of these documents in a particular sequence denotes an evolution in the type of content over time. Capturing this evolution of content over time describes the behavior of a temporal topic model. LDA and HDP are examples of atemporal models.

As a minor aside, temporal topic model behavior may sound like *concept drift*, however, the ideas are different. Concept drift indicates some change in the relationship between the distribution of the input data to its concepts. It is something that a model was unable to account for without the aid of new data to create a proper mapping. On the other hand, temporal topic models attempt to account for changes in concepts or changes in the distribution of words-to-topics over time. They do this by treating evolution over time as a stochastic process and having the current state depend on information from the previous state. In probability theory, we would represent this relationship as the number of topics K_t being dependent on information of any preexisting topics K_{t-1}, for some time $t \in T$.

Temporal topic models adhering to a generative process typically follow the dynamic topic modeling (DTM) process outlined by Blei and Lafferty. This process, as the authors note, is an extension to the original generative process for LDA. In the dynamic topic model formulation, the α and β values of a dynamic topic model for a time partition $T+1$ are dependent on their respective values in the previous time partition T. This dependency between parameters is also referred to as *chaining* by the authors. Note that because β describes the topic-term matrix, it is also representative of the topics themselves. Changes that occur in the topic-term matrix over time reflect changes in content over time too. The chaining of α and β with their respective components in the next time slice, as stated by Blei and Lafferty, reflect a smooth transition of topic and topic proportions as the topics evolve from a time T to a time $T+1$ for the k-th topic. The authors further state that, should the relationship between the distributions of these parameters break, then the framework they described would reduce to a set of independent topic models – that is, there would be T topic models with K number of topics each.

2.3 Continuous-infinite Dynamic Topic Modeling

A limiting aspect of the conventional dynamic topic model design is that the number of topics for a model will remain fixed over time. Elshamy and Hsu notes that the fixed-topic structure of the dynamic topic modeling is not conducive to the behavior of topic generation in many aspects of literature, especially news outlets. He goes on to claim that topics have a life cycle in which they are born and, may eventually, fade into obscurity, change form, or die. Blei and Lafferty are also aware of this limitation, agreeing with Elshamy and Hsu that an ideal direction for future work on the dynamic topic modeling process is the incorporation of a life-death process for topics.

To address the fixed-topic limitation of the original design, Elshamy and Hsu developed the ciDTM. The ciDTM is a dynamic topic model that makes inferences in continuous-time

while allowing for a variable number of topics over time. Elshamy and Hsu notes that the ciDTM is named after the two components that comprise it: a continuous-time dynamic topic model and an infinite dynamic topic model. In his work, he states that continuous-time dynamic topic models (cDTM) are models that have a fixed number of topics but can make inferences in continuous-time.

The concept of infinite dynamic topic models (iDTM) is a term coined by Ahmed and Xing. They state that an iDTM models the evolution of topics, topic popularity, and word distributions over time. In an iDTM, the number of possible topics is said to be infinite.

For DPs, CRFs and Sethuraman's stick breaking construction[8] are different ways of explaining equivocal solutions to the non-parametric problem of deciding how many clusters (topics) exist in some set of data (documents). With the stick-breaking construction, it easy to see that for every subdivision of a stick, the new sticks created can be further subdivided. This can continue infinitely, which would allow for an infinite number of topics with varying mixture weights – albeit, with the expectation that at some point, most if not all topics would have a negligible weight. CRFs adhere to a similar view as the stick-breaking construction, thus, they are capable of producing an infinite number of topic clusters for a whole dataset. The temporal version of a CRF is the recurrent Chinese Restaurant Franchise (RCRF). Following the RCRF, the idea of infinite number of topic clusters is now applicable to document collections that evolve over time.

One important aspect about the RCRF process is that the number of clusters that any model following a RCRF or any other similar process can produce is only *theoretically* infinite. In practice, a model that can have an infinite number of clusters is constrained to the main memory of the system it runs on. Due to these constraints, it is much more useful to think of an iDTM as having a variable number of topics rather than an infinite number. Especially because, in an iDTM, the number of topics both can increase and decrease. The life and death of a topic is determined by its presence in some user-defined window of time. For example, if a topic is not observed within three publications of a periodical, the topic is said

to be dead. Otherwise, that topic is said to be alive. Thus, an iDTM is capable of modeling a variable number of topics as content evolves over time.

Elshamy and Hsu states that the limitation to iDTMs is that they can only make inferences in discrete timesteps. Working in discrete timesteps, the inference by the model would become intractable for a small time granularity. However, Ahmed and Xing does not discuss whether or not the model is affected by small time quantums, such as seconds. In general, an iDTM will work at the level of the time quantum specified. If documents are ordered by their creation time in seconds, then the iDTM update over documents at the level of seconds. Thus, there may be very little distinction between a cDTM and an iDTM. The conventional dynamic topic model by Blei and Lafferty is an example of a continuous-time topic model. In the original ciDTM, the oHDP model developed by Wang et al. is cited as an example of an infinite dynamic topic model.

Chapter 3

Implementation Approach

Chapter 2 described the assumptions that underlie traditional topic models, the distinction between atemporal and dynamic topic models (temporal topic models, in general), and what inspired the creation for the original ciDTM. This chapter explains how to create a variant of the original ciDTM model using Gensim-based methods.

The principal components of the ciDTM are a continuous-time dynamic topic model and an infinite dynamic topic model. Recalling section 2.3, iDTMs provide an unbounded approximation for the number of topics, and cDTMs provide inferences at the smallest time granularity. Before explaining how Gensim may be used to recreate the ciDTM, it is important to, first, note what generative process the ciDTM follows.

3.1 The Dim Sum Process

The core of the ciDTM model is the assumption that the generative process for a set of time-partitioned documents follows a dim sum process. The dim sum process is much like the Chinese Restaurant Franchise Process (CRFP).

Imagine for a moment that you have a restaurant that you want to visit. The restaurant menu for your particular restaurant may differ from the menu at the same restaurant in another state. However, the dishes being served at both restaurants are always dishes from

a global menu of dishes known by all restaurants. Assume that you are the first customer to walk into your restaurant for the day. The tables are already prepped, and once seated, you may order a dish. Now, suppose a new customer walks into the restaurant. They have two choices for seating: they can sit at your table with a probability of $1/(1+\alpha)$ to enjoy the same dish or they can sit at a new table with probability $\alpha/(1+\alpha)$.

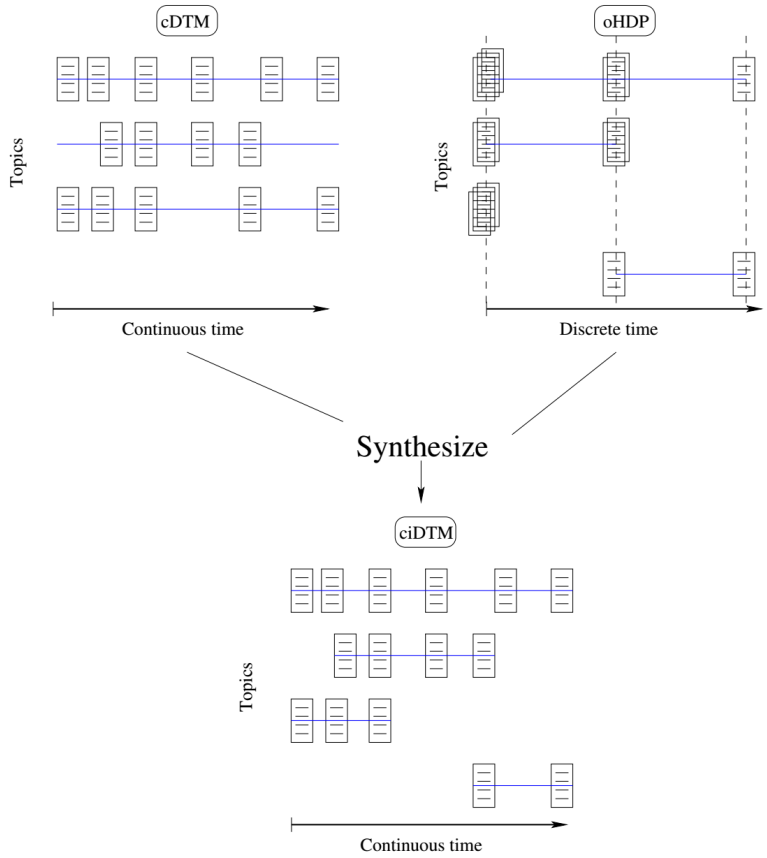

Figure 3.1: *The ciDTM expressed as a combination of two independent models. cDTMs always have a fixed number of topics, but can analyze documents at the small time quantum. oHDP works at the macroscopic level, working in discrete time steps. Synthesized, the cDTM assumes the number of topics present in the microscopic time quantum are the same number found by the oHDP model at the higher time quantum. From Continuous-Time Infinite Dynamic Topic Models, by W. Elshamy, 2012, Kansas State University. Copyright 2012 by Wesam Elshamy. Adapted with Permission*

By sitting at your table to sample the same dish that you are eating, you may think of

this as the customer inferring that the dish is worth trying because someone else is eating it. In other words, your presence is influencing their choice of table assignment. We can generalize this idea to any number of customers. The probability of the $n-1$ customer sitting a new table is $\alpha/(n+\alpha)$. The probability of the $n-1$-th customer sitting at an occupied table is dependent on the number of people currently occupying the table: $n_k/(\alpha+n)$, for the k-th table.

The above restaurant analogy can be translated to how documents are generated via a Chinese Restaurant Franchise Process (CRFP). In essence, there exists a global set of topics that can we can liken to a menu. A document denotes a restaurant in this scenario. Therefore, documents only contain topics that exist from an existing global set of topics. Restaurants must adhere to serving dishes that form some subset of the global menu. Tables, and the dishes placed on them, denote topics. Words are customer and select their topic of choice based on popularity of that topic. CRFPs allow for an unbounded number of topics, but they do not carry any temporal assumptions about how the data was generated. Therefore, all documents are treated as being generated at the same time. A CRFP that allows topics to be created, persist, or die over some period of time is said to follow a *Dim Sum Process*[3]. In a dim sum process, the global menu (set of topics) may change over time. A visual of this process is shown in 3.1.

3.2 A Gensim-based Formulation of the ciDTM

Knowing the distinction between a CRFP and the dim sum process is important because, whereas the ciDTM follows a dim sum process, the new library-based implementation follows a CRFP. This new formulation came about because of the nature of the models obtained from the library being used.

The *Gensim* is an API dedicated to topic modeling algorithms. Included in the library are two models that fit into the iDTM and cDTM categorizations. For the iDTM component,

the Gensim API provides an oHDP model. This model was based on the original design specified by Wang et al.. However, it should be noted that this model is not a perfect fit for what Elshamy and Hsu considers an iDTM because it follows a CRFP. Fortunately, in the oHDP model, not all topics are probable until they, perhaps, receive a certain set of documents that increases a topic's likelihood. For the cDTM component, the Gensim API provides the LDASeq model, which was based on the original design of the dynamic topic modeling paradigm conceived by Blei and Lafferty. The overall process is as follows:

- Take one collection from the timestamped corpus, and allow the oHDP model to train over this chunk of data.
- Obtain topic count from the oHDP model.
- Initialize a new LDASeq model with the previous document collection (if any), then train it over the current document collection.
- Repeat steps 1-3 until all timestamped collections are examined.

Chapter 4

Experiments

In this section, we will review the experiment plan for the implemented model.

4.1 Corpus Selection

Topic models vary in their needs for document sizes. Some topic models can work on text as small as tweets or text messages, others require documents with tens-of-thousands of words[5]. Both the original ciDTM and this implementation of it are topic models that work best over long texts. Scholarly publications such as, but not limited to theses, dissertations, journals, and conference publications or proceedings are ideal documents for these types of models. Just as important is the need for the selected corpus to have timestamped data to ensure proper content evolution in the dynamic topic model. Thus, I chose to utilize the NeurIPS dataset available on Kaggle[10].

The NeurIPS dataset contains 7000+ papers scraped from the NeurIPS proceedings through the years of 1987 to 2017. Each document has author data, paper titles, abstracts, and full texts. For our purposes, I use only the full-text of each document – making sure to partition all documents by year. For testing purposes, I used data from only the years 1987 through 2001 for training. For validation, I use documents for the next two years 2002 and 2003 because if topic transition should be smooth, then the topics in those years should be,

more or less, the same as 2001.

4.2 Pre-processing

Each document in the corpus underwent the following changes before being processed by the model:

- Removal of all numbers, punctuation marks, and non-alphabetic characters.
- Standardize casing for all words.
- Removal of stop words.
- Removal of words with four letters or fewer.
- Lemmatize all words in the document
- Reduce the document to a unigram BoW representation.

, where all lemmatization and removal of stop words are performed through the NLTK[11] library.

4.3 Models

For this experiment, I utilize standalone versions of the LDASeq and oHDP models from Gensim as baselines for the implementation. I do this because this ciDTM is composed of an oHDP and LDASeq model. Therefore, it is expected that its performance should be approximately equal to or better than either model on their own.

4.4 Evaluation

For evaluation of model performance, I utilize the lower bound of the log-likelihood of the data as a measure of performance. While there is no defined best value for the computed

Table 4.1: *Mean number of topics over years 1987 through 2002 of corpus, for varying values of the document-topic and topic-term concentration coefficients γ and α, respectively.*

	$\gamma = 1/100$	$\gamma = 1.0$	$\gamma = 100.0$
$\alpha = 1/100$	4	3	3
$\alpha = 1.0$	2	3	2
$\alpha = 100.0$	2	3	2

values, generally, the greater (tighter) the lower bound on the log likelihood, the better the model's performance.

4.5 Setting Model Hyperparameters

Since we are most interested in the effects on log-likelihood via manipulation of the iDTM portion of the ciDTM, I opted to focus on modifying the values of a small subset of hyperparameters of the oHDP model. These hyperparameters include α and γ.

Other hyperparameters such as the number of documents per chunk (chunksize) may be relevant because of how Gensim implements the oHDP model. By default, the model will update the variational stick values for every one chunk examined (for more information about the normal and variational stick breaking construction, refer to Sethuraman and Wang et al., respectively). However, α and γ are already well-known to affect the number of topics. Therefore, it is sensible to focus on these hyperparameters for this study.

In Gensim, the oHDP model is described using a CRFP for analogy. In actuality, it is implemented using the stick-breaking construction. Thus, the number of topics is fixed to the value selected for the model's truncation hyperparameter. In an attempt to simulate a variable number of topics in the oHDP model, I applied a thresholding function to the topic probabilities of the oHDP model. To calculate the topic probabilities, for every document in a collection, I took a weighted average over all documents in a collection. The amount of a topic that was present in a document denoted a weight that would be used in the average.

To avoid having too many topics of low probability appear over time, I selected the threshold to be 0.1 (10%) – double the value of what is typically used to denote statistical significance. I allow the oHDP to have a maximum of fifty topics since I deemed it is unlikely that fifty topics would be present within the dataset. Finally, given the approximate number of topics found by the oHDP model while testing various distant values of α and γ (refer to Table 4.1), I inferred that the number of topics present in the NIPS corpus is approximately two or three topics. Thus, I let the standalone LDASeq have three be its fixed number of topics.

To reflect uncertainty about the number of topics in the corpus, I will let $\alpha = 1$ and $\gamma = 1$ to allow a uniform beginning on the Dirichlet's simplex. I also allow the iDTM to have a maximum of fifty topics.

In these experiments, the cDTM and LDASeq have a minimum EM-step count of 5, a maximum EM-step count of ten, and a maximum LDA-inference iteration count of five.

Chapter 5

Results

5.1 Model Results

The results from the models as they trained over the data showed a negative log likelihood bound for oHDP and a consistently high log likelihood bound from the ciDTM model (see Figure 5.1). Unfortunately, due to the nature of the LDASeq model, log likelihood bounds could not be captured at any time other than the final iteration of training. However, to put the performance of LDASeq with 3 topics into perspective, the ciDTM had a value of 3934382 after finishing training over the documents from 2001. The bound for oHDP for documents was -3363952. LDASeq's computed log likelihood bound for the data over time was -18617118.

Over the validation set, the ciDTM model had a log likelihood bound of approximately -1733977. The standalone LDASeq model had a bound of -17900. Finally, the oHDP model had a bound of -6738797. So, over unseen documents, post-training, the LDASeq model has the greatest lower bound, the ciDTM follows with the second greatest lower bound, and oHDP is last.

When using log likelihood bounds as the measure for performance, we want larger values. In this case, that would mean that the ciDTM outperformed LDASeq and oHDP in

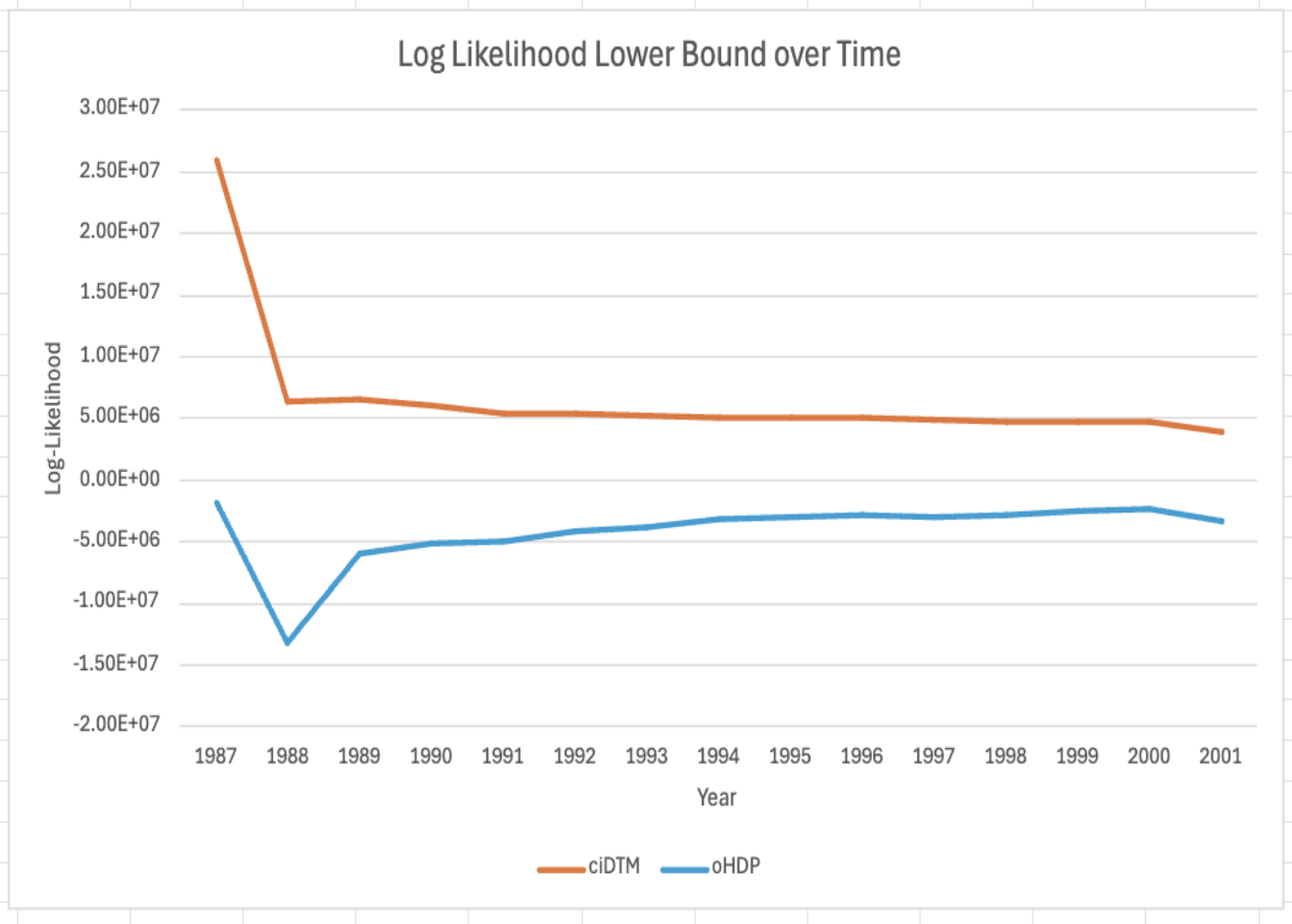

Figure 5.1: *ciDTM and oHDP log likelihood bound comparison over time.*

training. Similarly, oHDP outperformed LDASeq in training. On the other hand, LDASeq outperformed the ciDTM and LDASeq models over the validation data. Note that, while larger values should indicate better performance in a model, log likelihood bounds cannot indicate when performance increases become negligible. This is because it only represents the minimum value for the log likelihood of the marginal (or evidence).

5.2 Discussion

Overall, the ciDTM provided a log-likelihood that was several magnitudes larger than the LDASeq model's log-likelihood during training – falling slightly shorter than the LDASeq

model in an inference over unseen data. The oHDP model outperformed the LDASeq model in training, but had lower performance than both the ciDTM and LDASeq models during inference. The dramtic shift from ciDTM outperforming LDASeq and oHDP and training to being second in inference may be explainable given a number of factors. One of the most notable being how the hyperparameters were set for each model.

In the experiment, no model was optimized to maximize the log-likelihood of the data over time. Additionally, although both the LDASeq model and the LDASeq within the ciDTM implementation adhered to the same restrictions for EM steps, the LDASeq in the ciDTM was acquiring information about good initialization points for α from the oHDP model. The oHDP model, like the LDASeq model, uses variational methods to find good values for α. However, its EM-steps are not modifiable. Therefore, the oHDP model likely found suitable points of convergence, at each time step, before providing that information to the LDASeq component of the ciDTM implementation.

Chapter 6

Conclusion

In this report, I aimed to not only show that the ciDTM could be re-implemented using library-based methods, but to also discover what hyperparameters of the iDTM in a ciDTM have substantial impact on the topic count (number of probable topics). I used the lower bound on log likelihood as the performance measure, finding that the log-likelihood of the ciDTM implementation was consistently greater than the log-likelihood of the standalone LDASeq and oHDP models – provided that the models have the same restrictions on inference steps.

Despite the ciDTM implementation's performance compared to two standalone DTMs, there are some remaining questions that were not addressed in this paper.

One question is if linking the oHDP and LDASeq models via a CRFP is capable of performing as well as the original model. Without the original corpus or ciDTM available, there is no immediate way of measuring how close of an approximation the current implementation is to the original ciDTM model. Additionally, while this work sought to show that the ciDTM paradigm was extendable to any two pre-existing models that fit into the cDTM and iDTM categories, respectively, it only showed that it was possible on models known to exhibit these behaviors. An avenue for further study would be examining the possibility of making modifications to LLMs or Transformer-based models like BERTopic[12] to follow the

ciDTM paradigm.

Secondly, the original ciDTM worked off of the assumption that an oHDP model was an iDTM. However, even with its ability to take in data in stages, the model still follows a CRFP. While using an oHDP model may be satisfactory for a re-implementation of the original ciDTM model, the implementation falls short of the concept of the ciDTM. This may, in part, be due to ambiguities presented in the original work about if an oHDP model does follow an RCRF. Thus, a good first step to implementing a ciDTM is to address the problem of having an iDTM that follows a CRFP instead of the dim sum process. This can be accomplished by starting with a recreation of the original iDTM as it was specified by Ahmed and Xing.

Lastly, the model was tested over a corpus whose timestamps were only measured in years. Seeking out a corpus with large and small time quantums would be ideal for testing all aspects of the model.

Bibliography

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[2] Yee Teh, Michael Jordan, Matthew Beal, and David Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. *Advances in neural information processing systems*, 17, 2004.

[3] Wesam Elshamy and William H Hsu. Continuous-time infinite dynamic topic models: The dim sum process for simultaneous topic enumeration and formation. In *Emerging Methods in Predictive Analytics: Risk Management and Decision-Making*, pages 187–222. IGI Global, 2014.

[4] Radim Řehřek and Petr Sojka. Software framework for topic modelling with large corpora. 2010.

[5] Rob Churchill and Lisa Singh. The evolution of topic modeling. *ACM Computing Surveys*, 54(10s):1–35, 2022.

[6] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120, 2006.

[7] Amr Ahmed and Eric P Xing. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. *arXiv preprint arXiv:1203.3463*, 2012.

[8] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

[9] Chong Wang, John Paisley, and David M Blei. Online variational inference for the hierarchical dirichlet process. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 752–760. JMLR Workshop and Conference Proceedings, 2011.

[10] Ben Hamner. Nips papers, 2017. URL https://www.kaggle.com/dsv/9097.

[11] Steven Bird, Loper Edward, and Klein Ewan. *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

[12] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.