

CONTINUOUS-TIME INFINITE DYNAMIC TOPIC MODELS

by

WESAM SAMY ELSHAMY

B.S., Ain Shams University, Egypt, 2004

M.Sc., Cairo University, Egypt, 2007

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2013

Abstract

Topic models are probabilistic models for discovering topical themes in collections of documents. In real world applications, these models provide us with the means of organizing what would otherwise be unstructured collections. They can help us cluster a huge collection into different topics or find a subset of the collection that resembles the topical theme found in an article at hand.

The first wave of topic models developed were able to discover the prevailing topics in a big collection of documents spanning a period of time. It was later realized that these time-invariant models were not capable of modeling 1) the time varying number of topics they discover and 2) the time changing structure of these topics. Few models were developed to address this two deficiencies. The online-hierarchical Dirichlet process models the documents with a time varying number of topics. It varies the structure of the topics over time as well. However, it relies on document order, not timestamps to evolve the model over time. The continuous-time dynamic topic model evolves topic structure in continuous-time. However, it uses a fixed number of topics over time.

In this dissertation, I present a model, the continuous-time infinite dynamic topic model, that combines the advantages of these two models 1) the online-hierarchical Dirichlet process, and 2) the continuous-time dynamic topic model. More specifically, the model I present is a probabilistic topic model that does the following: 1) it changes the number of topics over continuous time, and 2) it changes the topic structure over continuous-time.

I compared the model I developed with the two other models with different setting values. The results obtained were favorable to my model and showed the need for having a model that has a continuous-time varying number of topics and topic structure.

CONTINUOUS-TIME INFINITE DYNAMIC TOPIC MODELS

by

WESAM SAMY ELSHAMY

B.S., Ain Shams University, Egypt, 2004

M.Sc., Cairo University, Egypt, 2007

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2013

Approved by:

Major Professor
William Henry Hsu

Copyright

Wesam Samy Elshamy

2013

Abstract

Topic models are probabilistic models for discovering topical themes in collections of documents. In real world applications, these models provide us with the means of organizing what would otherwise be unstructured collections. They can help us cluster a huge collection into different topics or find a subset of the collection that resembles the topical theme found in an article at hand.

The first wave of topic models developed were able to discover the prevailing topics in a big collection of documents spanning a period of time. It was later realized that these time-invariant models were not capable of modeling 1) the time varying number of topics they discover and 2) the time changing structure of these topics. Few models were developed to address this two deficiencies. The online-hierarchical Dirichlet process models the documents with a time varying number of topics. It varies the structure of the topics over time as well. However, it relies on document order, not timestamps to evolve the model over time. The continuous-time dynamic topic model evolves topic structure in continuous-time. However, it uses a fixed number of topics over time.

In this dissertation, I present a model, the continuous-time infinite dynamic topic model, that combines the advantages of these two models 1) the online-hierarchical Dirichlet process, and 2) the continuous-time dynamic topic model. More specifically, the model I present is a probabilistic topic model that does the following: 1) it changes the number of topics over continuous time, and 2) it changes the topic structure over continuous-time.

I compared the model I developed with the two other models with different setting values. The results obtained were favorable to my model and showed the need for having a model that has a continuous-time varying number of topics and topic structure.

Table of Contents

Table of Contents	vi
List of Tables	viii
List of Figures	ix
Acknowledgements	xiii
1 Introduction	1
1.1 Goal	1
1.1.1 Problem statement	1
1.1.2 Central thesis	2
1.1.3 Technical objectives	4
1.2 Existing approaches: atemporal models	5
1.2.1 Latent semantic analysis	6
1.2.2 Probabilistic latent semantic analysis	7
1.2.3 Latent Dirichlet allocation and topic models	7
1.3 Significance	8
2 Bayesian models and inference algorithms	11
2.1 Introduction	12
2.1.1 Factorized distributions	13
2.1.2 Variational parameters	16
2.2 Variational inference	17
3 Existing solutions and their limitations	20
3.1 Temporal topic models	20
3.2 Topics over time	21
3.3 Discrete-time infinite dynamic topic model	26
3.4 Online hierarchical Dirichlet processes	31
3.5 Continuous-time dynamic topic model	35
4 Continuous-time infinite dynamic topic model	38
4.1 Dim sum process	38
4.2 Mathematical model	40

5	Testbed development	47
5.1	News stories corpus	48
5.2	Corpora	53
5.2.1	Reuters	54
5.2.2	BBC News	55
5.3	Evaluation of proposed solution	55
6	Experimental design and results	58
6.1	Reuters corpus	58
6.2	BBC news corpus	64
6.2.1	Why ciDTM outperforms cDTM	70
6.2.2	Why ciDTM outperforms oHDP	70
6.3	Scalability	71
6.3.1	Scalability of ciDTM and oHDP vs cDTM	73
6.4	Timeline construction	74
6.4.1	Why ciDTM is better than oHDP in timeline construction	79
	Bibliography	92
A	Probability distributions	93
A.1	Uniform	93
A.2	Bernoulli	93
A.3	Binomial	94
A.4	Multinomial	94
A.5	Beta	94
A.6	Dirichlet	95
B	Dirichlet Process	97
C	Graphical models	101
C.1	Inference algorithms	103
C.1.1	Example	104
D	Information theory	108
D.1	Entropy	108
D.2	Mutual information	110
D.3	Information divergence	110
E	Table of notations	112

List of Tables

3.1	Table of notations for iDTM	30
3.2	Table of notations for cDTM	37
6.1	Setting values used for oHDP	61
6.2	Setting values used for ciDTM	63
6.3	Confusion matrix for timeline constuction using oHDP. AS stands for Arab Spring topic, and non-AS is non Arab Spring topic. The true class is presented in rows (True AS, and non-AS), and Inferred class in columns (Inferred AS and non-AS).	76
6.4	Confusion matrix for timeline construction using ciDTM. AS stands for Arab Spring topic, and non-AS is non Arab Spring topic. The true class is presented in rows (True AS, and non-AS), and Inferred class in columns (Inferred AS and non-AS).	78
E.1	Table of notations	112

List of Figures

- 1.1 Top left: the continuous-time dynamic topic model (cDTM) has a continuous-time domain; word and topic distributions evolve in continuous time. The number of topics in this model is fixed though. This may lead to having two separate topics being merged into one topic which was the case in the first topic from below. Top right: the online hierarchical Dirichlet process (oHDP) based topic model evolves the number of topics over time. The five documents belonging to the first and second topics from below were associated with their respective topics and were not merged into one topic which was the case with cDTM on the left. However, the model has a discrete-time domain which practically limits the degree of time granularity that we can use; very small time steps will make the model prohibitively expensive to inference. Bottom: The continuous-time infinite dynamic topic model (ciDTM) is a mixture of the oHDP and cDTM. It has a continuous-time domain like cDTM, and its number of topics evolve over time as in oHDP. It overcomes limitations of both models regarding evolution of the number of topics and time-step granularity. 3
- 1.2 Topic state-transition diagram: Oval shaped nodes represent states with their names typed inside of them. Arrows represent transitions from one state to another, or to itself. Each arrow has an **Event** / **Action** describing the event that triggered the transition, and the action taken by the system in response to that transition. The start state is represented by an arrow with no origin pointing to the state. 5
- 1.3 Latent Dirichlet allocation (LDA) graphical model. The rectangles, known as plates in this notation, represent replication. Plate with multiplicity D denotes a collection of documents. Each of these documents is made of a collection of words represented by plate with multiplicity N . Plate with multiplicity K represents a collection of topics. The shaded node W represents a word which is the only observed random variable in the model. The non-shaded nodes represent latent random variables in the model; α is the Dirichlet prior parameter for topic distribution per document. θ is a topic distribution for a document, while Z is the topic sampled from θ for word W . β is a Markov matrix giving the word distribution per topic, and η is the Dirichlet prior parameter used in generating that matrix. Shaded nodes are observed random variables, while non-shaded nodes are latent random variables. Nodes are labeled with the variable they represent. The rectangles are known as plates in this notation and they represent replication. For nested plates, an inner plate has multiplicity equal to the replication value shown on its lower-right corner times the multiplicity of its parent. 8

1.4	Visualization of a subset of 50 documents in two dimensions from the NIPS dataset. Each rectangle represents a document with its title inside of the rectangle. Distance between two documents is directly proportional to their similarity. The symmetrized Kullback-Leibler divergence is used to evaluate the distance between pairs of documents.	9
2.1	Variational transformation of the logarithmic function. The transformed function $(\lambda x - \log \lambda - 1)$ forms an upper bound for the original function. . . .	16
3.1	Topics over time graphical model. In this view, one timestamp t is sampled from a Beta distribution per word in a document. All words in a single document have the same timestamp. All the symbols in this figure follow the notation given in Figure 1.3 and ψ is the Beta distribution prior.	23
3.2	A graphical model representation of the infinite dynamic topic model using plate notation.	27
3.3	Graphical model representation of the continuous-time dynamic topic model using plate notation.	35
4.1	Continuous-time infinite dynamic topic model. None-shaded nodes represent unobserved latent variables, shaded nodes are observed variables, diamonds represent hyperparameters, and plates represent repetition. Symbols in this figure follow the notation in Figure 3.2 and H is the base distribution for the upper level DP, γ and α are the concentration parameters for the upper level, and lower level DPs respectively, and s_t is the timestamp for document at time t	41
4.2	A block diagram describing the workflow for the continuous-time infinite dynamic topic model working on a single document at a time.	42
5.1	Illustration of the birth/death/resurrection of five topics. Discontinuity of a topic indicates its death.	56
6.1	cDTM performance varies with different values for the number of topics using Reuters corpus. A value of 50 leads to best model fit among the six values tried. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.	59
6.2	oHDP per-word log-likelihood for different batch size values using Reuters corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.	60

6.3	ciDTM per-word log-likelihood for different batch size values using Reuters corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.	62
6.4	A comparison of ciDTM, oHDP and cDTM using their best setting values discovered earlier using Reuters corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.	64
6.5	Comparison of the per-word log-likelihood for cDTM, oHDP and ciDTM using 10,000 news stories from the Reuters21578 corpus [47]. Higher per-word log-likelihood indicate a better fit of the model to the data.	65
6.6	cDTM performance varies with different values for the number of topics using the BBC news corpus. A value of 200 leads to best model fit among the seven values tried. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.	66
6.7	oHDP per-word log-likelihood for different batch size values using the BBC news corpus. A value of 200 leads to best model fit among the seven values tried. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.	67
6.8	ciDTM per-word log-likelihood for different batch size values using the BBC news corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.	68
6.9	A comparison of ciDTM, oHDP and cDTM using their best setting values discovered earlier using the BBC news corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.	69
6.10	The wall clock running time in minutes for cDTM (10 topics), oHDP (batch size = 1) and ciDTM (batch size = 256) using subsets of the Reuters news corpus with different number of documents. These systems were run on a PC with 16GB of RAM and 3.4 GHz processor.	74

6.11	oHDP timeline construction for the Arab Spring topic over 61 documents covering this topic. The independent axis is the time, while the dependent axis is the topic index. It was found that topic 0 represents the Arab Spring events topic. A pink dot (●) represents the real topic that should be discovered, an black circle with a small dot (⊙) represents the topic/s that was discovered for the document. A black circle with a pink dot in the middle of it (⊗) is a real topic that was successfully discovered by the system.	76
6.12	ciDTM timeline construction for the Arab Spring topic over 61 documents covering this topic. The independent axis is the time, while the dependent axis is the topic index. It was found that topic 0 represents the Arab Spring events topic. A pink dot (●) represents the real topic that should be discovered, an black circle with a small dot (⊙) represents the topic/s that was discovered for the document. A black circle with a pink dot in the middle of it (⊗) is a real topic that was successfully discovered by the system.	77
C.1	Directed graphical model.	102
C.2	Undirected graphical model.	102
C.3	A skip-chain CRF model for an NER task.	105

Acknowledgments

I am greatly thankful to all those who helped me or gave me advice while working on this dissertation and the research that led to it.

Specifically, I would like to thank my advisor William Hsu for guiding and supporting me since I came to Kansas State University over five years ago. He gave me the right amount of freedom to explore different problems and different approaches to solve them. His guidance and feedback motivated and inspired me. I learned a lot from Bill throughout those years.

I wish to thank Tracey Hsu (Bill's wife) for proof reading and editing my dissertation. She is very attentive to details and my dissertation is in a better shape now thanks to her.

Dr Doina Cragea's advice helped me refine the work I presented in this dissertation. I would like to thank her for that and for the advice and guidance she gave me while working on an independent study class with her.

I was fortunate to have the help of undergraduate programmers in our group. Xinghuang Leon Hsu developed the crawler and parser used for crawling the BBC News corpus. He spent precious hours and days working on it making sure the corpus has what I need.

I enjoyed having good discussions with Surya Teja Kallumadi over the past few years. We discussed research and non-research problems. I want to thank him for hosting me in the summer of 2011 in Boston.

Special thanks to my beautiful wife Rachel who believed I will finish my PhD one day. She did not give up even though that day was a moving target. Always moving forward.

Finally, I am greatly indebted to my parents. Without their love and support throughout my life I could not have accomplished this.

Chapter 1

Introduction

1.1 Goal

1.1.1 Problem statement

In this thesis, I develop a continuous-time dynamic topic model. This model is an extension of the latent Dirichlet allocation (LDA) which is atemporal. I add two temporal components to: it i) the word distribution per topic, and ii) the number of topics. Both evolve in continuous time.

There exists similar temporal dynamic topic models. Ahmed and Xing [2] presented a model where the word distribution per topic and the number of topics evolve in discrete time. Bayesian inference using Markov chain Monte Carlo techniques in this model is feasible when the time step is big enough. It was efficiently applied to a system with 13 time steps [2]. Increasing the time granularity of the model dramatically increases the number of latent variables making inference prohibitively expensive. On the other hand, Wang et al. [72] presented a model where the word distribution per topic evolve in continuous time. They used variational Bayes methods for fast inference. A big limitation of their model is that it uses a predefined and fixed number of topics that does not evolve over time. In this thesis I develop and use a topic model that is a mixture of these two models where word distribution per topic and the number of topics evolve in continuous-time.

The need for the dynamic continuous-time topic model I am developing is evident in

the mass media business where news stories are published round-the-clock¹. To provide the news reader with a broad context and rich reading experience, many news outlets provide a list of related stories from news archives to the ones they present. As these archives are typically huge, manual search for these related stories is infeasible. Having the archived stories categorized into geographical areas or news topics such as politics and sports may help a little; a news editor can comb through a category looking for relevant stories. However, relevant stories may cross category boundaries. Keyword searching the entire archive may not be effective either; it returns stories based on keyword mentions not the topics the stories cover. A dynamic continuous-time topic model can be efficiently used to find relevant stories. It can track a topic over time even as a story develops and the word distribution associated with its topic evolves. It can detect the number of topics discussed in the news over time and fine-tune the model accordingly.

Goal: To develop a continuous-time topic model in which the word distribution per topic and the number of topics evolve in continuous time.

1.1.2 Central thesis

A continuous-time topic model with an evolving number of topics and a dynamic word distribution per topic (ciDTM) can be built using a Wiener process to model the dynamic topics in a hierarchical Dirichlet process [68]. This model cannot be efficiently emulated by the discrete-time infinite dynamic topic model when the topics it models evolve at different speeds. Using a fine time granularity to fit the fastest evolving topic would make inference in this system impractical. On the other hand, ciDTM cannot be emulated by a continuous-time dynamic topic model as it would require the use of a fixed number of topics which is a

¹Agence France-Presse (AFP) releases on average one news story every 20 seconds (5000 per day) while Reuters releases 800,000 English-language news stories annually. See: <http://www.afp.com/en/agency/afp-in-numbers> and <http://www.infotoday.com/it/apr01/news6.htm>

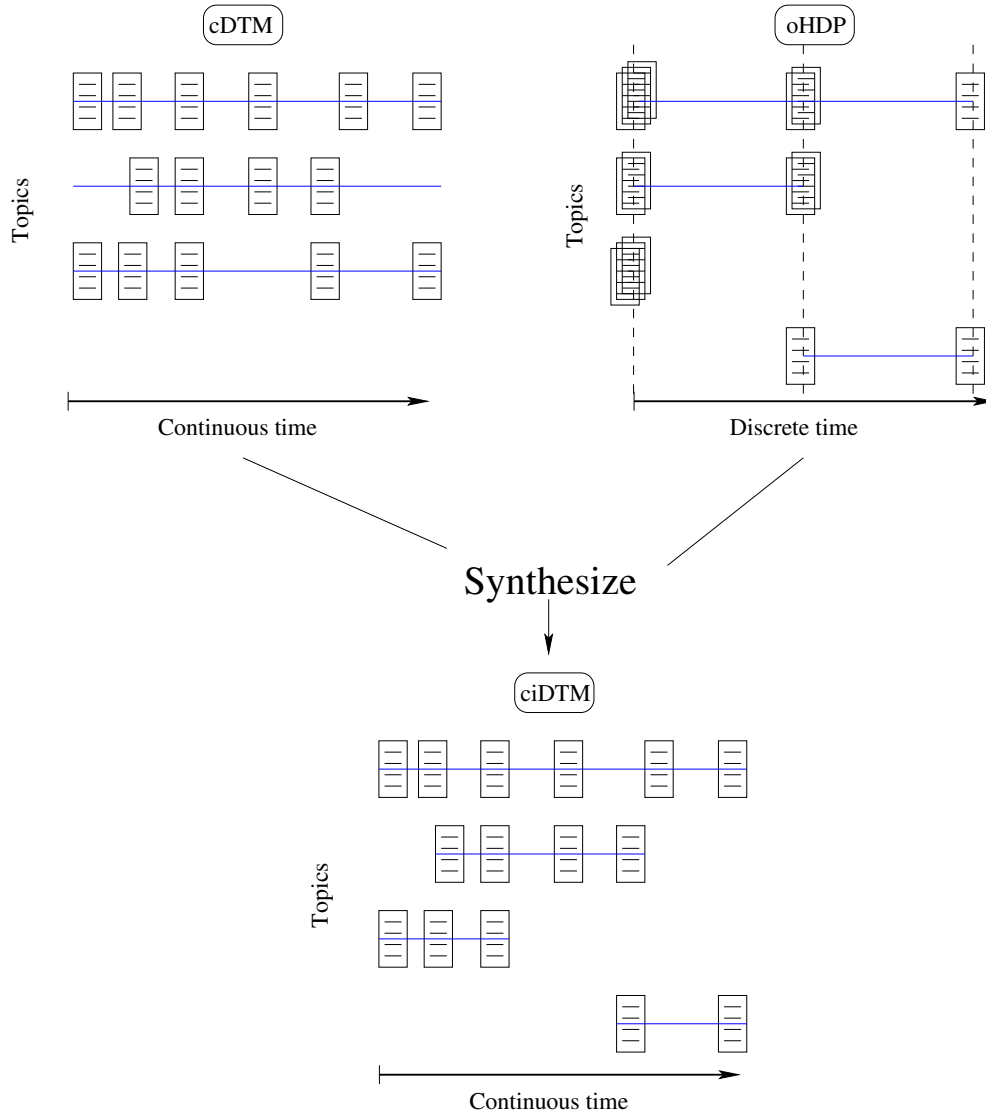


Figure 1.1: *Top left: the continuous-time dynamic topic model (cDTM) has a continuous-time domain; word and topic distributions evolve in continuous time. The number of topics in this model is fixed though. This may lead to having two separate topics being merged into one topic which was the case in the first topic from below. Top right: the online hierarchical Dirichlet process (oHDP) based topic model evolves the number of topics over time. The five documents belonging to the first and second topics from below were associated with their respective topics and were not merged into one topic which was the case with cDTM on the left. However, the model has a discrete-time domain which practically limits the degree of time granularity that we can use; very small time steps will make the model prohibitively expensive to inference. Bottom: The continuous-time infinite dynamic topic model (ciDTM) is a mixture of the oHDP and cDTM. It has a continuous-time domain like cDTM, and its number of topics evolve over time as in oHDP. It overcomes limitations of both models regarding evolution of the number of topics and time-step granularity.*

model parameter. Apart from the problem of finding an appropriate value for the parameter to start with, this value remains fixed over time leading to the potential problem of having multiple topics merge into one topic or having one topic split into multiple topics to keep the overall number of topics fixed.

For the problem of creating news timelines, ciDTM is more efficient and less computationally expensive than using a discrete-time unbounded-topic model [2] with a very small time unit (epoch), and more accurate than using a continuous-time bounded-topic model [72]. This is illustrated in Figure 1.1.

1.1.3 Technical objectives

A news timeline is characterized by a mixture of topics. When a news story arrives to the system, it should be placed on the appropriate timeline, if such one exists, or a new timeline will be created. A set is created of related stories that may not belong to the same timeline as the arriving story. In this process, care should be taken to avoid some pitfalls: If the dynamic number of topics generated by the model becomes too small, some news timelines may get conflated and distances between news stories may get distorted. If the dynamic number of topics becomes too large then number of variational parameters of the model will explode and the inference algorithm will become prohibitively expensive to run.

In a topic model system that receives text stream, a topic has two dormancy states: *Active* and *Dead*. Transition between these two states is illustrated in a state-transition diagram in Figure 1.2. When a new topic is born, it enters the **Active** state and a timer (**ActiveTimer**) starts. The timer is reset whenever a document relevant to the topic is received and processed by the system. When the timer expires, the topic transitions to the **Dead** state. The topic remains in this state as long as the system receives and processes non-relevant documents. If a relevant document is received and processed by the system, the topic transitions to the **Active** state, and the timer (**ActiveTimer**) is started.

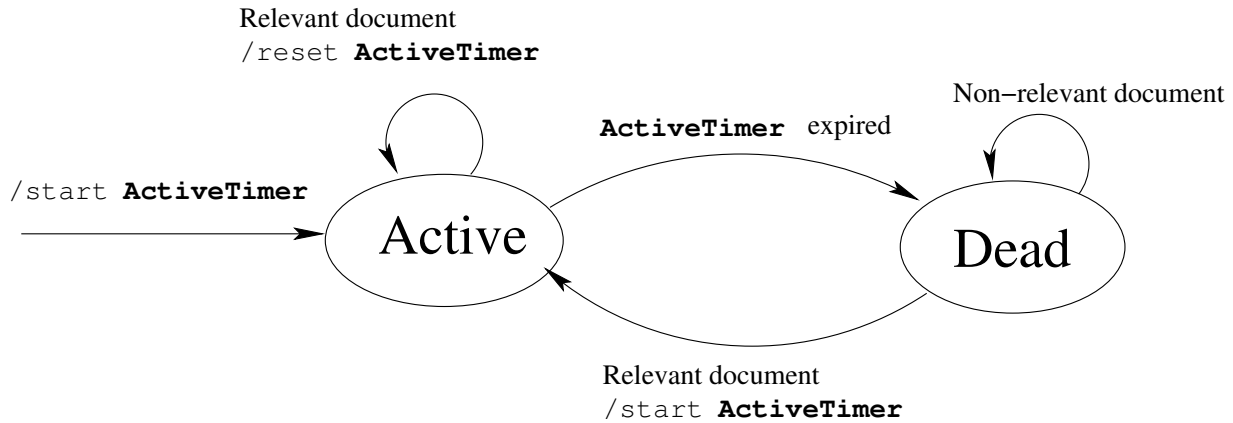


Figure 1.2: *Topic state-transition diagram: Oval shaped nodes represent states with their names typed inside of them. Arrows represent transitions from one state to another, or to itself. Each arrow has an **Event / Action** describing the event that triggered the transition, and the action taken by the system in response to that transition. The start state is represented by an arrow with no origin pointing to the state.*

1.2 Existing approaches: atemporal models

The volume of digitized knowledge has been increasing with an unprecedented rate recently. The past decade has witnessed the rise of projects like JSTOR [67], Google Books [21], the Internet Archive [32] and Europeana [31] that scan and index printed books, documents, newspapers, photos, paintings and maps. The volume of data that is born in digital format is even more rapidly increasing. Whether it is mainly unstructured user-generated content such as in social media, or generated by news portals.

New innovative ways were developed for categorizing, searching and presenting the digital material [17, 28, 48]. The usefulness of these services to the end user, which are free in most cases, is evident by their ever increasing utilization in learning, entertainment and decision-making.

A document collection can be indexed and keyword search can be used to search for a document in it. However, most keyword search algorithms are context-free and lack the ability to categorize documents based on their topic or find documents related to one of interest to us. The large volume of document collections precludes manual techniques of

annotation or categorization of the text.

The process of finding a set of documents related to a document at hand can be automated using topic models [8]. Most of these existing models are atemporal and perform badly in finding old relevant stories because the word collections that identifies the topics that these stories cover change over time and the model does not account for that change. Moreover, these models assume the number of topics is fixed over time, whereas in reality this number is constantly changing. Some famous examples of these models that were widely used in the past are latent semantic analysis (LSA) [46], probabilistic LSA (pLSA) [35] and latent Dirichlet allocation (LDA) [13].

1.2.1 Latent semantic analysis

Latent semantic analysis (LSA) is a deterministic technique for analyzing the relationship between documents or between words [46]. Like most topic models it treats a document as a bag of words; a document is a vector of term frequency ignoring term sequence. It is built on the idea that words that tend to appear in the same document together tend to carry similar meanings. In this document analysis technique, which is sometimes known as latent semantic indexing (LSI), the very large, noisy and sparse term-document frequency matrix gets its rank lowered using matrix factorization techniques such as singular value decomposition. In the lower-rank approximation matrix, each document is represented by a number of concepts in lieu of terms.

LSA is known for its inability to distinguish the different meanings of a polysemy word and place synonym words closer together in the semantic space; Different occurrences of a polysemous word could have different meanings based on its context. However, since LSA treats each word as a point in space, that point will be the semantic average of the different meanings of the word in the document. Different words with the same meaning (synonyms) represent another challenge to LSA in information retrieval applications. If a document is made out of the synonyms of terms found in another document, the correlation between

these two documents over their terms using LSA will not be as high as we would expect it to be.

1.2.2 Probabilistic latent semantic analysis

To overcome some of the LSA limitations, we can condition the occurrence of a word with a document on a latent topic variable. These will lead us to the probabilistic latent semantic analysis (pLSA). This model overcomes the discrepancy of the assumed joint Gaussian distribution between terms and documents in LSA and the observed Poisson distribution between them. However, pLSA is not a proper generative model for new documents, and the number of parameters in its model grows linearly with the number of documents. To overcome the generative process issue, a Dirichlet prior can be used for the document topic distribution, leading us to the latent Dirichlet allocation, which is the basis for topic models.

1.2.3 Latent Dirichlet allocation and topic models

Topic models are probabilistic models that capture the underlying semantic structure of a document collection based on a hierarchical Bayesian analysis of the original text [12, 13]. By discovering word patterns in the collection and establishing a pattern similarity measure, similar documents can be linked together and semantically categorized based on their topic.

A graphical model for the latent Dirichlet allocation is shown in Figure 1.3 [8]. The rectangles, known as plates in this notation, represent replication. Plate with multiplicity D denotes a collection of documents. Each of these documents is made of a collection of words represented by plate with multiplicity N . Plate with multiplicity K represents a collection of topics. The shaded node W represents a word which is the only observed random variable in the model. The non-shaded nodes represent latent random variables in the model; α is the Dirichlet prior parameter for topic distribution per document. θ is a topic distribution for a document, while Z is the topic sampled from θ for word W . β is a Markov matrix giving the word distribution per topic, and η is the Dirichlet prior parameter used in generating that matrix.

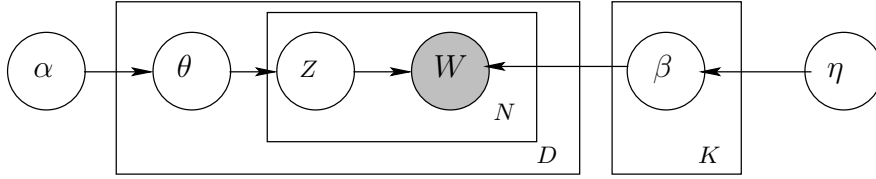


Figure 1.3: *Latent Dirichlet allocation (LDA) graphical model. The rectangles, known as plates in this notation, represent replication. Plate with multiplicity D denotes a collection of documents. Each of these documents is made of a collection of words represented by plate with multiplicity N . Plate with multiplicity K represents a collection of topics. The shaded node W represents a word which is the only observed random variable in the model. The non-shaded nodes represent latent random variables in the model; α is the Dirichlet prior parameter for topic distribution per document. θ is a topic distribution for a document, while Z is the topic sampled from θ for word W . β is a Markov matrix giving the word distribution per topic, and η is the Dirichlet prior parameter used in generating that matrix. Shaded nodes are observed random variables, while non-shaded nodes are latent random variables. Nodes are labeled with the variable they represent. The rectangles are known as plates in this notation and they represent replication. For nested plates, an inner plate has multiplicity equal to the replication value shown on its lower-right corner times the multiplicity of its parent.*

The reason for choosing the Dirichlet distribution to model the distribution of topics in a document and to model the distribution of words in a topic [15] is because the Dirichlet distribution is convenient distribution on the simplex [13], it is in the exponential family [70], has finite dimensional sufficient statistics [16], and is conjugate to the multinomial distribution [60]. These properties help developing an inference procedure and parameter estimation as pointed out by Blei et al. [13].

1.3 Significance

Even though news tracker and news aggregator systems have been used for a few years at a commercial scale for web news portals and news websites, most of them only provide relevant stories from the near past. It is understood that this is done to limit the number of relevant stories but this at the same time casts doubt over the performance of these systems when they try to dig for related stories that are a few years old and therefore have different word distributions for the same topic.

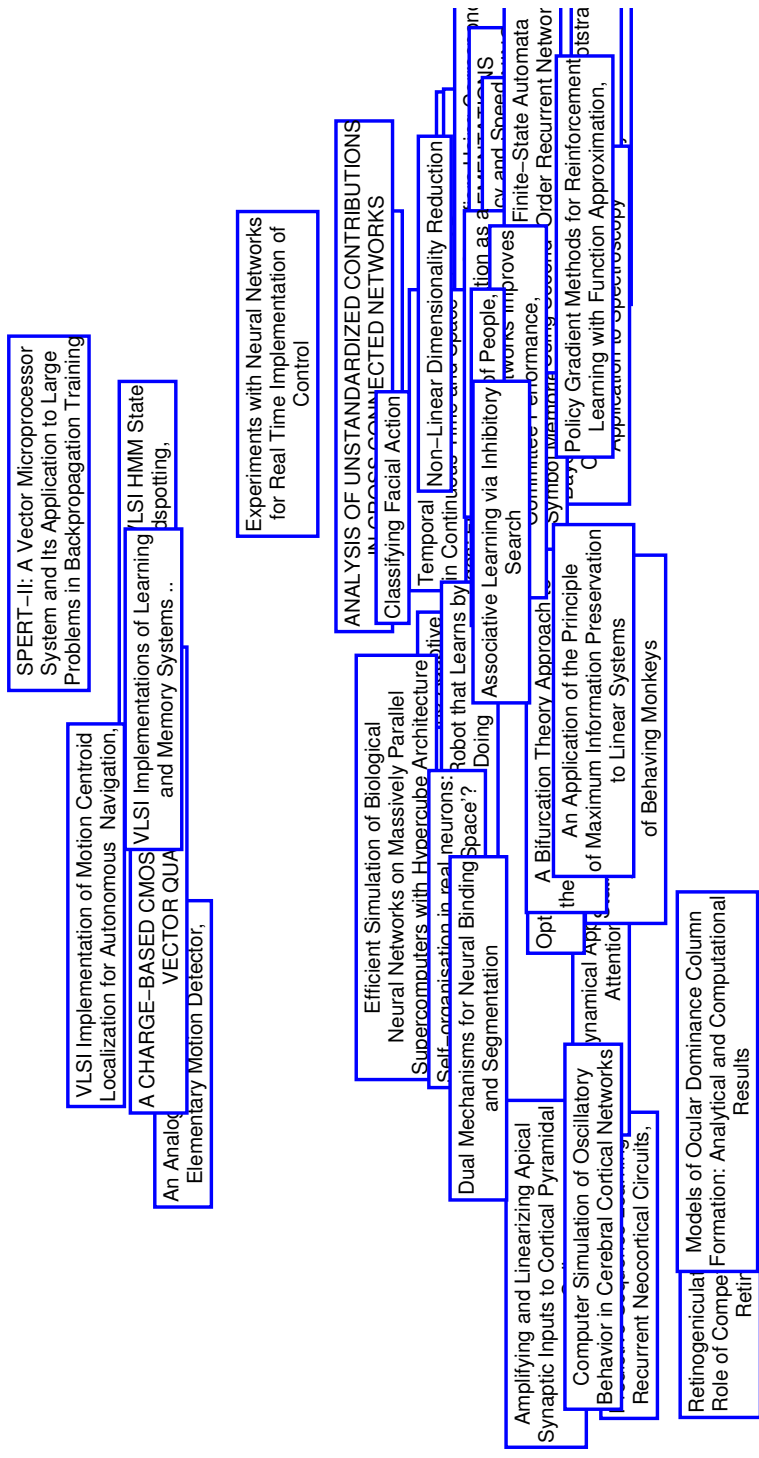


Figure 1.4: Visualization of a subset of 50 documents in two dimensions from the NIPS dataset. Each rectangle represents a document with its title inside of the rectangle. Distance between two documents is directly proportional to their similarity. The symmetrized Kullback-Leibler divergence is used to evaluate the distance between pairs of documents.

Topic models not only help us automate the process of text categorization and search, they enable us to analyze text in a way that cannot be done manually. Using topic models we can see how topics evolve over time [2], and how different topics are correlated with each other [10], and how this correlation changes over time. We can project the documents on the topic space and take a bird’s eye view to understand and analyze their distribution. We can zoom-in to analyze the main themes of a topic, or zoom-out to get a broader view of where this topic sits among other related topics. It should be noted that topic modeling algorithms do all that with no prior knowledge of the existence or the composition of any topic, and without text annotation.

The successful utilization of topic modeling in text encouraged researchers to explore other domains of applications for it. It has been used in software analysis [49] as it can be used to automatically measure, monitor and better understand software content, complexity and temporal evolution. Topic models were used to “improve the classification of protein-protein interactions by condensing lexical knowledge available in unannotated biomedical text into a semantically-informed kernel smoothing matrix” [57]. In the field of signal processing, topic models were used in audio scene understanding [39], where audio signals were assumed to contain latent topics that generate acoustic words describing an audio scene. It has been used in text summarization [22], building semantic question answering systems [18], stock market modeling [24] and music analysis [36].

Using variational methods for approximate Bayesian inference in the developed hierarchical Dirichlet allocation model for the dynamic continuous-time topic model will facilitate inference in models with a higher number of latent topic variables.

Other than the obvious application for the timeline creation system in retrieving a set of documents relevant to a document at hand, topic models can be used as a visualization technique. The user can view the timeline at different scale levels and understand how different events temporally unfolded.

Chapter 2

Bayesian models and inference algorithms

Evaluation of the posterior distribution $p(\mathbf{Z}|\mathbf{X})$ of the set of latent variables \mathbf{Z} given the observed data variable set \mathbf{X} is essential in topic modeling applications [6]. This evaluation is infeasible in real-life applications of practical interest due to the high number of latent variables we need to deal with [14]. For example, the time complexity of the junction tree algorithm is exponential in the size of the maximal clique in the junction tree. Expectation evaluation with respect to such a highly complex posterior distribution would be analytically intractable [70].

Just like the case with many engineering problems when finding an exact solution is not possible or too expensive to obtain, we resort to approximate methods. Even in some cases when the exact solution can be obtained, we might favor an approximate solution because the benefit of reaching the exact solution does not justify the extra cost spent to reach it. When the nodes or node clusters of the graphical model are almost conditionally independent, or when the node probabilities can be determined by a subset of its neighbors, an approximate solution will suffice for all practical purposes [55]. Approximate inference methods fall broadly under two categories, stochastic and deterministic [41].

Stochastic methods, such as Markov Chain Monte Carlo (MCMC) methods, can theoretically reach exact solution in limited time given unlimited computational power [29]. In

practice, the quality of the approximation obtained is tied to the available computational power. Even though these methods are easy to implement and therefore widely used, they are computationally expensive.

Deterministic approximation methods, like variational methods, make simplifying assumptions on the form of the posterior distribution or the way the nodes of the graphical model can be factorized [37]. These methods therefore could never reach an exact solution even with unlimited computational resources.

2.1 Introduction

As I stated earlier, the main problem in graphical model applications is finding an approximation for the posterior distribution $p(\mathbf{Z}|\mathbf{X})$ and the model evidence $p(\mathbf{X})$, where $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is the set of all observed variables and $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ is the set of all latent variables and model parameters. $p(\mathbf{X})$ can be decomposed using [6]:

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \text{KL}(q||p) \tag{2.1}$$

where

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \tag{2.2}$$

$$\text{KL}(q||p) = - \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \tag{2.3}$$

Our goal is to find a distribution $q(\mathcal{Z})$ that is as close as possible to $p(\mathbf{Z}|\mathbf{X})$. To do so, we need to minimize the Kullback-Leibler (KL) distance between them [43]. Minimizing this measure while keeping the left-hand side value fixed means maximizing the lower bound $\mathcal{L}(q)$ on the log marginal probability. The approximation in finding such a distribution arises from the set of restrictions we put on the family of distributions we pick q from. This family of distributions has to be rich enough to allow us to include a distribution that is

close enough to our target posterior distribution, yet the distributions have to be tractable [6].

This problem can be transformed into a non-linear optimization problem if we use a parametric distribution $p(\mathbf{Z}|\omega)$, where ω is its set of parameters. $\mathcal{L}(q)$ becomes a function of ω and the problem can be used using non-linear optimization methods such as Newton or quasi-Newton methods [63].

2.1.1 Factorized distributions

Instead of restricting the form of the family of distributions we want to pick $q(\mathbf{Z})$ from, we can make assumptions on the way it can be factored. We can make some independence assumptions [70]. Let us say that our set of latent variables \mathbf{Z} can be factorized according to [37]:

$$q(\mathbf{Z}) = \prod_{i=1}^M q_i(\mathbf{Z}_i) \quad (2.4)$$

This approximation method is known in the physics domain by *mean field theory* [4].

To maximize the lower bound $\mathcal{L}(q)$ we need to minimize each one of the factors $q_i \mathbf{Z}_i$.

We can do so by substituting (2.4) into (2.2) to get the following [6]:

$$\mathcal{L}(q) = \int \prod_i q_i \left\{ \ln p(\mathbf{X}, \mathbf{Z}) - \sum_i \ln q_i \right\} d\mathbf{Z} \quad (2.5)$$

$$= \int q_j \left\{ \int \ln p(\mathbf{X}, \mathbf{Q}) \prod_{i \neq j} q_i d\mathbf{Z}_i \right\} - \int q_j \ln q_j d\mathbf{Z}_j + c \quad (2.6)$$

$$= \int q_j \ln \tilde{p}(\mathbf{X}, \mathbf{Z}_j) d\mathbf{Z}_j - \int q_j \ln q_j d\mathbf{Z}_j + c \quad (2.7)$$

where

$$\ln \tilde{p}(\mathbf{X}, \mathbf{Z}_j) = \mathbb{E}_{i \neq j} [\ln p(\mathbf{X}, \mathbf{Q})] + c \quad (2.8)$$

and

$$\mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})] = \int \ln p(\mathbf{X}, \mathbf{Z}) \prod_{i \neq j} q_i d\mathbf{Z}_i \quad (2.9)$$

Where $\mathbb{E}_{i \neq j}$ is the expectation with respect to q over z_i such that $i \neq j$.

Since (2.7) is a negative KL divergence between $q_j(Z_j)$ and $\tilde{p}(\mathbf{X}, \mathbf{Z}_j)$, we can maximize $\mathcal{L}(q)$ with respect to $q_j(\mathbf{Z}_j)$ and obtain [6]:

$$\ln q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})] + c \quad (2.10)$$

To get rid of the constant we can take the exponential of both sides of this equation to get:

$$q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})])}{\int \exp(\mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Q})]) d\mathbf{Z}_j} \quad (2.11)$$

Example

For illustration, we are going to take a simple example for the use of factorized distribution in the variational approximation of parameters of a simple distribution [50]. Let us take the univariate Gaussian distribution over x . In this example, we will infer the posterior distribution of its mean (μ) and precision ($\tau = \sigma^{-2}$), given a data set $\mathcal{D} = \{x_1, \dots, x_N\}$ of i.i.d. observed values with a likelihood:

$$p(\mathcal{D}|\mu, \tau) = \left(\frac{\tau}{2\pi}\right)^{N/2} \exp\left\{-\frac{\tau}{2} \sum_{n=1}^N (x_n - \mu)^2\right\} \quad (2.12)$$

with a Gaussian and Gamma conjugate prior distributions for μ and τ as follows:

$$p(\mu|\tau) = \mathcal{N}(\mu|\mu_0, (\lambda_0\tau)^{-1}) \quad (2.13)$$

$$p(\tau) = \text{Gamma}(\tau|a_0, b_0) \quad (2.14)$$

By using factorized variational approximation, we can assume the posterior distribution factorizes according to:

$$q(\mu, \tau) = q_\mu(\mu)q_\tau(\tau) \quad (2.15)$$

We can evaluate the optimum for each of the factors using (2.12). The optimal factor for the mean is:

$$\ln q_\mu^*(\mu) = -\frac{\mathbb{E}_\tau}{2} \left\{ \lambda_0(\mu - \mu_0)^2 + \sum_{n=1}^N (x_n - \mu)^2 \right\} + c \quad (2.16)$$

Which is the Gaussian $\mathcal{N}(\mu|\mu_N, \lambda_N^{-1})$ with mean and precision given by:

$$\mu_N = \frac{\lambda_0\mu_0 + N\bar{x}}{\lambda_0 + N} \quad (2.17)$$

$$\lambda_N = (\lambda_0 + N)\mathbb{E}[\tau] \quad (2.18)$$

And the optimal factor for the precision is:

$$\ln q_\tau^*(\tau) = (a_0 - 1 + \frac{N}{2}) \ln \tau - b_0\tau - \frac{\tau}{2} \mathbb{E}_\mu \left[\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0(\mu - \mu_0)^2 \right] + c \quad (2.19)$$

Which is a gamma distribution $\text{Gamma}(\tau|a_N, b_N)$ with parameters:

$$a_N = a_0 + \frac{N}{2} \quad (2.20)$$

$$b_N = b_0 + \frac{1}{2} \mathbb{E}_\mu \left[\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0(\mu - \mu_0)^2 \right] \quad (2.21)$$

As we can see from (2.18) and (2.21), the optimal distributions for the mean and precision factors depend on expectations function of the other variable. Their values can be evaluated iteratively by first assuming an initial value, lets say for $\mathbb{E}[\tau]$ and use this value to evaluate $q_\mu(\mu)$. Given this value, we can use it to extract $\mathbb{E}[\mu]$ and $\mathbb{E}[\mu^2]$ and use them to evaluate $q_\tau(\tau)$. This value can in turn be used to extract a new revised value for \mathbb{E}_τ which can be used to update $q_\mu(\mu)$ and continue the cycle until their values converge to the optimum values.

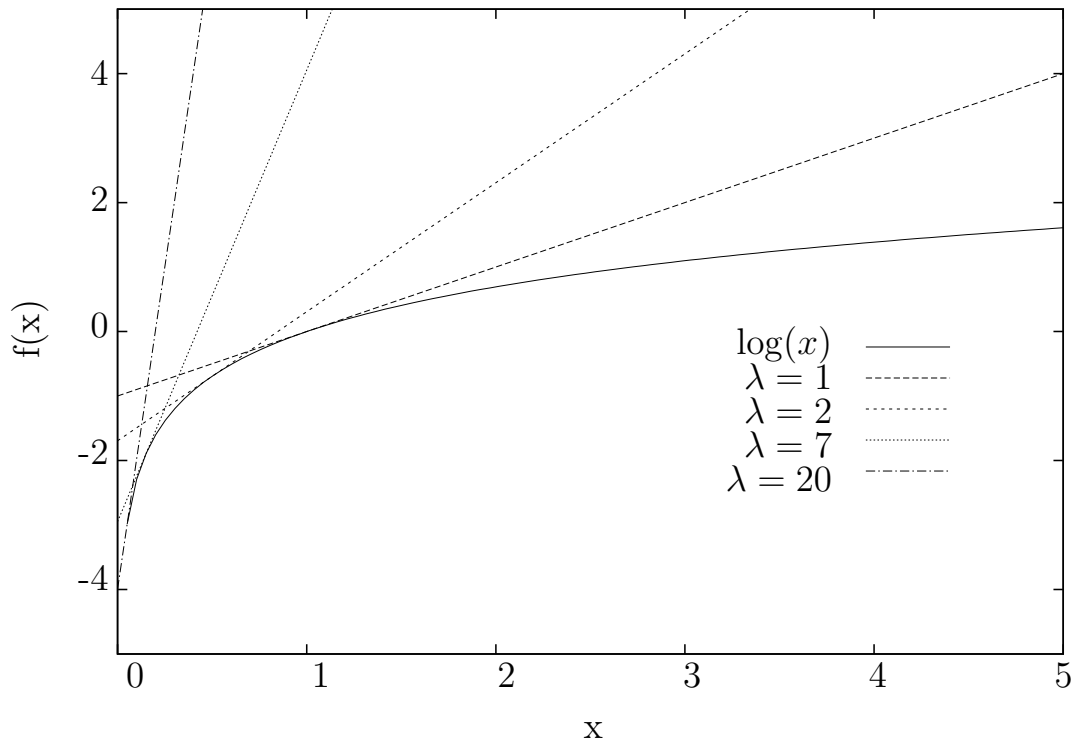


Figure 2.1: Variational transformation of the logarithmic function. The transformed function $(\lambda x - \log \lambda - 1)$ forms an upper bound for the original function.

2.1.2 Variational parameters

Variational methods transform a complex problem into a simpler form by decoupling the degrees of freedom of this problem by adding variational parameters. For example, we can transform the logarithm function as follows [37]:

$$\log(x) = \min_{\lambda} \{\lambda x - \log \lambda - 1\} \quad (2.22)$$

Here I introduced the variational parameter λ Which we are trying its value that minimizes the function.

As we can see in Figure 2.1, for different values of λ , there is a tangent line to the concave logarithmic function, and the set of lines formed by varying over the values of λ forms a family of upper bounds for the logarithmic function. Therefore,

$$\log(x) \leq \lambda x - \log \lambda - 1 \quad \forall \lambda \quad (2.23)$$

2.2 Variational inference

In this section, I use variational inference to find an approximation to the true posterior of the latent topic structure [76]; The topic distribution per word, the topic distribution per document, and the word distribution over topics.

I use variational Kalman filtering [38] in continuous time for this problem. The variational distribution over the latent variables can be factorized as follows:

$$\begin{aligned}
 q(\beta_{1:T}, z_{1:T}, 1:N, \theta_{1:T} | \hat{\beta}, \phi, \gamma) = & \\
 & \prod_{k=1}^K q(\beta_{1,k}, \dots, \beta_{T,k} | \hat{\beta}_{1,k}, \dots, \hat{\beta}_{T,k}) \times \\
 & \prod_{t=1}^T \left(q(\theta_t | \gamma_t) \prod_{n=1}^{N_t} q(z_{t,n} | \phi_{t,n}) \right)
 \end{aligned} \tag{2.24}$$

Where β is the word distribution over topics and $\beta_{1:T}, z_{1:T}, 1:N$ is the word distribution over topics for time $1 : T$, topic $z_{1:T}$ and word index $1 : N$, where N is the size of the dictionary.

In equation (2.24), γ_t is a Dirichlet parameter at time t for the multinomial per document topic distribution θ_t , and $\phi_{t,n}$ is a multinomial parameter at time t for word n for the topic $z_{t,n}$. $\{\hat{\beta}_{1,k}, \dots, \hat{\beta}_{T,K}\}$ are Gaussian variational observations for the Kalman filter [72].

In discrete-time topic models, a topic at time t is represented by a distribution over all terms in the dictionary including terms not observed at that time instance. This leads to high memory requirements specially when the time granularity gets finer. In my model, I use sparse variational inference [30] in which a topic at time t is represented by a multinomial distribution over terms observed at that time instance; variational observations are only made for observed words. The probability of the variational observation $\hat{\beta}_{t,w}$ given $\beta_{t,w}$ is Gaussian [72]:

$$p(\hat{\beta}_{t,w} | \beta_{t,w}) = \mathcal{N}(\beta_{t,w}, \hat{v}_t) \tag{2.25}$$

I use the forward-backward algorithm [59] for inference for the sparse variational Kalman filter. The variational forward distribution $p(\beta_{t,w}|\hat{\beta}_{1:t,w})$ is Gaussian [11]:

$$p(\beta_{t,w}|\hat{\beta}_{1:t,w}) = \mathcal{N}(m_{t,w}, V_{t,w}) \quad (2.26)$$

where

$$\begin{aligned} m_{t,w} &= \mathbb{E}(\beta_{t,w}|\hat{\beta}_{1:t,w}) \\ &= \left(\frac{\hat{v}_t}{V_{t-1,w} + v\Delta_{s_t} + \hat{v}_t} \right) m_{t-1,w} \\ &\quad + \left(1 - \frac{\hat{v}_t}{V_{t-1,w} + v\Delta_{s_t} + \hat{v}_t} \right) \hat{\beta}_{t,w} \end{aligned} \quad (2.27)$$

$$\begin{aligned} V_{t,w} &= \mathbb{E}((\beta_{t,w} - m_{t,w})^2|\hat{\beta}_{1:t,w}) \\ &= \left(\frac{\hat{v}_t}{V_{t-1,w} + v\Delta_{s_t} + \hat{v}_t} \right) (V_{t-1,w} + v\Delta_{s_t}) \end{aligned} \quad (2.28)$$

Similarly, the backward distribution $p(\beta_{t,w}|\hat{\beta}_{1:T,w})$ is Gaussian [11]:

$$p(\beta_{t,w}|\hat{\beta}_{1:T,w}) = \mathcal{N}(m_{t,w}, V_{t,w}) \quad (2.29)$$

where

$$\begin{aligned} \tilde{m}_{t-1} &= \mathbb{E}(\beta_{t-1}|\hat{\beta}_{1:T}) \\ &= \left(\frac{v\Delta_{s_t}}{V_{t-1,w} + v\Delta_{s_t}} \right) m_{t-1} \\ &\quad + \left(1 - \frac{v\Delta_{s_t}}{V_{t-1,w} + v\Delta_{s_t}} \right) \tilde{m}_{t,w} \end{aligned} \quad (2.30)$$

$$\begin{aligned} \tilde{V}_{t-1,w} &= \mathbb{E}((\beta_{t-1} - \tilde{m}_{t-1})^2|\hat{\beta}_{1:T}) \\ &= V_{t-1,w} + \left(\frac{V_{t-1}}{V_{t-1} + v\Delta_{s_t}} \right)^2 (\tilde{V}_t - V_{t-1} - v\Delta_{s_t}) \end{aligned} \quad (2.31)$$

The likelihood of the observations has a lower bound defined by:

$$\begin{aligned}
\mathcal{L}(\hat{\beta}) &\geq \sum_{t=1}^T \mathbb{E}_q \left[\log p(w_t|\beta_t) - \log p(\hat{\beta}_t|\beta_t) \right] \\
&\quad + \sum_{t=1}^T \log q(\hat{\beta}_t|\hat{\beta}_{1:t-1})
\end{aligned} \tag{2.32}$$

where

$$\begin{aligned}
\mathbb{E}_q \log p(w_t|\beta_t) &= \sum_w n_{t,w} \mathbb{E}_q \left(\beta_{t,w} - \log \sum_w \exp(\beta_{t,w}) \right) \\
&\geq \sum_w n_{t,w} \tilde{m}_{t,w} \\
&\quad - n_t \log \sum_w \exp(\tilde{m}_{t,w} + \tilde{V}_{t,w}/2)
\end{aligned} \tag{2.33}$$

$$\mathbb{E}_q \log p(\hat{\beta}_t|\beta_t) = \sum_w \delta_{t,w} \mathbb{E}_q \log q(\hat{\beta}_{t,w}|\beta_{t,w}) \tag{2.34}$$

$$\log q(\hat{\beta}_t|\hat{\beta}_{1:t-1}) = \sum_w \delta_{t,w} \log q(\hat{\beta}_{t,w}|\hat{\beta}_{1:t-1,w}) \tag{2.35}$$

where $\delta_{t,w}$ is the Dirac delta function [23] and it is equal to 1 iff $\hat{\beta}_{t,w}$ is in the variational observations. $n_{t,w}$ is the number of words in document d_t , and $n_t = \sum_w n_{t,w}$.

Chapter 3

Existing solutions and their limitations

Several studies have been done to account for the changing latent variables of the topic model. Xing [77] presented a dynamic logistic-normal-multinomial and logistic-normal-Poisson models that he used later as building blocks for his models. Wang and McCallum [74] presented a non-Markovian continuous-time topic model in which each topic is associated with a continuous distribution over timestamps. Blei and Lafferty [11] proposed a dynamic topic model in which the topic's word distribution and popularity are linked over time, though the number of topics was fixed. This work was picked up by other researchers who extended this model. In the following I describe some of these extended models.

3.1 Temporal topic models

Traditional topic models which are manifestations of graphical models model the occurrence and co-occurrence of words in documents disregarding the fact that many document collections cover a long period of time. Over this long period of time, topics which are distributions over words could change drastically. The word distribution for a topic covering a branch of medicine is a good example of a topic that is dynamic and evolves quickly over time. The terms used in medical journals and publications change over time as the field develops. Learning the word distribution for such a topic using a collection of old medical documents

would not be good in classifying new medical documents as the new documents are written using more modern terms that reflect recent medical research directions and discoveries that keep advancing every day. Using a fixed word distribution for such a topic, and for many other topics that usually evolve in time, would result in wrong document classification and wrong topic inference. The error could become greater over the course of time as the topic evolves more and more and the distance between the original word distribution that was learned using the old collection of documents and the word distribution of the same topic in a more recent document collection for the same field becomes greater. Therefore, there is a strong need to add a temporal model to such topic models to reflect the changing word distribution for topics in a collection of documents to improve topic inference in a dynamic collection of documents.

3.2 Topics over time

Several topic models were suggested that add a temporal component to the model. I will refer to them in this dissertation as temporal topic models. These temporal topic models include the Topics over Time (TOT) topic model [74]. This model directly observes document timestamp. In its generative process, the model generates word collection and a timestamp for each document. In their original paper, Wang and McCallum gave two alternative views for the model. The first one has a generative process in which for each document a multinomial distribution θ is sampled from a Dirichlet prior α . And then from that multinomial θ one timestamp t is sampled for that document from a Beta distribution, and one topic z is sampled for each word w in the document. Another multinomial β is sampled for each topic in the document from a Dirichlet prior η , and a word w is sampled from that multinomial β given the topic z sampled for this document. This process which seems natural to document collections in which each document has one timestamp contrasts the other view which the authors presented and is given in Figure 3.1. In this view which the authors adopted in their model, the generative process is similar to the generative process

presented earlier for the first view, but instead of sampling one timestamp t from a Beta distribution for each document, one timestamp is sampled from a Beta distribution for each word w in the document. All words in the same document however have the same timestamp. The authors claim that this view makes it easier to implement and understand the model.

It is to be noted that the word distribution per topic in this TOT model is fixed over time though. “TOT captures changes in the occurrence (and co-occurrence conditioned on time) of the topics themselves, not changes of the word distribution of each topic.” [74] The authors argue that evolution in topics happens by the changing occurrence and co-occurrence of topics as two co-occurring topics would be equivalent to a new topic that is formed by merging both topics, and losing the co-occurrence would be equivalent to splitting that topic into two topics.

In this TOT model, topic co-occurrences happen in continuous-time and the timestamps are sampled from a Beta distribution. A temporal topic model evolving in continuous-time has a big advantage over a discrete-time topic model. Discretization of time usually comes with the problem of selecting a good timestep. Picking a large timestep leads to the problem of having documents covering a large time period over which the word distributions for the topics covered in these documents evolved significantly used in learning these distributions, or are inferred using one fixed distribution over that period of time. Picking a small timestep would complicate inference and learning as the number of parameters would explode as the timestep granularity increases. Another problem that arises with discretization is that it does not account for varying time granularity over time. Since topics evolve at different paces, and even one topic may evolve with different speeds over time, having a small timestep at one point in time to capture the fast dynamics of an evolving topic may be unnecessary later in time when the topic becomes stagnant and does not evolve as quickly. Keeping a fine grained timestep at that point will make inference and learning slower as it increases the number of model parameters. On the other hand, having a coarse timestep at

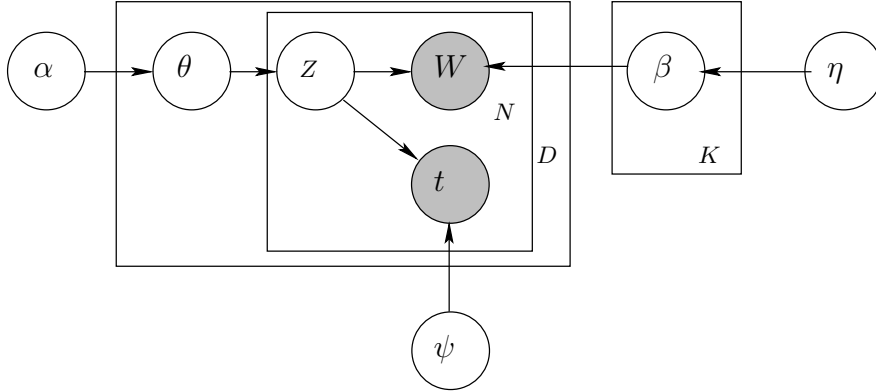


Figure 3.1: *Topics over time graphical model. In this view, one timestamp t is sampled from a Beta distribution per word in a document. All words in a single document have the same timestamp. All the symbols in this figure follow the notation given in Figure 1.3 and ψ is the Beta distribution prior.*

one point in time when a topic does not evolve quickly may be suitable for that time but would become too big of a timestep when the topic starts evolving faster in time, and documents that fall within one timestep would be inferred using the fixed word distributions for the topics not reflecting the change that happened to these topics. To take this case to the extreme, a very large timestep covering the time period over which the document collection exists would be equivalent to using a classical latent Dirichlet allocation model that has no notion of time at all.

It is to be noted that the TOT topic model uses a fixed number of topics. This limitation has major implications because not only do topics evolve over time, but topics get born, die, and reborn over time. The number of active topics over time should not be assumed to be fixed. Assuming a fixed number could lead to having topics conflated or merged in a wrong way. Assuming a number of topics that is greater than the actual number of topics in document collection at a certain point in time causes the actual topics to be split over more than one topic. In an application that classifies news articles based on the topics they discuss, this will cause extra classes to be created and makes the reader distracted between two classes that cover the same topic. On the other hand, having a number of topics that is smaller than the actual number of topics covered by a collection of documents

makes the model merge different topics into one. In the same application of new articles classification, this leads to having articles covering different topics appearing under the same class. Article classes could become very broad and this is usually undesirable as the reader relies on classification to read articles based on his/her focused interest.

In the TOT model, exact inference cannot be done. Wang and McCallum [74] resorted to Gibbs sampling in this model for approximate inference. Since a Dirichlet which is a conjugate prior for the multinomial distribution is used, the multinomials θ and ϕ can be integrated out in this model. Therefore, we do not have to sample θ and ϕ . This makes the model more simple, faster to simulate, faster to implement, and less prone to errors. The authors claim that because they use a continuous Beta distribution rather than discretizing time, sparsity would not be a big concern in fitting the temporal part of the model. In the training phase or learning the parameters of the model, every document has a single timestamp, and this timestamp is associated with every word in the document. Therefore, all the words in a single document have the same timestamp. Which is what we would naturally expect. However, in the generative graphical model presented in Figure 3.1 for the topics over time topic model, one timestamp is generated for each word in the same document. This would probably result in different words appearing in the same document having different timestamp. This is typically something we would not expect, because naturally, all words in a single document have the same timestamp because they were all authored and released or published under one title as a single publication unit or publication entity. In this sense, the generative model presented in Figure 3.1 is deficient as it assigns different timestamps to words within the same document. The authors of the paper that this model was presented in argue that this deficiency does not distract a lot from this model and it still remains a powerful in modeling large dynamic text collections.

An alternative generative process for the topics over time was also presented by the authors of this model. In this alternative process, one timestamp is generated for each document using rejection sampling or importance sampling from a mixture of per topic

Beta distributions over time with mixture weight as the per document θ_d over topics.

Instead of jointly modeling co-occurrence of words and timestamps in document collections, other topic models relied on analyzing how topics change over time by dividing the time covered by the documents into regions for analysis or by discretizing time. Griffiths and Steyvers [33] used atemporal topic model to infer the topic mixtures of the proceedings of the National Academy of Sciences (PNAS). They then ordered the documents in time based on their timestamps and assigned them to different time regions and analyzed their topic mixtures over time. This study does not infer or learn the timestamps of documents and merely analyzes the topics learned using a simple latent Dirichlet allocation model [13]. Instead of learning one topic model for the entire document collection and then analyzing the topic structure of the documents in the collection, Wang et al. [75] first divided the documents into consecutive time regions based on their timestamps. They then trained a different topic model for each region and analyzed how topics changed over time. This model has several limitations: First, the alignment of topics from one time region to the next is hard to do and would probably be done by hand, which is hard to do even with relatively small number of topics. Second, the number of topics was held constant throughout time even at times when the documents become rich in context and they naturally contain more topics, or at times when the documents are not as rich and contain relatively less number of topics. The model does not account for topics dying out and others being born. Third, finding the correct time segment and number of segments is hard to do as it typically involves manual inspection of the documents. The model however benefited from the fact that different models for documents in adjacent time regions are similar and the Gibbs sampling parameters learned for one region could be used as a starting point for learning parameters for the next time region [64].

In their TimeMines system, Swan and Jensen [66] generated a topic model that assigns one topic per document for a collection of news stories used to construct timelines in a topic detection and tracking task.

The topics over time topic model is a temporal model but not a Markovian one. It does not make the assumption that a topic state at time $t + 1$ is independent on all previous states of this topic except for its state at time t . Sarkar and Moore [61] analyze the dynamic social network of friends as it evolves over time using a Markovian assumption. Nodelman et al. [56] developed a Continuous-time Bayesian network (CTBN) that does not rely on time discretization. In their model, a Bayesian network evolves based on a continuous-time transition model using a Markovian assumption. Kleinberg [40] created a model that relies on the relative order of documents in time instead of using timestamps. This relative ordering may simplify the model and may be suitable for when the documents are released on a fixed or near fixed time interval but would not take into account the possibility that in some other applications like in news streams, the pace at which new stories are released and published varies over time.

3.3 Discrete-time infinite dynamic topic model

Ahmed and Xing [2] proposed a solution that overcomes the problem of having a fixed number of topics. They proposed an infinite Dynamic Topic Model (iDTM) that allows for an unbounded number of topics and an evolving representation of topics according to a Markovian dynamics. They analyzed the birth and evolution of topics in the NIPS community based on conference proceedings. Their model evolved topics over discrete time units called epochs. All proceedings of a conference meeting fall into the same epoch. This model does not suit many applications as news articles production and tweet generation is more spread over time and does not usually come out in bursts.

In many topic modeling applications, such as for discussion forums, news feeds and tweets, the time duration of an epoch may not be clear. Choosing too coarse a resolution may render invalid the assumption that documents within the same epoch are exchangeable. Different topics and storylines will get conflated, and unrelated documents will have similar topic distribution. If the resolution is chosen to be too fine, then the number of variational

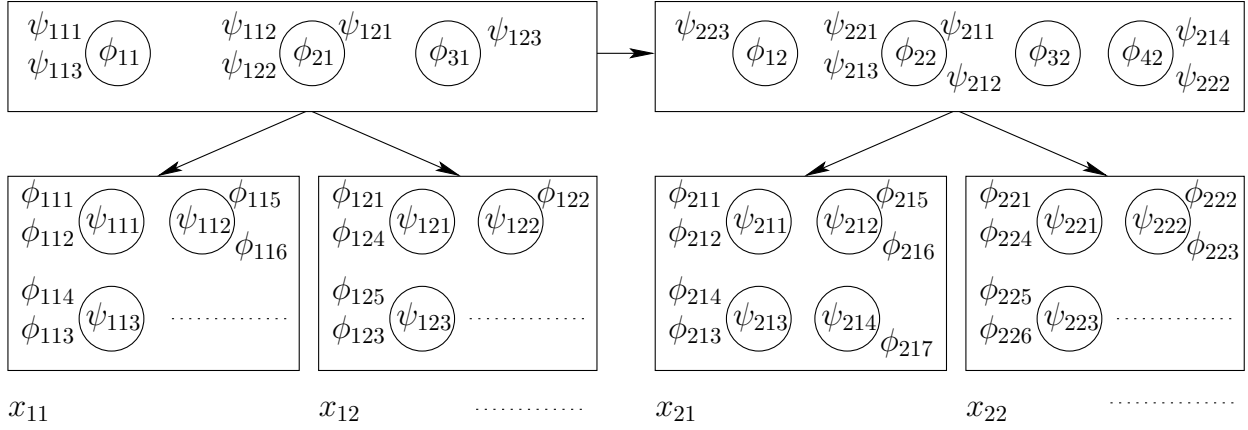


Figure 3.2: A graphical model representation of the infinite dynamic topic model using plate notation.

parameters of the model will explode with the number of data points. The inference algorithm will become prohibitively expensive to run. Using a discrete time dynamic topic model could be valid based on assumptions about the data. In many cases, the continuity of the data which has an arbitrary time granularity prevents us from using a discrete time model.

To summarize: in streaming text topic modeling applications, the discrete-time model given above is brittle. An extension to continuous-time will give it the needed flexibility to account for change in temporal granularity.

Statistical model

Figure 3.2 shows a graphical model representation for an order one recurrent Chinese restaurant franchise process (RCRF). The symbols in this figure follow the same notation used for LDA in Figure 1.3 In RCRF, documents in each epoch are modeled using a Chinese restaurant franchise process (CRFP). The menus that the documents were sampled from at different timesteps are tied over time.

For a given Dirichlet Process (DP), $DP(G_0, \alpha)$, with a base measure G_0 , and a concentration parameter α , $G \sim DP(G_0, \alpha)$ is a Dirichlet distribution over the parameter space θ . By integrating out G , θ follows a Polya urn distribution [7] or a recurrent Chinese restaurant

process [2]

$$\theta_i | \theta_{1:i-1}, G_0, \alpha \sim \sum_k \frac{m_k}{i-1+\alpha} \delta(\phi_k) + \frac{\alpha}{i-1+\alpha} G_0 \quad (3.1)$$

where, $\phi_{1:k}$ are the distinct values of θ , and m_k is the number of parameter θ with ϕ_k value. A Dirichlet process mixture model (DPM) can be built using the given DP on top of a hierarchical Bayesian model.

One disadvantage of using RCRP is that in it each document is generated using a single topic. This assumption is unrealistic specially in some domains like in modeling news stories where each document is typically a mixture of different topics. To overcome this limitation, I can use a hierarchical Dirichlet process (HDP) in which each document can be generated from multiple topics.

To add temporal dependence in our model, we can use the temporal Dirichlet process mixture model (TDPM) proposed in [1] which allows unlimited number of mixture components. In this model, G evolves as follows [2]:

$$G_t | \phi_{1:k}, G_0, \alpha \sim DP(\zeta, D) \quad (3.2)$$

$$\zeta = \alpha + \sum_k m'_{kt} \quad (3.3)$$

$$D = \sum_k \frac{m'_{kt}}{\sum_l m'_{lt} + \alpha} \delta(\phi_k) + \frac{\alpha}{\sum_l m'_{lt} + \alpha} G_0 \quad (3.4)$$

$$m'_{kt} = \sum_{\delta=1}^{\Delta} \exp^{-\delta/\lambda} m_{k,t-\delta} \quad (3.5)$$

where m'_{kt} is the prior weight of component k at time t , Δ , λ are the width and decay factor of the time decaying kernel. For $\Delta = 0$, this TDPM will represent a set of independent DPMs at each time step, and a global DPM when $\Delta = T$ and $\lambda = \infty$. As we did with the time-independent DPM, we can integrate out $G_{1:T}$ from our model to get a set of parameters $\theta_{1:t}$ that follows a Poly-urn distribution:

$$\theta_{ti} | \theta_{t-1:t-\Delta}, \theta_{t,1:i-1}, G_0, \alpha \propto \sum_k (m'_{kt} + m_{kt}) \delta(\phi_{kt}) + \alpha G_0 \quad (3.6)$$

In this process, topic popularity at epoch t depends on its use at this epoch m_{kt} , and its use in the previous Δ epochs, m'_{kt} . Therefore, after Δ epochs of not being used, a topic can be considered dead. This makes sense as higher-order models will require more epochs to pass by without using a topic before it is considered dead. This is because the length of the chain is longer and the effect of the global menu of topics carries along for more epochs. On the other hand, in lower-order models the effect of the global menu of topics can only be passed along for less epochs than in the previous case.

By placing this model on top of a hierarchical Dirichlet process as indicated earlier, we can tie all random measures G_d , from which the multinomially distributed parameters $\theta_{\mathbf{d}}$ are drawn for each $\mathbf{w}_{\mathbf{d}}$ document in a topic model scheme by modeling G_0 as a random measure sampled from $DP(\gamma, H)$. By integrating G_d out of this model, we get the Chinese restaurant franchise process (CRFP) [11];

$$\theta_{di}|\theta_{d,1:i-1}, \alpha, \boldsymbol{\psi} \sim \sum_{b=1}^{b=B_d} \frac{n_{db}}{i-1+\alpha} \delta_{\psi_{db}} + \frac{\alpha}{i-1+\alpha} \delta_{\psi_{db}^{new}} \quad (3.7)$$

$$\psi_{db}^{new}|\boldsymbol{\psi}, \gamma \sim \sum_{k=1}^K \frac{m_k}{\sum_{l=1}^K m_l + \gamma} \delta_{\phi_k} + \frac{\gamma}{\sum_{l=1}^K m_l + \gamma} H \quad (3.8)$$

where ψ_{db} is topic b for document d , n_{db} is the number of words sampled from it, ψ_{db}^{new} is a new topic, B_d is the number of topics in document d , and m_k is the number of documents sharing topic ϕ_k .

We can make word distributions and topic trends evolve over time if we tie together all hyper-parameter base measures G_0 through time. The model will now take the following form:

$$\theta_{tdi}|\theta_{td,1:i-1}, \alpha, \boldsymbol{\psi}_{t-\Delta:t} \sim \sum_{b=1}^{b=B_d} \frac{n_{tdb}}{i-1+\alpha} \delta_{\psi_{tdb}} + \frac{\alpha}{i-1+\alpha} \delta_{\psi_{tdb}^{new}} \quad (3.9)$$

$$\begin{aligned}
\psi_{tdb^{new}}|\psi, \gamma &\sim \sum_{k:m_{kt}>0} \frac{m_{kt} + m'_{kt}}{\sum_{l=1}^{K_t} m_{lt} + m'_{lt} + \gamma} \delta_{\phi_{kt}} \\
&+ \sum_{k:m_{kt}=0} \frac{m_{kt} + m'_{kt}}{\sum_{l=1}^{K_t} m_{lt} + m'_{lt} + \gamma} \mathbb{P}(\cdot|\phi_{k,t-1}) \\
&+ \frac{\gamma}{\sum_{l=1}^{K_t} m_{lt} + m'_{lt} + \gamma} H
\end{aligned} \tag{3.10}$$

where ϕ_{kt} evolves using a random walk kernel like in [11]:

$$H = N(0, \sigma I) \tag{3.11}$$

$$\phi_{k,t}|\phi_{k,t-1} \sim N(\phi_{k,t-1}, \rho I) \tag{3.12}$$

$$w_{tdi}|\phi_{kt} \sim \mathcal{M}(L(\phi_{kt})) \tag{3.13}$$

$$L(\phi_{kt}) = \frac{\exp(\phi_{kt})}{\sum_{w=1}^W \exp(\phi_{ktw})} \tag{3.14}$$

We can see from (3.9), (3.10), (3.11) the non-conjugacy between the base measure and the likelihood.

Symbol	Definition
$DP(\cdot)$	Dirichlet process
G_0	base measure
α	concentration parameter
θ	Dirichlet distribution parameter space
$\phi_{1:k}$	Distinct values of θ
m_k	number of parameter θ with ϕ_k value
m'_{kt}	prior weight of component k at time t
Δ	width of time decaying kernel
λ	decay factor of a time decaying kernel
w	word in a document
H	base measure of the DP generating G_0
γ	concentration parameter for the DP generating G_0
ψ_{tdb}	topic b for document d at time t
n_{tdb}	number of words sampled from ψ_{tdb}
$\psi_{tdb^{new}}$	a new topic

Table 3.1: Table of notations for *iDTM*

The model given above is suitable for discrete time topic models with evolving number of topics. The topic trends and the word distribution of topics will evolve in discrete time though.

Evolving topics in continuous time can be achieved by using a Brownian motion model [54] that models the natural parameters of a multinomial distribution for the words over topics. A Dirichlet distribution can be used to model the natural parameters of multinomial distribution of the topics given the words of the parameter.

3.4 Online hierarchical Dirichlet processes

Traditional variational inference algorithms are suitable for some applications in which the document collection to be modeled is known before model learning or posterior inference takes place. If the document collection changes, however, the entire posterior inference procedure has to be repeated to update the learned model. This clearly incurs the additional cost of relearning and re-analyzing a potentially huge volume of information specially as the collection grows over time. This cost could become very high and a compromise should be made if this traditional variational inference algorithm is to be used about having an up-to-date model against saving computational power.

Online inference algorithms do not require several passes over the entire dataset to update the model which is a requirement for traditional inference algorithms. Sato [62] introduced an online variational Bayesian inference algorithm that gives variational inference algorithms an extra edge over their MCMC counterparts.

Traditional variational inference algorithms approximate the true posterior over the latent variables by suggesting a simpler distribution that gets refined to minimize its Kullback-Leibler (KL) distance to the true posterior. In online variational inference, this optimization is done using stochastic approximation [62, 73].

Statistical model

At the top level of a two level hierarchical Dirichlet process (HDP) a Dirichlet distribution G_0 is sampled from a Dirichlet process (DP). This distribution G_0 is used as the base measure for another DP at the lower level from which another Dirichlet distribution G_j is drawn.

This means that all the distributions G_j share the same set of atoms they inherited from their parent with different atom weights. Formally put:

$$G_0 \sim DP(\gamma, H) \tag{3.15}$$

$$G_j \sim DP(\alpha_0, G_0) \tag{3.16}$$

Where γ and H are the concentration parameter and base measure for the first level DP, α_0 and G_0 are the concentration parameter and base measure for the second level DP, and G_j is the Dirichlet distribution sampled from the second level DP.

In a topic model utilizing this HDP structure, a document is made of a collection of words, and each topic is a distribution over the words in the document collection. The atoms of the top level DP are the global set of topics. Since the base measure of the second level DP is sampled from the first level DP, then the sets of topics in the second level DP are subsets of the global set of topics in the first level DP. This ensures that the documents sampled from the second level process share the same set of topics in the upper level. For each document j in the collection, a Dirichlet G_j is sampled from the second level process. Then, for each word in the document a topic is sampled then a word is generated from that topic.

In Bayesian non-parametric models, variational methods are usually represented using a stick-breaking construction. This representation has its own set of latent variables on which an approximate posterior is given [9, 44, 69]. The stick-breaking representation used for this HDP is given at two levels: corpus-level draw for the Dirichlet G_0 from the top-level DP, and a document-level draw for the Dirichlet G_j from the lower-level DP. The corpus-level sample can be obtained as follows:

$$\beta'_k \sim \text{Beta}(1, \gamma) \quad (3.17)$$

$$\beta_k = \beta'_k \prod_{l=1}^{k-1} (1 - \beta'_l) \quad (3.18)$$

$$\phi_k \sim H \quad (3.19)$$

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k} \quad (3.20)$$

where γ is a parameter for the Beta distribution, β_k is the weight for topic k , ϕ_k is atom (topic) k , H is the base distribution for the top level DP, and δ is the Dirac delta function.

The second level (document-level) draws for the Dirichlet G_j are done by applying Sethuraman stick-breaking construction of the DP again as follows:

$$\psi_{jt} \sim G_0 \quad (3.21)$$

$$\pi'_{jt} \sim \text{Beta}(1, \alpha_0) \quad (3.22)$$

$$\pi_{jt} = \pi'_{jt} \prod_{l=1}^{t-1} (1 - \pi'_{jl}) \quad (3.23)$$

$$G_j = \sum_{t=1}^{\infty} \pi_{jt} \delta_{\psi_{jt}} \quad (3.24)$$

where ψ_{jt} is a document-level atom (topic) and π_{jt} is the weight associated with it.

This model can be simplified by introducing indicator variables \mathbf{c}_j that are drawn from the wights $\boldsymbol{\beta}$ [73]

$$c_{jt} \sim \text{Mult}(\boldsymbol{\beta}) \quad (3.25)$$

The variational distribution is thus given by:

$$q(\boldsymbol{\beta}', \boldsymbol{\pi}', \mathbf{c}, \mathbf{z}, \boldsymbol{\phi}) = q(\boldsymbol{\beta}')q(\boldsymbol{\pi}')q(\mathbf{c})q(\mathbf{z})q(\boldsymbol{\phi}) \quad (3.26)$$

$$q(\boldsymbol{\beta}') = \prod_{k=1}^{K-1} q(\beta'_k | u_k, v_k) \quad (3.27)$$

$$q(\boldsymbol{\pi}') = \prod_j \prod_{t=1}^{T-1} q(\pi'_{jt} | a_{jt}, b_{jt}) \quad (3.28)$$

$$q(\mathbf{c}) = \prod_j \prod_t q(c_{jt} | \varphi_{jt}) \quad (3.29)$$

$$q(\mathbf{z}) = \prod_j \prod_n q(z_{jn} | \zeta_{jn}) \quad (3.30)$$

$$q(\boldsymbol{\phi}) = \prod_k q(\phi_k | \lambda_k) \quad (3.31)$$

where $\boldsymbol{\beta}'$ is the corpus-level stick proportions and (u_k, v_k) are parameters for its beta distribution, $\boldsymbol{\pi}'_j$ is the document-level stick proportions and (a_{jt}, b_{jt}) are parameters for its beta distribution, \mathbf{c}_j is the vector of indicators, $\boldsymbol{\phi}$ is the topic distributions, and \mathbf{z} is the topic indices vector. In this setting, the variational parameters are φ_{jt} , ζ_{jn} , and λ_k .

The variational objective function to be optimized is the marginal log-likelihood of the document collection \mathcal{D} given by [73]:

$$\log p(\mathcal{D} | \gamma, \alpha_0, \zeta) \geq \mathbb{E}_q[\log p(\mathbf{D}, \boldsymbol{\beta}', \boldsymbol{\pi}', \mathbf{c}, \mathbf{z}, \boldsymbol{\phi})] + H(q) \quad (3.32)$$

$$= \sum_j \{ \mathbb{E}_q[\log(p(\mathbf{w}_j | \mathbf{c}_j, \mathbf{z}_j, \boldsymbol{\phi})p(\mathbf{c}_j | \boldsymbol{\beta}')p(\mathbf{z}_j | \boldsymbol{\pi}'_j)p(\boldsymbol{\pi}'_j | \alpha_0))] \} \quad (3.33)$$

$$+ H(q(\mathbf{c}_j)) + H(q(\mathbf{z}_j)) + H(q(\boldsymbol{\pi}'_j)) \} \quad (3.34)$$

$$+ \mathbb{E}_q[\log p(\boldsymbol{\beta}')p(\boldsymbol{\phi})] + H(q(\boldsymbol{\beta}')) + H(q(\boldsymbol{\phi})) \quad (3.35)$$

$$= \mathcal{L}(q) \quad (3.36)$$

Where $H(\cdot)$ is the entropy term for the variational distribution.

3.5 Continuous-time dynamic topic model

On the other hand, Wang et al. [72] proposed a continuous-time dynamic topic model. That model uses Brownian motion to model [54] the evolution of topics over time. Even though that model uses a novel sparse variational Kalman filtering algorithm for fast inference, the number of topics it samples from is bounded, and that severely limits its application in news feed storyline creation and article aggregation. When the number of topics covered by the news feed is less than the pre-tuned number of topics set of the model, similar stories will show under different storylines. On the other hand, if the number of topics covered becomes greater than the pre-set number of topics, topics and storylines will get conflated.

Statistical model

A graphical model representation for this model using plate notation is given in Figure 3.3.

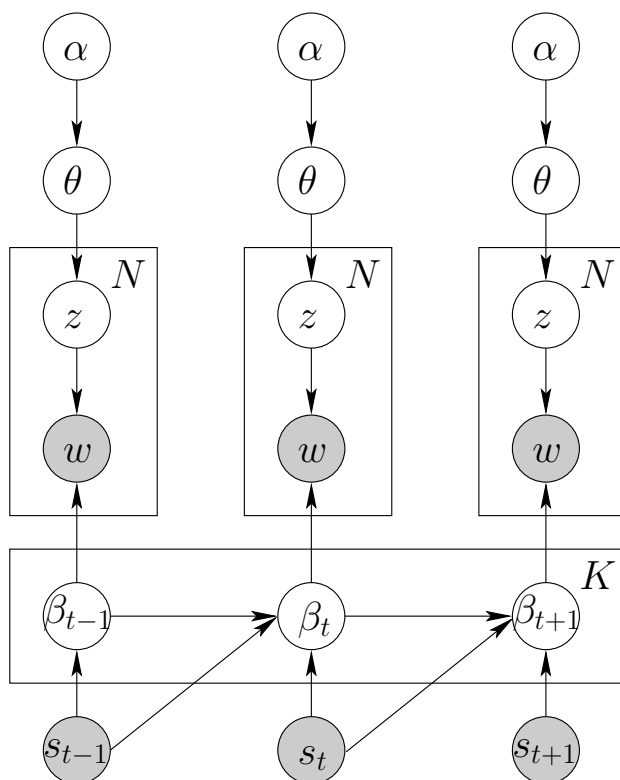


Figure 3.3: Graphical model representation of the continuous-time dynamic topic model using plate notation.

Let the distribution of the k^{th} topic parameter for word w be:

$$\beta_{0,k,w} \sim \mathcal{N}(m, v_0) \quad (3.37)$$

$$\beta_{j,k,w} | \beta_{i,k,w}, s \sim \mathcal{N}(\beta_{i,k,w}, v \Delta_{s_j, s_i}) \quad (3.38)$$

where i, j are time indexes, s_i, s_j are timestamps, and Δ_{s_j, s_i} is the time elapsed between them. In this model, the multinomially distributed topic distribution at time t is sampled from a Dirichlet distribution $\theta_t \sim \text{Dir}(\alpha)$, and then a topic $z_{t,n}$ is sampled from a multinomial distribution parametrized by θ_t , $z_{t,n} \sim \text{Mult}(\theta_t)$.

To make the topics evolve over time, we define a Wiener process [54] $\{X(t), t \geq 0\}$ and sample β_t from it. The obtained unconstrained β_t can then be mapped on the simplex. More formally:

$$\beta_{t,k} | \beta_{t-1,k}, s \sim \mathcal{N}(\beta_{t-1,k}, v \Delta_{st} I) \quad (3.39)$$

$$\pi(\beta_{t,k})_w = \frac{\exp(\beta_{t,k,w})}{\sum_w \exp(\beta_{t,k,w})} \quad (3.40)$$

$$w_{t,n} \sim \text{Mult}(\pi(\beta_{t,z_{t,n}})) \quad (3.41)$$

where $\pi(\cdot)$ maps the unconstrained multinomial natural parameters to its mean parameters, which are on the simplex.

The posterior, which is the distribution of the latent topic structure given the observed documents, is intractable. We resort to approximate inference. For this model, sparse variational inference presented in [72] could be used.

Symbol	Definition
B_d	number of topics in document d
$\mathcal{N}(m, v_0)$	a Gaussian distribution with mean m and variance v_0^2
$\beta_{i,k,w}$	distribution of words over topic k at time i for word w
s_i	timestamp for time index i
Δ_{s_j, s_i}	time duration between s_i and s_j
$Dir(\cdot)$	Dirichlet distribution
$z_{t,n}$	topic n sampled at time t
I	identity matrix
$\text{Mult}(\cdot)$	multinomial distribution
$\pi(\cdot)$	mapping function

Table 3.2: *Table of notations for cDTM*

Chapter 4

Continuous-time infinite dynamic topic model

In this chapter I describe my own contribution, the continuous-time infinite dynamic topic model.

4.1 Dim sum process

I propose building a topic model called the *continuous-time infinite dynamic topic model* (ciDTM) that combines the properties of 1) the continuous-time Dynamic Topic Model (cDTM), and 2) the online Hierarchical Dirichlet Process model (oHDP).

I will refer to the stochastic process I develop that combines the properties of these two systems as the *Dim Sum Process*.

Dim sum is a style of Chinese food. One important feature of dim sum restaurants is the freedom given to the chef to create new dishes depending on seasonal availability and what he thinks is auspicious for that day. This leeway given to the cook leads to change over time of the ingredient mixture of the different dishes the restaurant serve to better satisfy the customers taste and suit seasonal changes and availability of these ingredients.

The generative process of a dim sum process is similar to that of a Chinese restaurant franchise process [68] in the way customers are seated in the restaurant and the way dishes are assigned to tables, but differs in that the ingredient (mixture) of the dishes evolves in

continuous time.

When the dim sum process is used in a topic modeling application, each customer corresponds to a word and the restaurant represents a document. A dish is mapped to a topic and a table is used to group a set of customers to a dish.

The generative process of the dim sum process proceeds as follows:

- Initially the dim sum restaurant (document) is empty.
- The first customer (word) to arrive sits down at a table and orders a dish (topic) for his table.
- The second customer to arrive has two options. 1) She can sit at a new table with probability $\alpha/(1 + \alpha)$ and orders food for her new table, or 2) she can sit at the occupied table with probability $1/(1 + \alpha)$ and eat from the food that has been already ordered for that table.
- When the $n - 1$ customer enters the restaurant, he can sit down at a new table with probability $\alpha/(n + \alpha)$, or he can sit down at table k with probability $n_k/(n + \alpha)$ where n_k is the number of customers currently sitting at table k .

Higher values of α leads to higher number of occupied tables and dishes (topics) sampled in the restaurant (document). This model can be extended to a franchise restaurant setting where all the restaurants share one global menu from which the customers order. To do so, each restaurant samples its parameter α and its local menu from a global menu. This global menu is a higher level Dirichlet process. This two level Dirichlet process is known in the literature as the Chinese restaurant franchise process (CRFP) [68].

The main difference between a dim sum process and a CRFP is in the global menu. This global menu is kept fixed over time in the CRFP setting, whereas it evolves in continuous time in the dim sum process using a Brownian motion model [54].

The dim sum process can be described using plate notation as shown in Figure 4.1. This model combines the two models in that it gets rid of the highest level time specific

hierarchical Dirichlet process. It implicitly ties the base measures G_0 across all documents. This measure which ensures sharing of the topics through time and over documents, is sampled from a $DP(\gamma, H)$, and it has been integrated out from our model. Note that I reduced the hierarchical structure of the model one level, and it now represents a Chinese restaurant process instead of a recurrent Chinese restaurant process.

The other modification I make to the model is to evolve its topic distribution using a Brownian motion model [54], similar to the one used in the continuous-time dynamic topic model.

The true posterior is not tractable; we have to resort to approximate inference for this model. Moreover, due to non-conjugacy between the distribution of words for each topic and the word probabilities, I cannot use collapsed Gibbs sampling. Instead I plan to use variational methods for inference.

Figure 4.2 shows a block diagram for the continuous-time infinite dynamic topic model operating on a single document at a time.

4.2 Mathematical model

The continuous-time infinite dynamic topic model (ciDTM) is a mixture of the continuous-time dynamic topic model presented earlier in Section 3.5, and the online hierarchical Dirichlet process (oHDP) model presented in Section 3.4.

A generative process for this ciDTM using dim sum process proceeds as follows:

We build a two level hierarchical Dirichlet process (HDP) like the one presented in Section 3.4.

At the top level of a two level hierarchical Dirichlet process (HDP), a Dirichlet distribution G_0 is sampled from a Dirichlet process (DP). This distribution G_0 is used as the base measure for another DP at the lower level from which another Dirichlet distribution G_j is drawn. This means that all the distributions G_j share the same set of atoms they inherited from their parent with different atom weights. Formally put:

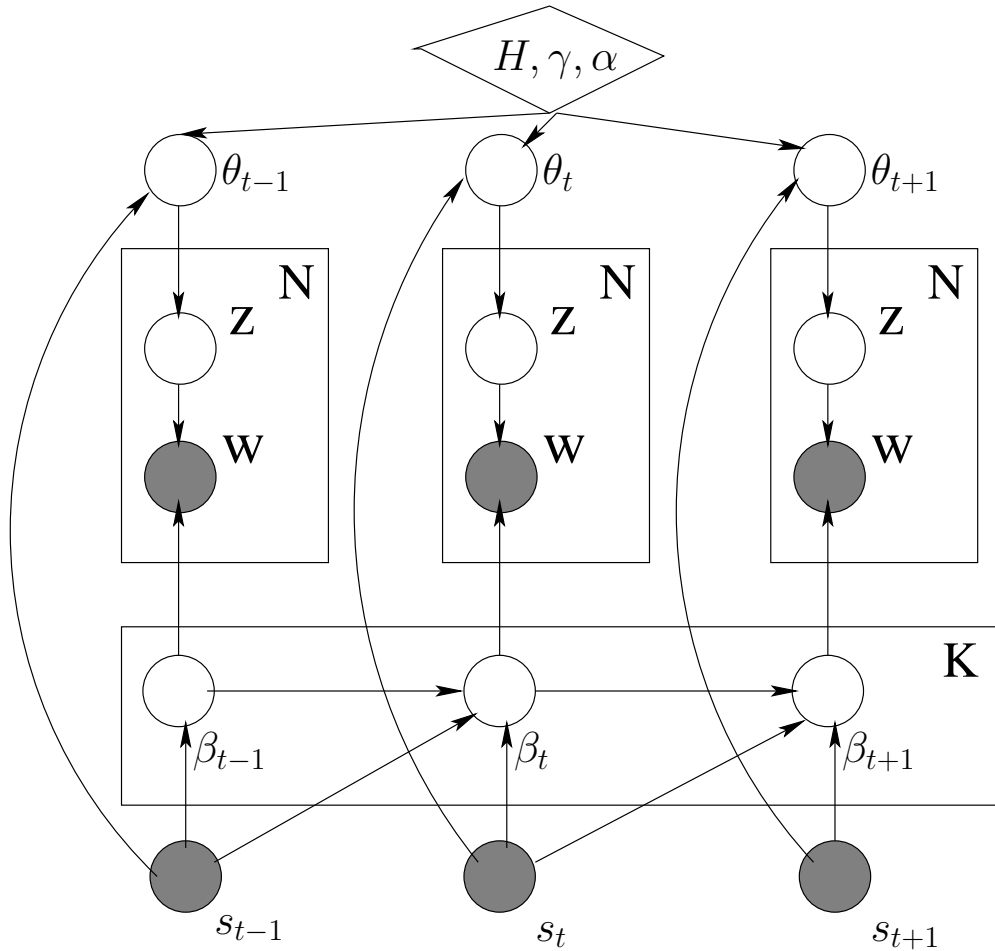


Figure 4.1: *Continuous-time infinite dynamic topic model.* None-shaded nodes represent unobserved latent variables, shaded nodes are observed variables, diamonds represent hyperparameters, and plates represent repetition. Symbols in this figure follow the notation in Figure 3.2 and H is the base distribution for the upper level DP, γ and α are the concentration parameters for the upper level, and lower level DPs respectively, and s_t is the timestamp for document at time t .

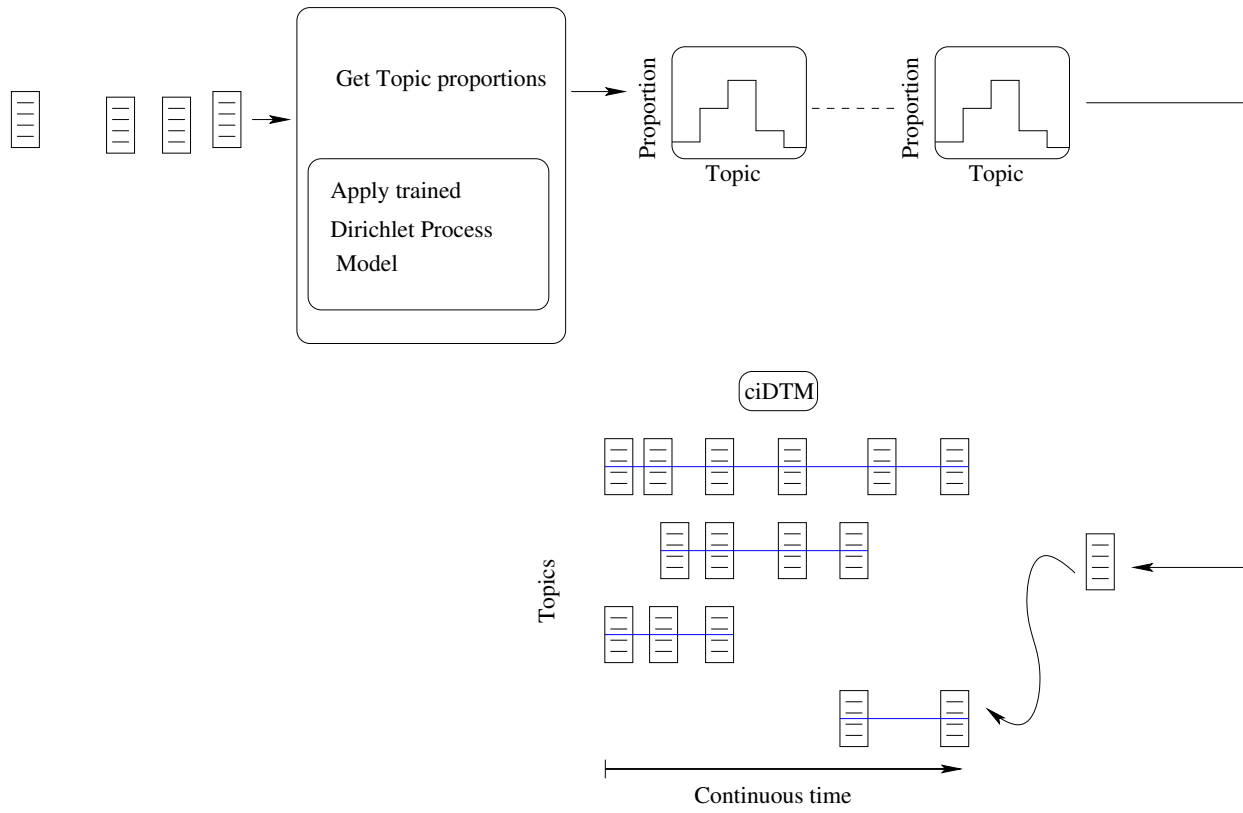


Figure 4.2: A block diagram describing the workflow for the continuous-time infinite dynamic topic model working on a single document at a time.

$$G_0 \sim DP(\gamma, H) \tag{4.1}$$

$$G_j \sim DP(\alpha_0, G_0) \tag{4.2}$$

Where γ and H are the concentration parameter and base measure for the first level DP, α_0 and G_0 are the concentration parameter and base measure for the second level DP, and G_j is the Dirichlet distribution sampled from the second level DP.

In a topic model utilizing this HDP structure, a document is made of a collection of words, and each topic is a distribution over the words in the document collection. The atoms of the top level DP are the global set of topics. Since the base measure of the second level DP is sampled from the first level DP, then the sets of topics in the second level DP are subsets of the global set of topics in the first level DP. This ensures that the documents sampled from the second level process share the same set of topics in the upper level. For each document j in the collection, a Dirichlet G_j is sampled from the second level process. Then, for each word in the document a topic is sampled then a word is generated from that topic.

In Bayesian non-parametric models, variational methods are usually represented using a stick-breaking construction. This representation has its own set of latent variables on which an approximate posterior is given [9, 44, 69]. The stick-breaking representation used for this HDP is given at two levels: corpus-level draw for the Dirichlet G_0 from the top-level DP, and a document-level draw for the Dirichlet G_j from the lower-level DP. The corpus-level sample can be obtained as follows:

$$\beta'_k \sim \text{Beta}(1, \gamma) \quad (4.3)$$

$$\beta_k = \beta'_k \prod_{l=1}^{k-1} (1 - \beta'_l) \quad (4.4)$$

$$\phi_k \sim H \quad (4.5)$$

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k} \quad (4.6)$$

where γ is a parameter for the Beta distribution, β_k is the weight for topic k , ϕ_k is atom (topic) k , H is the base distribution for the top level DP, and δ is the Dirac delta function.

The second level (document-level) draws for the Dirichlet G_j are done by applying Sethuraman stick-breaking construction of the DP again as follows:

$$\psi_{jt} \sim G_0 \quad (4.7)$$

$$\pi'_{jt} \sim \text{Beta}(1, \alpha_0) \quad (4.8)$$

$$\pi_{jt} = \pi'_{jt} \prod_{l=1}^{t-1} (1 - \pi'_{jl}) \quad (4.9)$$

$$G_j = \sum_{t=1}^{\infty} \pi_{jt} \delta_{\psi_{jt}} \quad (4.10)$$

where ψ_{jt} is a document-level atom (topic) and π_{jt} is the weight associated with it.

This model can be simplified by introducing indicator variables \mathbf{c}_j that are drawn from the weights $\boldsymbol{\beta}$ [73]

$$c_{jt} \sim \text{Mult}(\boldsymbol{\beta}) \quad (4.11)$$

The variational distribution is thus given by:

$$q(\boldsymbol{\beta}', \boldsymbol{\pi}', \mathbf{c}, \mathbf{z}, \boldsymbol{\phi}_0) = q(\boldsymbol{\beta}')q(\boldsymbol{\pi}')q(\mathbf{c})q(\mathbf{z})q(\boldsymbol{\phi}_0) \quad (4.12)$$

$$q(\boldsymbol{\beta}') = \prod_{k=1}^{K-1} q(\beta'_k | u_k, v_k) \quad (4.13)$$

$$q(\boldsymbol{\pi}') = \prod_j \prod_{t=1}^{T-1} q(\pi'_{jt} | a_{jt}, b_{jt}) \quad (4.14)$$

$$q(\mathbf{c}) = \prod_j \prod_t q(c_{jt} | \varphi_{jt}) \quad (4.15)$$

$$q(\mathbf{z}) = \prod_j \prod_n q(z_{jn} | \zeta_{jn}) \quad (4.16)$$

$$q(\boldsymbol{\phi}_0) = \prod_k q(\phi_k | \lambda_k) \quad (4.17)$$

where $\boldsymbol{\beta}'$ is the corpus-level stick proportions and (u_k, v_k) are parameters for its beta distribution, $\boldsymbol{\pi}'_j$ is the document-level stick proportions and (a_{jt}, b_{jt}) are parameters for its beta distribution, \mathbf{c}_j is the vector of indicators, $\boldsymbol{\phi}_0$ is the topic distributions, and \mathbf{z} is the topic indices vector. In this setting, the variational parameters are φ_{jt} , ζ_{jn} , and λ_k .

The variational objective function to be optimized is the marginal log-likelihood of the document collection \mathcal{D} given by [73]:

$$\log p(\mathcal{D} | \gamma, \alpha_0, \zeta) \geq \mathbb{E}_q[\log p(\mathbf{D}, \boldsymbol{\beta}', \boldsymbol{\pi}', \mathbf{c}, \mathbf{z}, \boldsymbol{\phi}_0)] + H(q) \quad (4.18)$$

$$= \sum_j \{ \mathbb{E}_q[\log(p(\mathbf{w}_j | \mathbf{c}_j, \mathbf{z}_j, \boldsymbol{\phi}_0)p(\mathbf{c}_j | \boldsymbol{\beta}')p(\mathbf{z}_j | \boldsymbol{\pi}'_j)p(\boldsymbol{\pi}'_j | \alpha_0))] \} \quad (4.19)$$

$$+ H(q(\mathbf{c}_j)) + H(q(\mathbf{z}_j)) + H(q(\boldsymbol{\pi}'_j)) \} \quad (4.20)$$

$$+ \mathbb{E}_q[\log p(\boldsymbol{\beta}')p(\boldsymbol{\phi}_0)] + H(q(\boldsymbol{\beta}')) + H(q(\boldsymbol{\phi}_0)) \quad (4.21)$$

$$= \mathcal{L}(q) \quad (4.22)$$

Where $H(\cdot)$ is the entropy term for the variational distribution.

I use coordinate ascent to maximize the log-like likelihood given in (4.18). Next, given the per-topic word distribution $\boldsymbol{\phi}_0$, I use a Wiener motion process [54] to make the topics

evolve over time. I define the process $\{X(t), t \geq 0\}$ and sample ϕ_t from it. The obtained unconstrained ϕ_t can then be mapped on the simplex. More formally:

$$\phi_{t,k} | \phi_{t-1,k}, s \sim \mathcal{N}(\phi_{t-1,k}, v\Delta_{st}I) \quad (4.23)$$

$$\pi(\phi_{t,k})_w = \frac{\exp(\phi_{t,k,w})}{\sum_w \exp(\phi_{t,k,w})} \quad (4.24)$$

$$w_{t,n} \sim \text{Mult}(\pi(\phi_{t,z_t,n})) \quad (4.25)$$

where $\pi(\cdot)$ maps the unconstrained multinomial natural parameters to its mean parameters, which are on the simplex.

The posterior, which is the distribution of the latent topic structure given the observed documents, is intractable. We resort to approximate inference. For this model, sparse variational inference presented in [72] could be used.

Chapter 5

Testbed development

In this chapter, I will motivate the need for a continuous-time infinite dynamic topic model and present the dataset/corpus I will be using for my testbed and evaluation of the model I will develop as well as other competing models. I start by describing the corpus:

1. Why do I need the corpus?
2. What should this corpus be made of?
3. What makes a good corpus?
4. How can a good corpus be created?
5. Are there corpora that fit my needs?
6. Can I modify existing corpora to fit my needs?
7. What are the challenges in creating such a corpus?
8. Who else could benefit from this corpus if I can publicly publish it? and
9. Can I publicly publish this corpus?

After that, I will define the problem of creating timelines for different news stories which has many potential applications especially in the news media industry. I will follow that by several attempts to solve this problem, at first using a simple topic model, then successively using more advanced models that alleviate some of the shortcomings of the earlier models.

5.1 News stories corpus

In order to test the performance of my ciDTM and create news timelines I need a news corpus. This corpus should be made of a collection of news stories. These stories could be collected from news outlets, like newspapers, newswire, transcripts of radio news broadcasts, or crawled from websites of news agencies or newspapers websites. In my case, I can crawl newspapers websites, news agencies websites, or news websites. For one of these sources to be considered a valid source for my corpus, each of the news stories they publish should contain: 1) an identification number, 2) story publication date and time, 3) story title, 4) story text body, and 5) a list of related stories.

Many news agencies like Reuters and Associated Press, news websites like BBC News and Huffington Post, and newspaper websites like The Guardian and New York Times all meet these conditions in their published stories. There are few differences though that make some of them better than the others, and each one of them has its advantages and disadvantages.

News agencies typically publish more stories per day than the other sources I considered. This makes the news timeline richer with stories and makes more news timelines. They tend to publish more follow-up stories than other sources which contributes to the richness of the timeline as well. They also cover a bigger variety of topics and larger geographical region, usually all world countries, than other sources. They do not have limitations on the word count of their stories as they are not restricted by page space or other space requirements which make their stories richer in syntax and vocabulary than other sources.

On the other hand, news agencies only publish stories produced by their own journalists, and their own journalists have to follow the agency's guidelines and rules of writing, editing and updating news stories. This restriction makes the stories more uniform regarding editorial structure, and sometimes they have vocabulary uniformity also. Even though this uniformity does not affect the journalistic quality of the news stories they publish, it

makes the task of news timeline creation easier for the topic model. The topic model could just learn the set of keywords used for the news stories that fall under a timeline that gets repeated over and over by the same journalist who covers the topic for the news agency over a certain period of time. The model could be better tested if the stories are written by different journalists belonging to different organizations and following different rules and guidelines.

The process of crawling news agencies news websites is usually easy. Their web pages are usually well formatted, have few advertisements and little unrelated content. They are usually well tagged also. News agencies often tend not to have strong political bias in their news coverage when compared to newspapers and news websites. This will be reflected in the lack of opinionated adjectives that comes with political bias which affects the vocabulary structure of the news stories.

News websites typically only exist in electronic form on the web and collect their news stories from different sources. They purchase stories from different news agencies. Some of them have their own dedicated journalists and freelance journalists, and some purchase stories from other news websites and newspapers. Many of the stories they gather from other resources pass through an editorial step in which some sections of the story may be removed for publication space limitations. In other cases, sections written by their own journalists or collected from other sources could be added to the story, or two stories could be merged to fill up publication space. This editorial process could lead to syntactic and vocabulary richness in the news stories as the stories belonging to the same timeline would have lots of synonym words, and this synonymy should be learned by the topic model in order to classify the related stories as belonging to the same timeline.

As news websites only exist in electronic form on the web, they tend to make the most of it and be rich in media format; a story could have the main story text, text commentary, audio, video and links to other websites covering the same story. This richness in media format could help in modeling the document and in placing it on the correct timeline.

This richness makes it harder for automated crawling applications to extract the news story components, such as title, story text, and related stories, from a page crammed with different kinds of content like advertisements, unrelated stories, headers, footers, and such.

News websites tend to have soft publication space limitation; a news story could run few pages long or be a couple of paragraph. The list of related stories could be rich in the case of news websites as the media richness discussed earlier usually includes links to news stories published by other sources, let us call them external sources, and by the same news website, let us call them internal sources, and that again helps in better testing the news timeline creation. Different news stories along the same news timeline will include many synonym words, and the synonymy should be learned by the topic model to identify these stories as belonging to the same timeline.

News websites tend to have more political bias than news agencies, but less bias than newspapers. This means related news stories they publish would not carry so much at the same opinionated adjectives in their coverage of the story. It is to be noted that this will usually only apply to the stories written by their own journalists, and not collected or purchased from news agencies.

Newspaper websites tend to fall in the middle between news agencies websites and news websites regarding news stories diversity, coverage and richness. Newspapers usually collect news stories from news agencies for regions of the world they do not cover. They have their own journalists who write according to the newspaper's editorial and political rules and guidelines, and they purchase stories from other newspapers also. This makes a similar syntactic and vocabulary richness in their content, even though it does not match that of the news websites. The stories collected or purchased from other sources typically go through an editorial process in which a story may be cut short, extended by adding content from other sources to it, or merged with another story purchased or collected from another sources covering the same topic. Newspapers usually put on their website all the stories they publish on paper. They sometimes publish extended versions online, and may

also have online exclusive content. The online exclusive content usually has soft space limitations. The paper-published content usually has hard space limitations though.

The process of crawling newspaper websites is usually easier than crawling news websites as the former's web pages usually contain less advertisements, less unrelated content, and less links to other sources. Newspapers, news web pages are usually well tagged. Newspaper websites tend to heavily link to their own sources, unlike news websites which link heavily to external sources. If I am interested in creating a rich news timeline, this will be counted in favor of newspaper websites as I usually crawl web pages with the same format, which means crawling only internal sources not external sources because they usually have different web page format.

Newspapers usually tend to have stronger political bias in their news coverage when compared to news agencies and news websites. They tend to use more strongly opinionated adjectives. This has the side effect of having news stories belonging to the same timeline sharing a collection of adjective keywords that makes the process of clustering the news stories into different timelines easier. This is usually not desirable as this could make the model more reliant on clustering stories based on keywords, instead of using word distribution, and even better, a word distribution that changes over time.

A good text news stories corpus would contain a syntactically and semantically rich collection of stories. The stories should be collected from different sources and written by different authors who follow different writing and editorial rules and guidelines, or even better if they do not share any set of these standards. This diversity in vocabulary and syntactic structure will translate to a larger set of synonyms and antonyms being used in the collection. The topic model is tested on the different relationships among these words and the degree to which it correctly learns these relationship translates to good performance in document classification and correct placement of a news story on its natural timeline.

A good corpus should have a big set of related stories for each news story it contains, if such related stories exist in the corpus. The bigger the set, the richer the news timeline

becomes, and the more chances of success and failure the topic model will have in creating the timeline. This will generate another challenge in discovering the birth/death/revival of topics. The longer the timeline gets, the more of this topic life cycle can be detected or missed.

The set of related news stories provided by many news outlets for each of the news stories they release could be either manually created or automatically generated. In the manual creation process, a news editor sifts through past news stories manually or assisted by search tools and looks for relevant stories. The relevancy judgment is done by a human. The number of stories in the set of related stories is usually kept within a certain limit to emphasize the importance and relatedness of the stories in the set. The set usually includes the more recent five or seven related stories. A related stories set created manually this way is needed for my corpus to be used in testing the performance of the timeline creation algorithm and the topic detection and tracking algorithm. This manually created set is called the *gold standard*. It represents the highest possible standard that any algorithm trying to solve this problem should seek to match.

Not all news outlets use human annotators to judge the relatedness of the news stories to create a set of related stories. Some of them use algorithms to do this job. This is usually driven by the need to avoid the cost of having a human annotator. Related stories generated by such systems, like Newstracker¹ system being used by BBC News, cannot be used as a gold standard for my system. They do not represent the ultimate standard that cannot be surpassed. Their performance can be improved upon by other systems addressing the same problem. They can be used as a baseline for the performance of other systems that tries to match or even exceed their performance.

A good corpus should have news stories time stamped with high time granularity. Some news sources like news agencies publish news stories round-the-clock; Agence France-Presse (AFP) releases, on average, one story every 20 seconds². Such a fast-paced publication

¹BBC links to other news sites: <http://www.bbc.co.uk/news/10621663>

²Agence France-Presse (AFP) releases on average one news story every 20 seconds (5000 per day) while

needs a few minutes' resolution and accurate time stamped stories to correctly place the story on its timeline.

A news timeline typically contains many news stories, usually over ten stories, and extends over a relatively long period of time, several months long. One set of related news stories cannot be used to create a timeline; it typically contains less than ten related stories, and in many cases, the stories extend over a few days or a week. To be able to create a timeline, different sets of related news stories have to be chained. For each story in the set of related stories, I get its set of related stories. For each one of these stories in turn, I get a set of related stories. This process can repeat and I can extend the chain as desired. However, the longer the chain gets, the less related the stories at the end of the chain will be to the original story at the other end of the chain. This is because at each step along the chain we compare the stories to the current node (story) in the chain, and not the first node (original) that we want to get a set of stories related to it.

I create a news corpus by crawling news websites like the British newspaper *The Guardian* website. I start by a set of diverse news stories seeds, or links. This set typically covers a wide variety of topics and geographical areas. For each one of these links, I use the link as the story identifier. I crawl the main news story text and title, its release date and time, and the set of related news stories. I follow the links to the related news stories section to create the set of related news stories. These related stories were hand-picked by a news editor and therefore I can use them for my gold standard. I repeat this process until I crawl a predefined number of news stories.

5.2 Corpora

For my experiments I use two corpora: a Reuters corpus, and a BBC news corpus.

Reuters releases 800,000 English-language news stories annually. See: <http://www.afp.com/en/agency/afp-in-numbers> and <http://www.infotoday.com/it/apr01/news6.htm>

5.2.1 Reuters

The Reuters-21578 corpus [47]³ is currently the most widely used test collection for text categorization research, and it is available for free online. The corpus is made of 21578 documents in English that appeared on the Reuters newswire in 1987. The average number of unique words in a news story in this corpus is 58.4. It has a vocabulary of 28569 unique words. This collection of news articles comes in XML format. Each document comes with:

Date The date the story was published. This date is accurate to milliseconds. E.g. 26-FEB-1987 15:01:01.79.

Topic A manually assigned topic that the news story discusses. There are 135 topics. E.g. Cocoa.

Places A Geographical location where the story took place. E.g. El-Salvador.

People Names of famous people mentioned in the corpus. E.g. Murdoch.

Organizations Names of organizations mentioned in the corpus. E.g. ACM.

Exchanges Abbreviations of the various stock exchanges mentioned in the corpus. E.g. NASDAQ.

Companies Names of companies mentioned in the corpus. E.g. Microsoft.

Title The title of the news story. E.g. Bahia cocoa review.

Body The body of the news story. E.g. Showers continued throughout the week in the Bahia cocoa zone, ...

In my experiments, I only use *Date* and *Body*.

³<http://www.daviddlewis.com/resources/testcollections/reuters21578/>

5.2.2 BBC News

The BBC news corpus is made of 10,000 news stories in English I collected myself from the BBC news website⁴. The stories in the corpus cover about two and a half years time period, from April 2010 to November 2012. The average number of unique words in a news story in this corpus is 189.6 and the corpus has a vocabulary of 64374 unique words. For each news story I collect:

ID An identification for the news story. I use the story web page URL for that. E.g. <http://www.bbc.co.uk/news/world-europe-10912658>

Date The date the story was published. This date is accurate to seconds. E.g. 2010/08/09 15:51:53

Title The title of the news story. E.g. Death rate doubles in Moscow as heatwave continues

Body The body of the news story. E.g. Death rate doubles in Moscow as heatwave continues Extreme heat and wildfires have led to ...

Related The ID of a related news story. E.g. <http://www.bbc.co.uk/news/world-europe-10916011>

5.3 Evaluation of proposed solution

To evaluate the performance of the proposed solution I am going to use the per-word log-likelihood of a news story conditioned on the rest of the data. More formally, the performance measure will be:

$$likelihood_{pw} = \frac{\sum_{\mathbf{w} \in \mathcal{D}} \log p(\mathbf{w} | \mathcal{D}_{-\mathbf{w}})}{\sum_{\mathbf{w} \in \mathcal{D}} |\mathbf{w}|} \quad (5.1)$$

⁴<http://www.bbcnews.com>

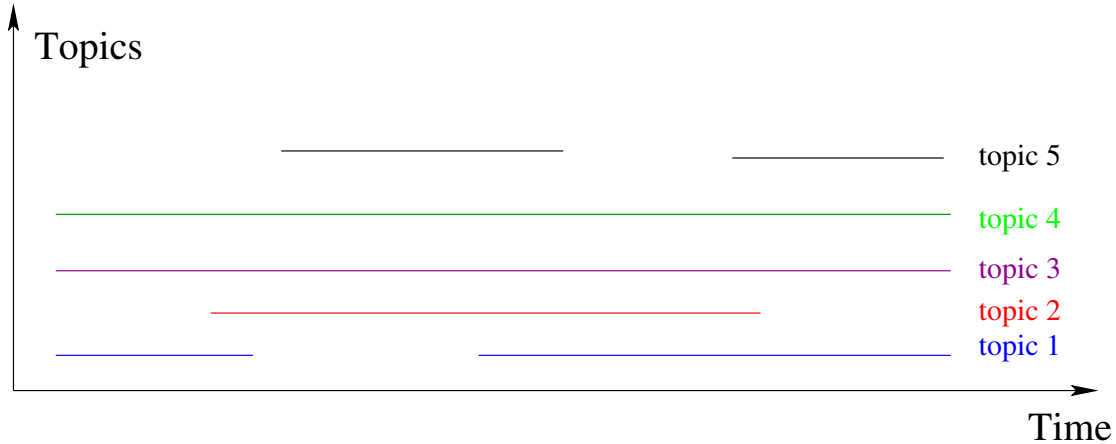


Figure 5.1: *Illustration of the birth/death/resurrection of five topics. Discontinuity of a topic indicates its death.*

Where \mathbf{w} is a document in the collection of documents \mathcal{D} , $|\mathbf{w}|$ is the number of words in document \mathbf{w} , and $\mathcal{D}_{-\mathbf{w}}$ is the collection of documents \mathcal{D} minus document \mathbf{w} .

The use of log-likelihood as an evaluation measure for topic models is widely accepted in the machine learning community [19, 69, 71]. It is a measure of the goodness of fit of the model to the data [3]. The better the fit, the more likely the model will label the documents with the right topics.

Using the natural log function is convenient for several reasons. 1) It reduces the chances of running into an underflow problem due to very small likelihood values when simulating the model. 2) It allows us to use summation of terms (computationally cheap) instead of product of terms (more computationally expensive), and 3) The natural log function is a monotone transformation which means that extrema of the log-likelihood function is equivalent to the extrema of the likelihood function [58].

Scalability A real world news feed contains hundreds of thousands of archived news articles with more articles being generated every minute. An offline model is expected to be able to build the storyline of a hundred thousand documents in less than a day running on a desktop machine. An online model, if implemented, should be able to streamline the incoming documents in real time.

Accuracy The accuracy of the model will be measured with the per-word log-likelihood given in (5.1). A higher value of the per-word log-likelihood is desirable. It means that the documents have topics with word distributions that are consistent with the per-topic word distributions the model has learned.

Chapter 6

Experimental design and results

I ran experiments to evaluate the performance of my ciDTM model against the continuous-time dynamic topic model (cDTM) presented in Section 3.5, and the online Hierarchical Dirichlet process presented in Section 3.4. To find the best settings for each of the competing models, I evaluated their performance with different practical setting values. In the next two sections I show the results obtained by running these experiments on the Reuters-21578 and the BBC news corpora presented in Section 5.2 on page 53.

I used the coded provided by Wang et al. [72]¹ to build and run the cDTM system, and used the code provided by Wang et al. [73]² to build and run the oHDP system. In my own system, ciDTM, I used some code from the previous two software packages to build and run it. I used MALLET [53] also for corpus creation, parsing and filtering.

6.1 Reuters corpus

cDTM number of topics: Since the cDTM has a fixed number of topics, it is expected that its performance will vary with different values of this parameter. The closer this value is to the average number of topics this model can fit to the data over time the higher the performance will be. I first evaluated the performance of the cDTM with different values for the number of topics ranging from 1 to 200 topics. The per-word log-likelihood values

¹<http://www.cs.cmu.edu/~chongw/software/cdtm.tar.gz>

²<http://www.cs.cmu.edu/~chongw/software/onlinehdp.tar.gz>

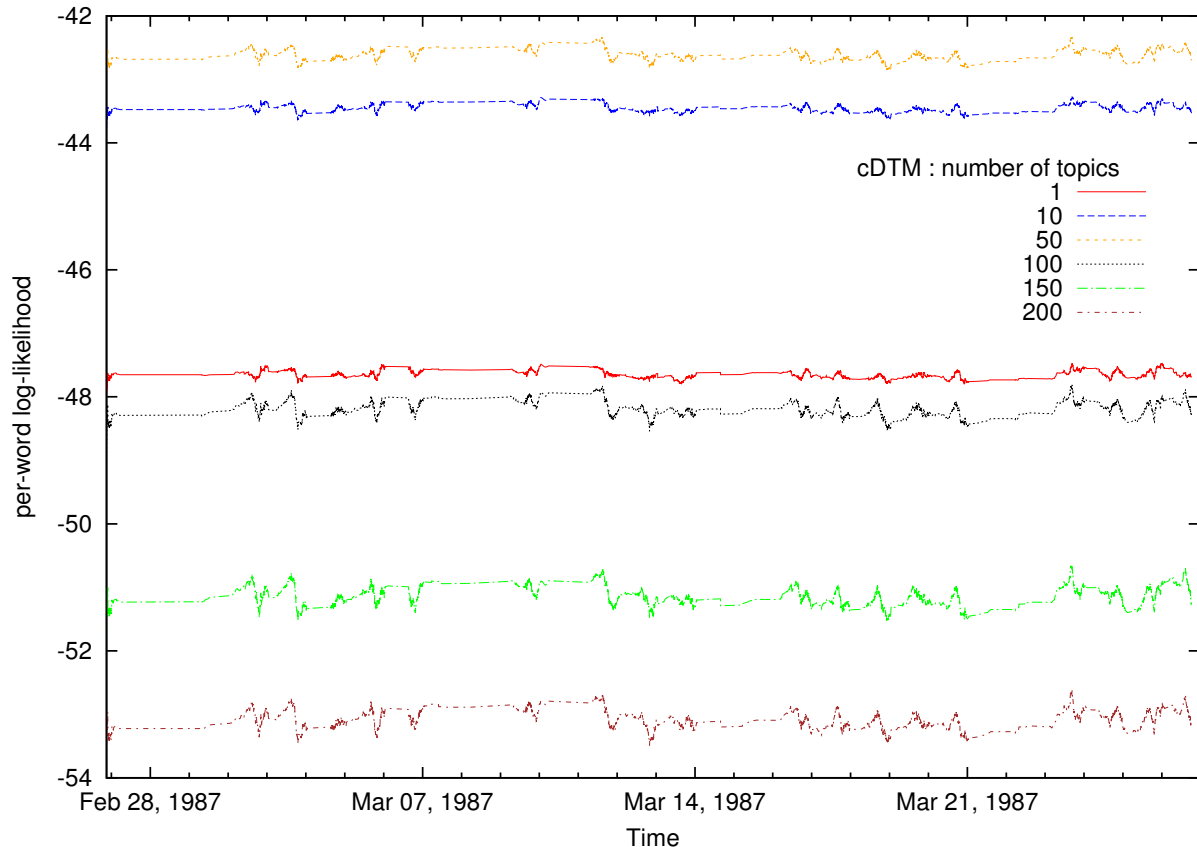


Figure 6.1: *cDTM* performance varies with different values for the number of topics using Reuters corpus. A value of 50 leads to best model fit among the six values tried. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.

for the inferred documents is presented in Figure 6.1. Higher values indicate better model performance. In all *cDTM* experiments to follow, the *cDTM* model was first (unsupervised) trained on a set of 5,000 documents sampled uniformly from the entire corpus of 10,000 documents, then the performance of the trained model was evaluated using the other documents in the corpus. This guarantees that the documents the model is tested on cover the same time range as the training documents.

Figure 6.1 shows that among the six values tried, the model performed best with 50 topics. This reflects the moderate richness of the 5,000 Reuters news stories the model was

tested on. By manually inspecting the corpus, I found that most of its documents mainly discuss economy and finance and slightly covers politics. This corpus is considered somehow limited in the topics it covers as compared to the wide spectrum of topics covered by the stories released by Reuters news agency.

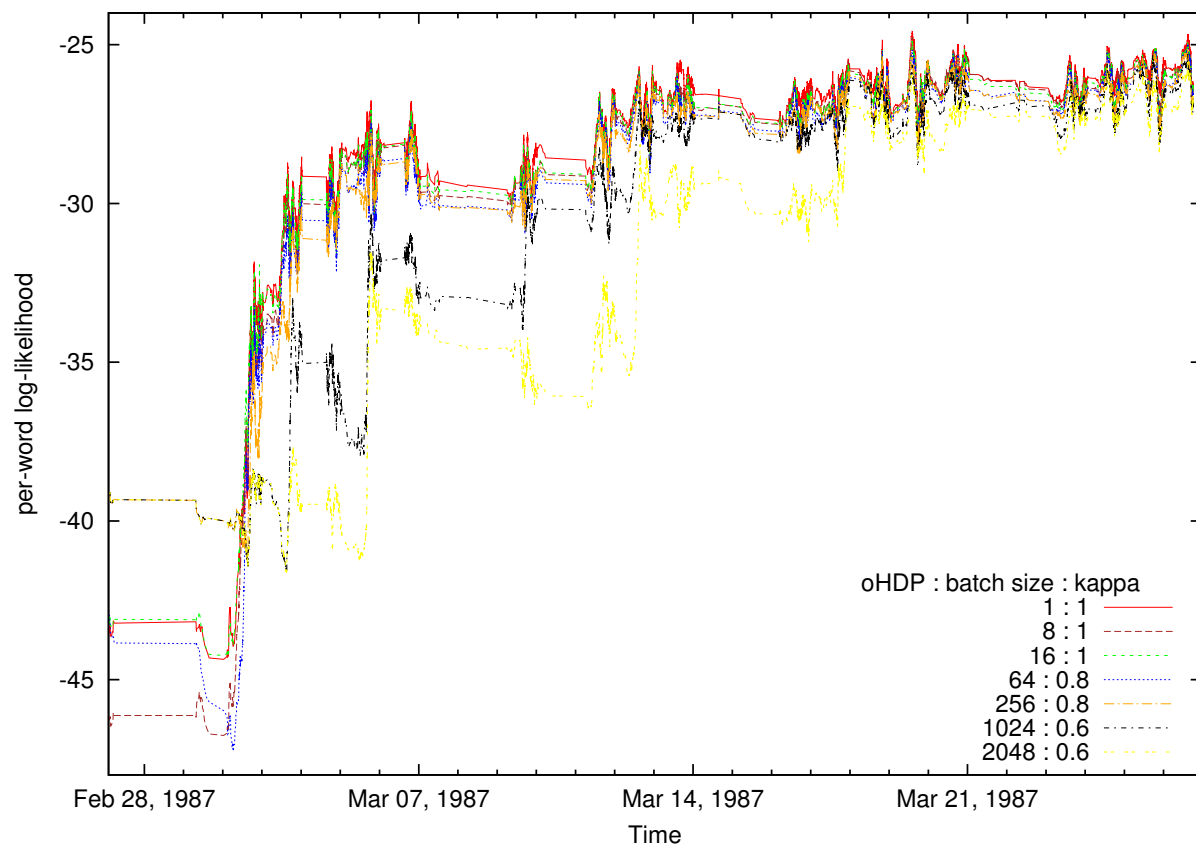


Figure 6.2: *oHDP per-word log-likelihood for different batch size values using Reuters corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.*

oHDP batch size oHDP uses online variational inference to fit its parameters. For each iteration of the algorithm, it uses a small batch of documents to update these parameters. I experimented with different batch size and model parameter (κ) values to determine the best value to use. These values were suggested by Wang et al. [73] as they generated the

best results in their case. Lower κ values favor larger batch sizes. It is to be noted that this batch size value should be relatively small to effectively retain the online inference feature of this algorithm. Experiment results with seven values are presented in Figure 6.2. The two largest batch sizes, 1024 and 2048, were only included to show the trend of decreasing performance with increasing batch size. It is not used later on because such a large batch size requires longer period of waiting time to collect the documents and update the system making the system out-of-date in the meantime.

For the oHDP model, I use the same values suggested by Wang et al. [73] and given in Table 6.1.

T	300
K	20
α	1
ζ	0.01
γ	1

Table 6.1: *Setting values used for oHDP*

The per-word log-likelihood values for the model is better with smaller batch sizes. The difference is minor for batches smaller than 64.

ciDTM batch size The size of the batch of documents ciDTM processes with every iteration of its algorithm has a double effect: 1) it affects the convergence of the online variational inference algorithm as was the case with oHDP above, and 2) it affects the variational Kalman filter used to evolve the per-topic word distribution in continuous-time. It is expected that larger batches of documents would improve the performance of the model as the document timestamp (arrival time) information and inter-arrival times between documents will be used by the filter to dynamically evolve the per-topic word distribution. It is to be noted that a batch of size one cannot be used in ciDTM. The Kalman filter used by the model evolves per-topic word distribution based on documents inter-arrival times in the batch. The process running the filter starts fresh with every new batch of documents.

Figure 6.3 shows the results of an experiment in which four different batch size values were tried.

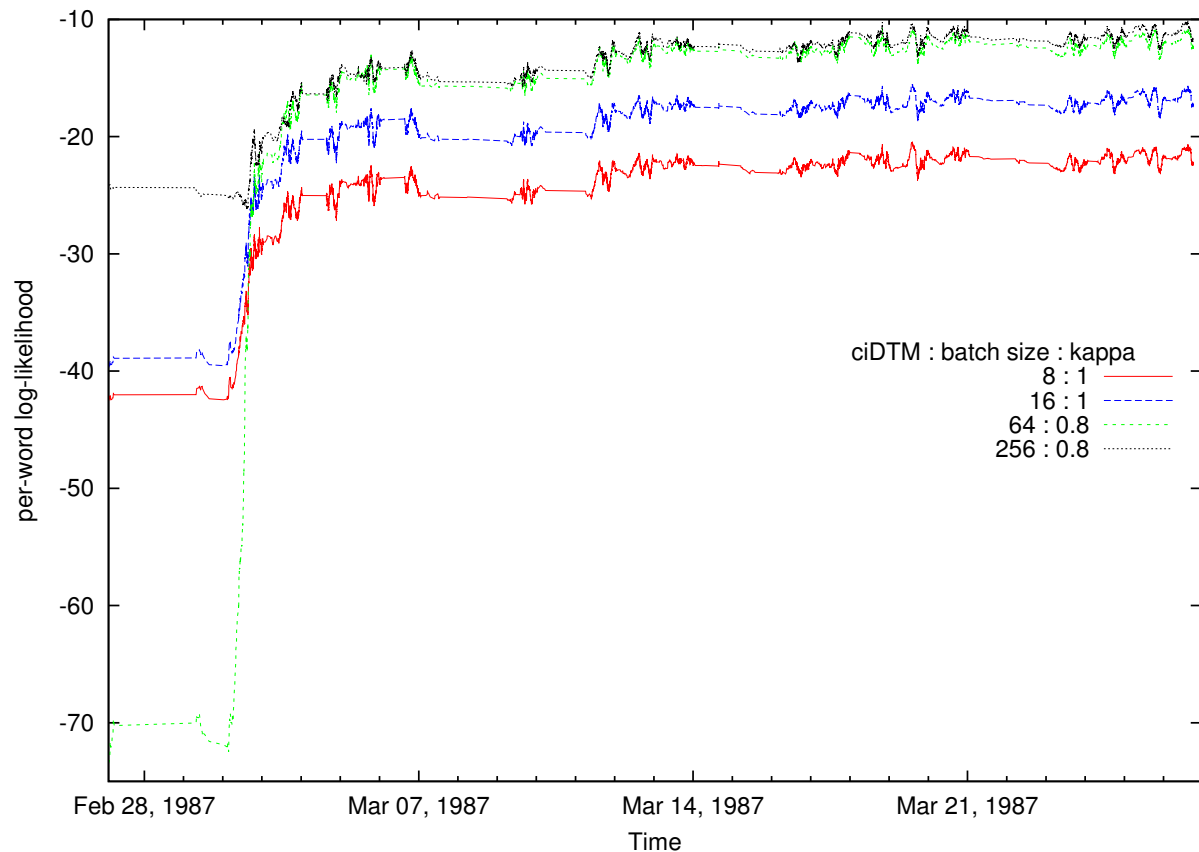


Figure 6.3: *ciDTM* per-word log-likelihood for different batch size values using Reuters corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.

Since this model has the benefit of using an online variational inference algorithm, the batch size should be kept small to maintain this desirable feature. From the Figure we can see that a point of diminishing return is reached with a batch of size 64. The performance gained by increasing the batch size from 64 to 256 is negligible. On the other hand, if documents arrive to the system at a fixed rate. Then 64 documents will need 4.5 hours to gather, while 256 documents would require 18 hours to collect. With the pace of 14

documents per hour encountered in this corpus, once a day update for the model would be enough to keep it up-to-date. I use a batch size of 256 as the optimal batch size for this model.

The ciDTM model parameter values I used are equal to their corresponding parameter values in the cDTM and oHDP models, where applicable. More specifically, the model parameters I used are:

T	300
K	20
α	1
ζ	0.01
γ	1
α_0	0.2

Table 6.2: *Setting values used for ciDTM*

Comparing three models Figure 6.4 shows the per-word log-likelihood performance for the three competing models using their best setting values discovered earlier. It is clear that ciDTM outperform both cDTM and oHDP by a big margin when tested on this news corpus especially ciDTM with batch sizes greater than 16.

Figure 6.5 shows the non-smoothed per-word log-likelihood values for each of the three models with ciDTM with batch size of 10, oHDP with batch size of 1, and cDTM with 10 topics. ciDTM outperformed the oHDP, while the cDTM performed the words of all three. The improvement the ciDTM model over oHDP is significant knowing that the value on the independent axis is a log value. The per-word log-likelihood value is evaluated for each document in the mini batch of documents processed by the ciDTM in each iteration of the algorithm.

Since cDTM relies on an offline learning algorithm. The per-word log-likelihood obtained is evaluated after the model was trained on the training data set. This explains the steady state performance of the cDTM algorithm as the word distribution for the topics covered by the corpus were learned by the model first before the log-likelihood values were evaluated.

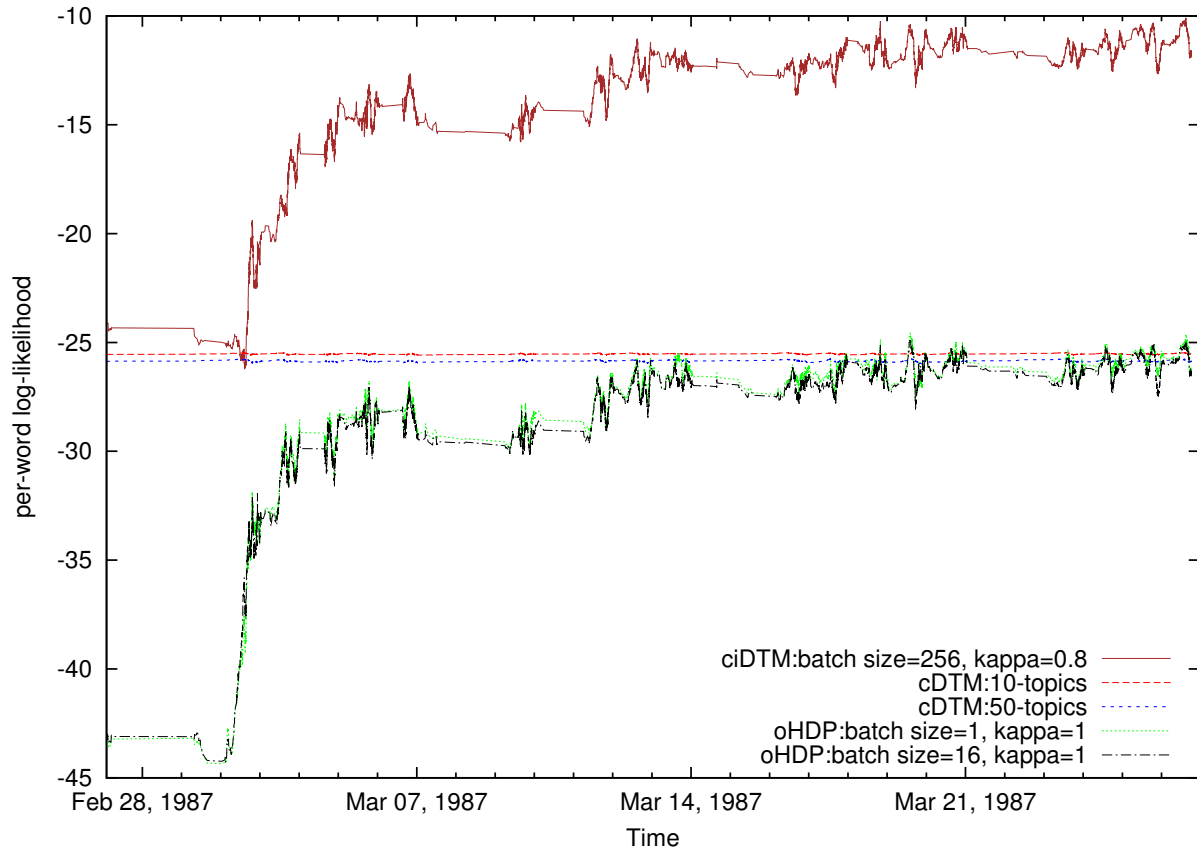


Figure 6.4: A comparison of *ciDTM*, *oHDP* and *cDTM* using their best setting values discovered earlier using Reuters corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.

The fluctuation in the data in the case of *ciDTM* and *oHDP* is due to the fact that they are encountering new documents with every time step. As the model learns the word distribution for these topics, the fluctuations starts decreasing over time. The gap periods in the graph are dormancy periods when no news stories were being published.

6.2 BBC news corpus

I repeated the same set of experiments I ran above and used the BBC news corpus presented in Section 5.2.2 on page 55.

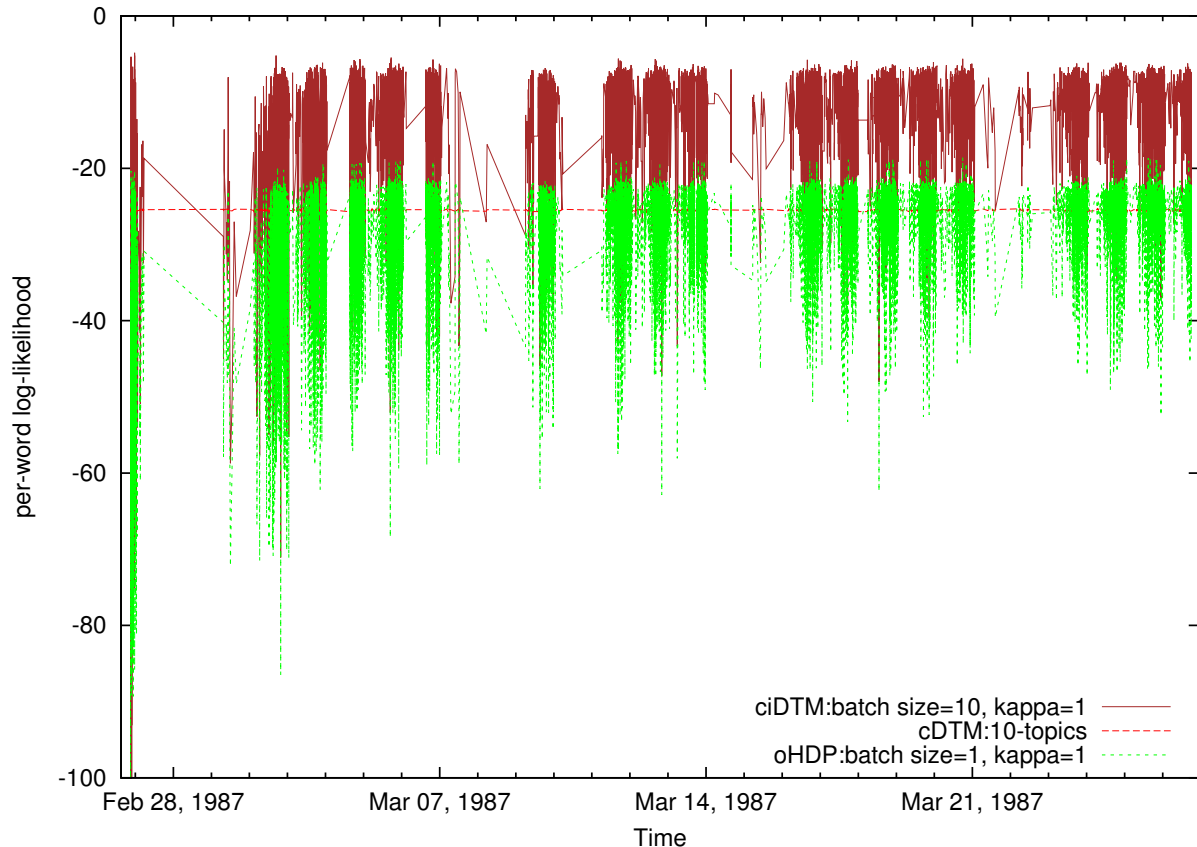


Figure 6.5: Comparison of the per-word log-likelihood for *cDTM*, *oHDP* and *ciDTM* using 10,000 news stories from the Reuters21578 corpus [47]. Higher per-word log-likelihood indicate a better fit of the model to the data.

cDTM number of topics The *cDTM* model was trained on a set of 5,000 documents sampled uniformly from the 10,000 document corpus. The trained model was then used to infer the documents’ topic mixture and the topics’ word mixture for the other half of the corpus. Figure 6.6 shows the per-word log-likelihood value obtained using different values for the number of topics.

The per-word log-likelihood performance of this model improved as the number of topics used increased. This improvement reached its peak with a model of 200 topics. Higher number of topics hurt the model performance as can be seen in the 250 topics model case which performs slightly worse than the 200 topics model. The reason this model reached

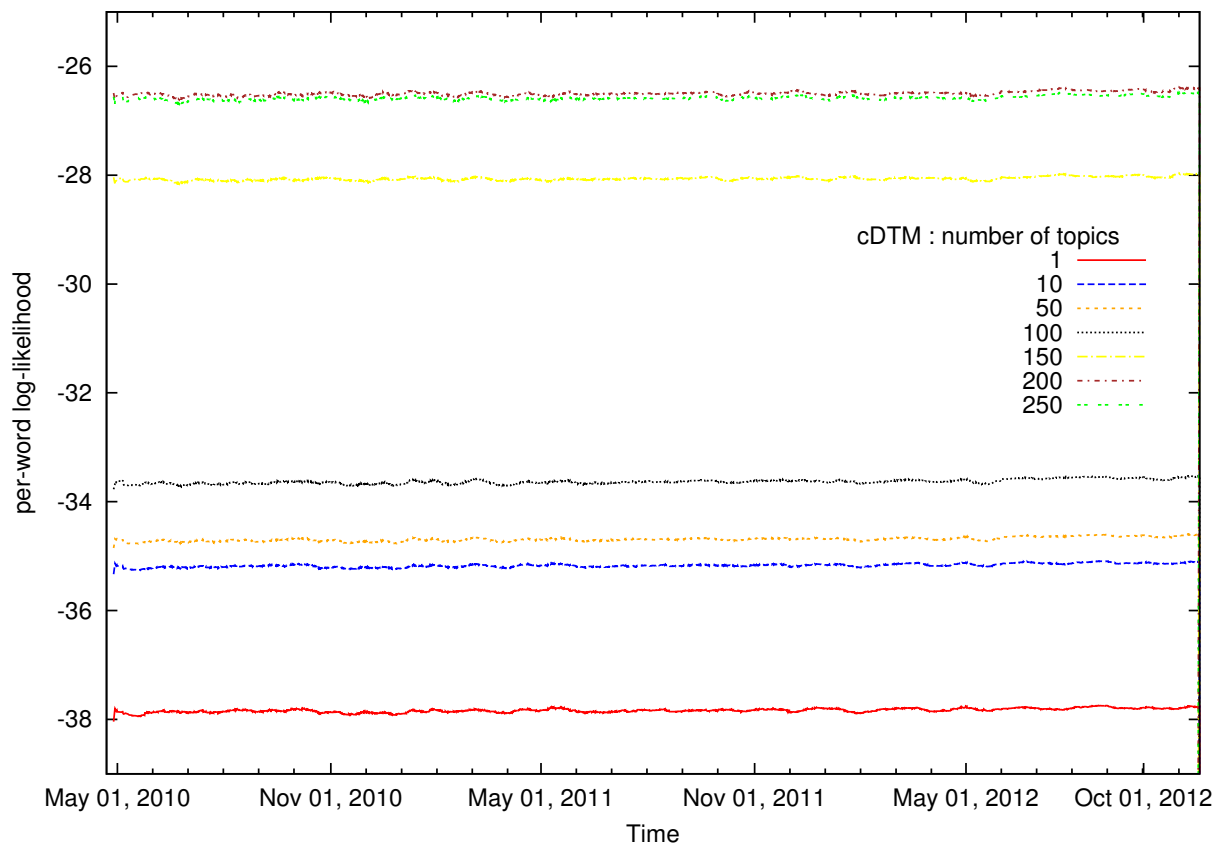


Figure 6.6: *cDTM* performance varies with different values for the number of topics using the BBC news corpus. A value of 200 leads to best model fit among the seven values tried. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.

its peak performance with 200 topics with this corpus while the same model reached a peak performance with 50 topics with the Reuters corpus can be explained by the properties of these two corpora. The BBC news corpus has a higher vocabulary richness and the relatively lengthy documents as compared to the Reuters news corpus. This BBC news corpus has a vocabulary of 64374 unique words and an average document length of 189.6 unique words. The Reuters news corpus has a vocabulary of 28569 unique words and an average document length of 58.4 unique words.

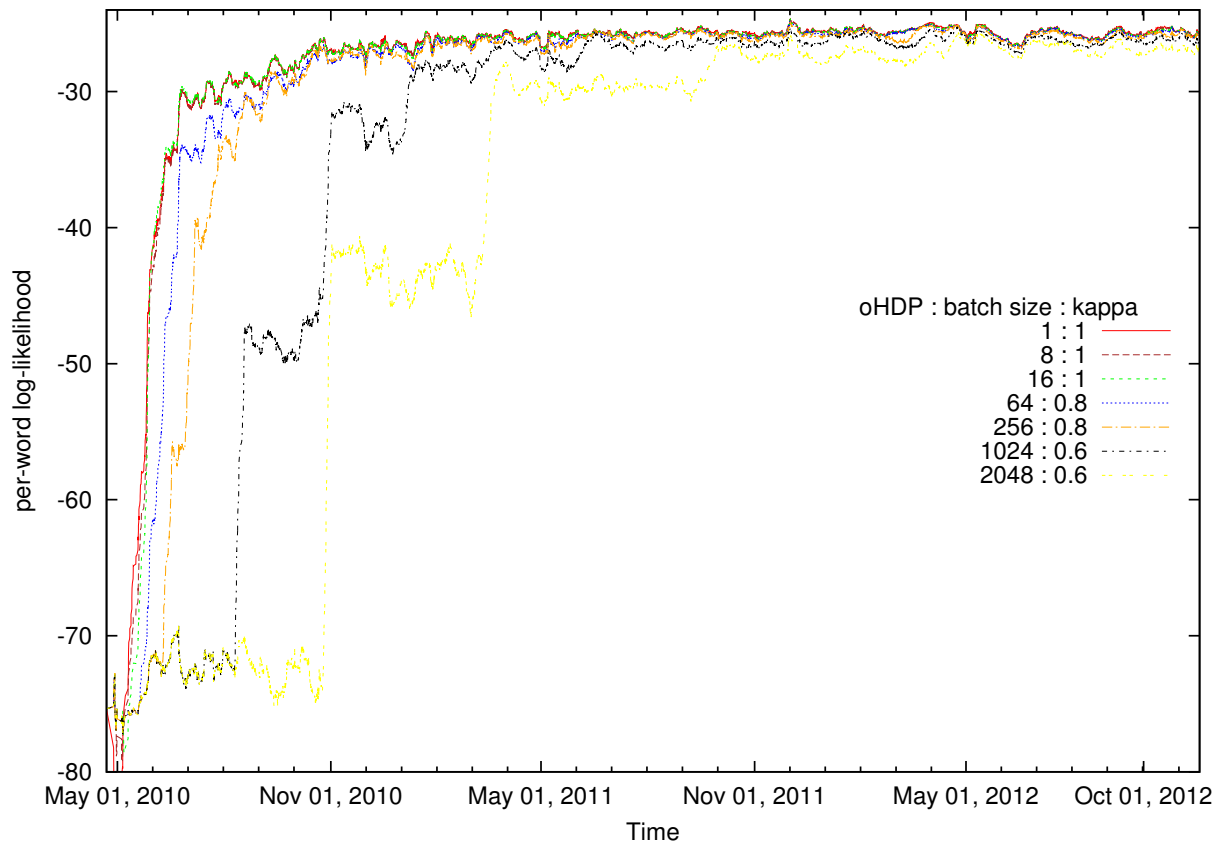


Figure 6.7: *oHDP* per-word log-likelihood for different batch size values using the BBC news corpus. A value of 200 leads to best model fit among the seven values tried. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.

oHDP batch size I experimented with different batch size and κ values for the oHDP model. I used the values suggested by Wang et al. [73] and other values that follow the same trend of these values. Wang et al. [73] found that higher κ values favor smaller batch sizes.

Figure 6.7 show the results of running these experiments. The trend obtained is similar to the one obtained with the Reuters corpus. The best per-word log-likelihood was obtained using a batch of size 1. As the batch size increased, the performance tended to consistently drop.

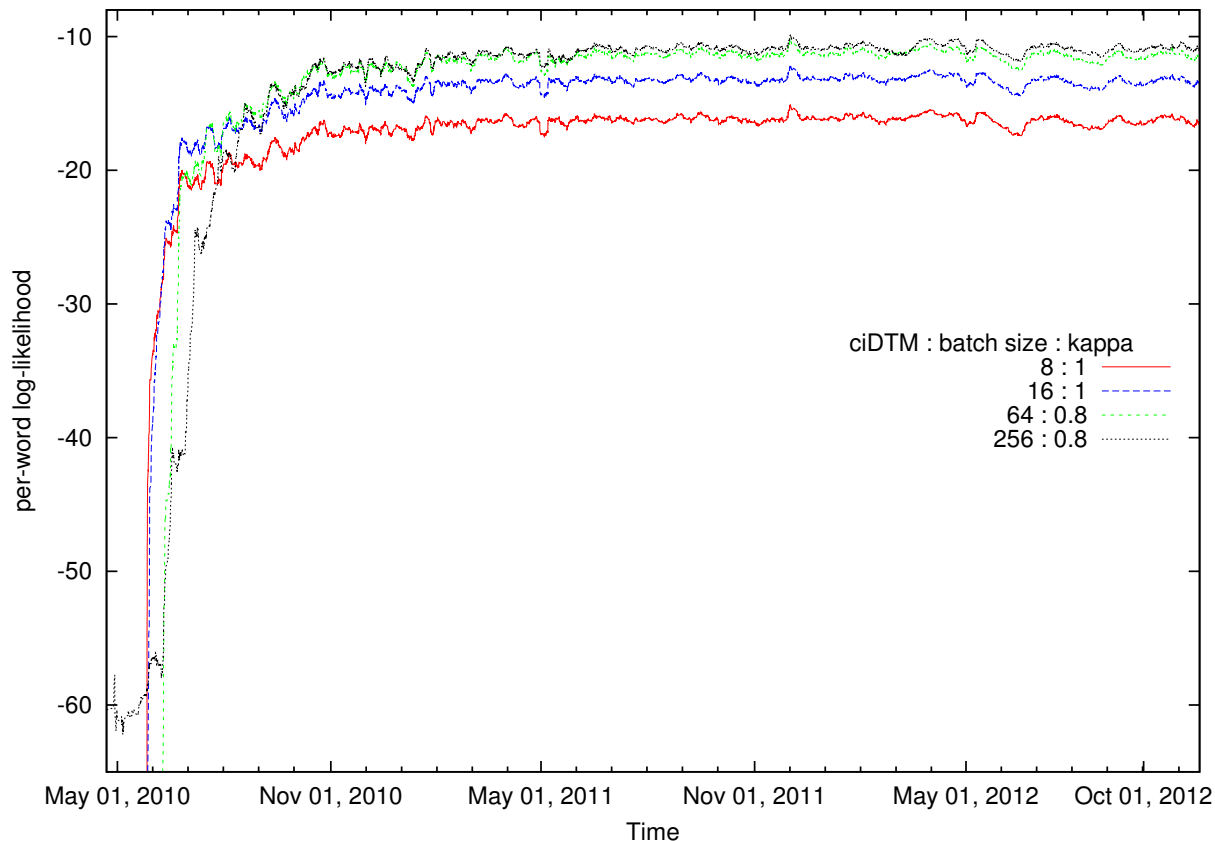


Figure 6.8: *ciDTM* per-word log-likelihood for different batch size values using the BBC news corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.

ciDTM batch size The size of the batch of documents the ciDTM model processes in each iteration affects the convergence of the inference algorithm and the variational Kalman filter used to evolve the per-topic word distribution in continuous time.

Figure 6.8 shows how the per-word log-likelihood performance of the model changes with different batch size values. The trend shown mirrors the one obtained using the Reuters corpus. The model performance improves as the batch size increases. A point of diminishing returns is reached around a value of 64 for the batch size. Minor performance gain obtained using higher batch size values is outweighed by the longer periods separating model updates.

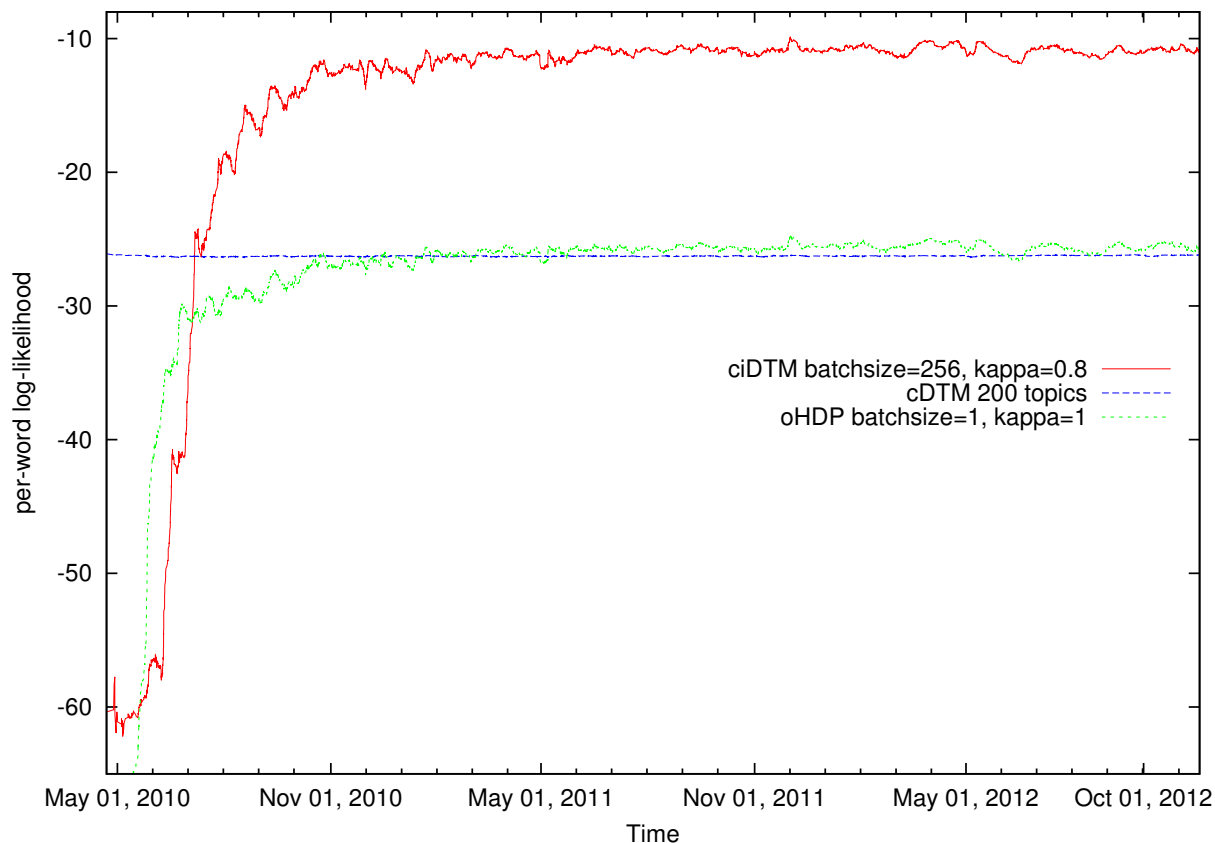


Figure 6.9: A comparison of *ciDTM*, *oHDP* and *cDTM* using their best setting values discovered earlier using the BBC news corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.

Comparing three models Figure 6.9 shows the per-word log-likelihood performance for the *cDTM*, *oHDP* and *ciDTM* models using their best setting values discovered by running the previous experiments. *ciDTM* outperforms both *oHDP* and *cDTM* by a big margin. The steady state performance of *cDTM* and *oHDP* is about the same. It took *oHDP* about 2,000 documents to reach this stage.

6.2.1 Why ciDTM outperforms cDTM

ciDTM outperforms cDTM because ciDTM evolves the number of topics per document while cDTM uses a fixed number of topics per document.

Limitations of cDTM Because cDTM uses a fixed and predefined number of topics per document, it does not allow for topic discovery. When cDTM infers the topic mixture of a document that discusses a new topic, it has to label the document with the predefined set of topics. The new topic will then get absorbed by that predefined set of topics and a new topic will *not* be learned. There will be no single topic that all the words in that document can be sampled from. This will directly reduce the per-word log-likelihood of sampling the words of the document given a topic which is the measure I use to evaluate the performance of the three competing models.

Advantage of ciDTM ciDTM evolves the number of topics per document to reflect the number of topics discovered so far in the corpus and to reflect the richness of the document. As more documents get processed by ciDTM, more topics will be discovered and their word distribution will be learned using the words in these documents. Discovering new topics instead of absorbing them into a set of predefined topics directly improves the per-word log-likelihood of sampling the words in a document given a single topic which is the measure I use to evaluate the performance of the three competing models. This is why ciDTM achieves a per-word log-likelihood value higher than that of cDTM.

6.2.2 Why ciDTM outperforms oHDP

ciDTM outperforms oHDP because ciDTM evolves the per-topic word distribution in continuous-time using document timestamps while oHDP only relies on document ordering and does not use document timestamps.

Advantage of ciDTM Because ciDTM knows how much time elapsed between documents that belong to the same topic, it evolves the topic word distribution accordingly to reflect the real change in that topic word distribution. The longer the time period separating two documents covering the same topic gets, the more drastic the change to the topic word distribution over that period will become. This direct proportional relationship between the length of the time period and the degree of change to the topic word distribution is plausible and can be seen in real life. The closeness of the inferred topic word distribution to the real topic word distribution translates to a higher per-word log-likelihood value for the model because it helps ciDTM better fit the model to the data.

Limitations of oHDP Since oHDP is unaware of the time period separating documents, its change to the topic word distribution is unaffected by the length of this time period. If this period gets big enough, the real topic word distribution will drastically change and oHDP will not change its topic word distribution to match it. This is because oHDP relies on document ordering to make this change and is not aware of the length of that time period. The big distance between the inferred topic word distribution and the real topic word distribution will lead to a lower per-word log-likelihood value for oHDP.

6.3 Scalability

A practical topic model system that infers the topic structure of news stories arriving in real time from a news outlet like a news agency should be able to process these documents in real time. All the three topic model systems I experimented with earlier, that is, the continuous-time dynamic topic model (cDTM), the online hierarchical Dirichlet process based topic model (oHDP), and the continuous-time infinite dynamic topic model (ciDTM), each one of them was able to train on the entire set of 10,000 documents in less than six hours. The topic inference process (the test phase of the model) in which the topic mixtures of news stories

arriving in real time from a news outlet takes much less time to finish than the training phase. The cDTM model was able to finish this task for a test set of 5,000 documents in less than 5 minutes. Because oHDP and ciDTM are online learners, the learning and inference phases alternated. The time spent running the test phase was included in the overall run time for these models and was not measured separately. By manual inspection I found that the time spent on the test phase is very negligible compared to the time spent on the training phase.

Given that a news agency like Agence France-Presse (AFP) releases on average 5,000 news stories per day that is equivalent to one news story every 20 seconds, and assuming that the running time complexity of all three systems is monotonically increasing, all the three systems presented can train on the entire set of documents released by AFP and then infer the topic mixture of a set of documents of the same size in real time.

Taking into account that the cDTM system is an offline learner, with every newly arriving document or set of documents it has to be retrained from scratch. As time passes by, the number of released documents that the model needs to train on increases linearly with time and at some point in the future the system will not be able to train on all released documents in time before new batch of documents that requires analysis arrives to the system from the news outlet. This lag will keep increasing linearly with time. To have the cDTM work in real time a compromise should be made. The system could maintain a limited history of the most recently released documents to train on. This history should be limited so that the cDTM system could train on it and then infer the topic mixture in real time. This solution will have a negative side effect on the performance of the system as the per-topic word distribution evolved by the system will not reflect the status of the per-topic word distribution of documents outside the limited history. The system will ignore older documents because it does not have enough time to train on it.

The oHDP and ciDTM systems are online learners. They can train on every news story arriving live from a news agency like AFP due to their fast online learning algorithm. They

can infer the topic mixture of news stories as their arrive to the system as well. The effect of the per-topic word distribution seen in a document will be retained by the system and carried on to affect the topic composition of documents arriving to the system in some arbitrary point in the future as long as the system is up and running processing documents. These status of these two systems will reflect the entire history of documents they received.

To illustrate the time requirements of the three systems and how they scale up I measured the wall clock time it takes each one of them to train on subsets of the Reuters corpus. I used subsets of size 1,000; 2,000; 5,000 and 10,000. The inference (testing) time was very negligible compared to training time (about 1 to 100 ratio) for the three systems. For each one of the three systems I used the settings that resulted in the best performance when experimented with the Reuters news corpus earlier. I used a cDTM model with 50-topics, an oHDP model with a batch size of 1 and a ciDTM model with a batch size of 256. The results are presented in Figure 6.10.

6.3.1 Scalability of ciDTM and oHDP vs cDTM

As shown in the figure, the cDTM does not scale up well with the size of the corpus. It becomes practically unfeasible to use it in real life applications with corpora of average size ($\sim 10,000$ documents). This growth in running time that is almost exponential can be attributed to the fixed number of topics in the model.

The reason cDTM does not scale up well with the size of the corpus is because the number of variables in the variational distribution over the latent variables is exponential in the number of topics and corpus size [72]. Since cDTM needs to tune the variational distribution variables to minimize the KL distance between the true posterior and the variational posterior, the time it needs to do so will grow linearly with the number of these variables which in turn grows exponentially with the size of the corpus.

On the other hand oHDP and ciDTM scale up well with the size of the corpus. This

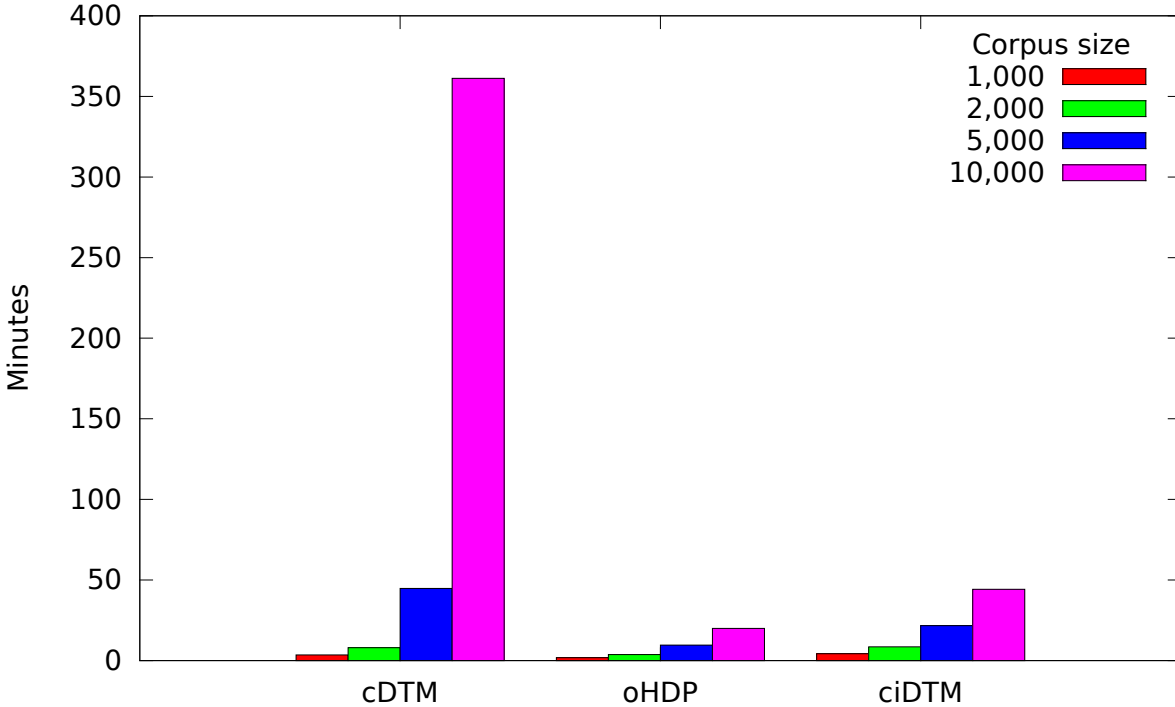


Figure 6.10: *The wall clock running time in minutes for cDTM (10 topics), oHDP (batch size = 1) and ciDTM (batch size = 256) using subsets of the Reuters news corpus with different number of documents. These systems were run on a PC with 16GB of RAM and 3.4 GHz processor.*

graceful scalability can be attributed to two factors: 1) the online variational inference algorithm that scales linearly with the number of documents, and 2) the number of topics that varies from one document to another. It is to be noted however that the oHDP based topic model is faster than the ciDTM. This is due to the extra time needed by ciDTM to run and sample from the Brownian motion model [54] that evolves the per-topic word distribution over time.

6.4 Timeline construction

For the timeline construction task, I manually selected 61 news stories from the BBC news corpus covering the Arab Spring events since its inception in January 2011 until the near

present (November 2012). I will call this set of news stories the Arab Spring set. Discovering such a topic is challenging as the events that fall under it are geographically scattered across the middle east and geographical names in the news stories will not give much clue to the topic model. More importantly, the events associated with this topic evolve rapidly over a short period of time and the set of vocabulary associated with it changes as well.

The cDTM is not suitable for this task. It assigns a fixed number of topics to every document in the corpus. If this number is high, the topic we are trying to discover will be split over more than one topic, and if the number is low, the topic will be merged with other topics. Moreover, since the number of topics typically vary from one document to another, finding one fixed value for it will always be inferior to other approaches that evolve it dynamically.

I trained the oHDP based topic model and the ciDTM on the entire BBC news corpus and inferred the topics associated with each document in the Arab Spring set. Then, I manually inspected the inferred topics and found the one that corresponds to the Arab Spring events. Both systems were trained using their best settings found in earlier experiments on the BBC news corpus in Section 6.2.

Figure 6.11 shows the inferred topics for documents in the Arab Spring set using oHDP. The notation I used in this Figure is described in its caption. After the start of the Arab Spring in early January of 2011, the documents discussing its events were assigned topic 3. This topic in the oHDP model corresponds to accidents and disasters. One month later, the model was able to infer the correct topic and associate it with the documents of the Arab Spring events. The performance of the system was steady from mid February 2011 until June 2012. After that, the topic evolved so fast after the Egyptian president assumed office in late June 2012. The oHDP was unable to evolve the word distribution associated with that topic quickly enough to keep track of the topic. We can see that after that date the model was unable to infer the correct topic for 3 documents in this set. The documents were associated with the accidents and disasters topic instead. Overall, the model was unable to

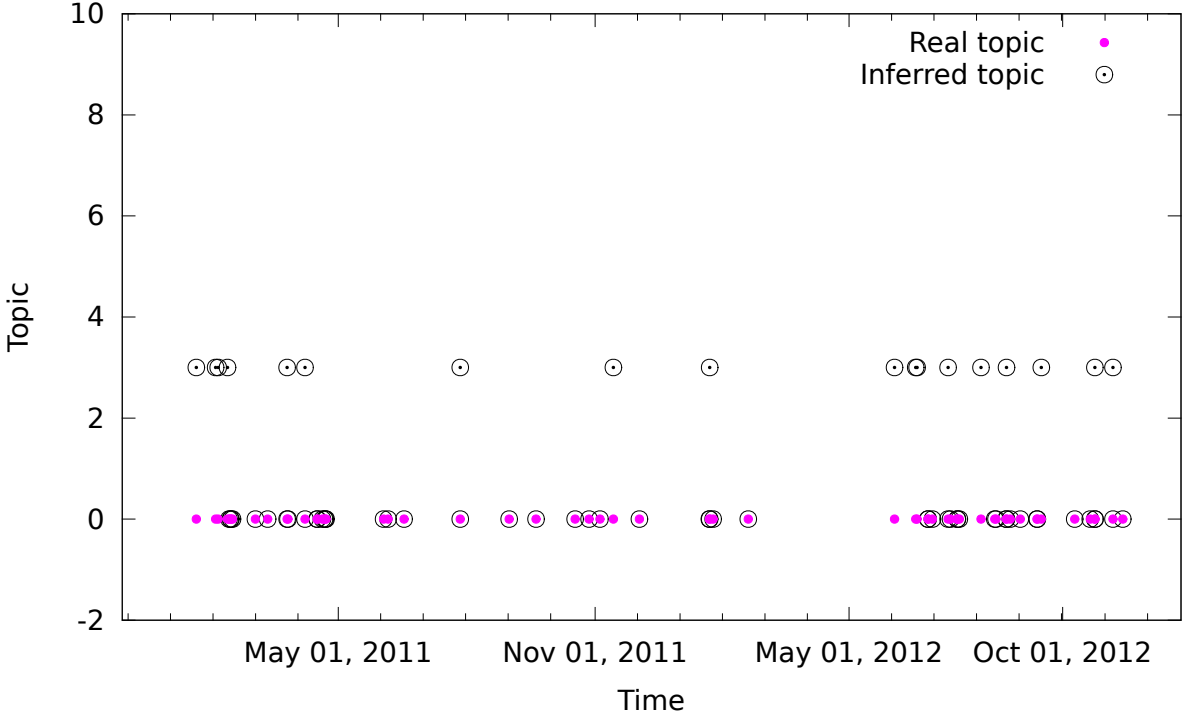


Figure 6.11: *oHDP* timeline construction for the Arab Spring topic over 61 documents covering this topic. The independent axis is the time, while the dependent axis is the topic index. It was found that topic 0 represents the Arab Spring events topic. A pink dot (\bullet) represents the real topic that should be discovered, a black circle with a small dot (\odot) represents the topic/s that was discovered for the document. A black circle with a pink dot in the middle of it (\odot) is a real topic that was successfully discovered by the system.

infer the correct topic for 10 out of 61 documents in the Arab Spring set.

		Inferred		Accuracy
		AS	non-AS	
True	AS	51	10	0.836
	non-AS	0	13	1.000
				0.865

Table 6.3: *Confusion matrix for timeline construction using oHDP. AS stands for Arab Spring topic, and non-AS is non Arab Spring topic. The true class is presented in rows (True AS, and non-AS), and Inferred class in columns (Inferred AS and non-AS).*

Table 6.3 shows the confusion matrix for the oHDP topic assignment to news stories belonging to the Arab Spring timeline. oHDP failed to correctly tag 10 out of 61 stories as

belonging to that topic. However, it successfully labeled all 13 stories which had non Arab Spring topic components with the correct topic label. This gives oHDP a recall score of 83.6% and a precision of 100%.

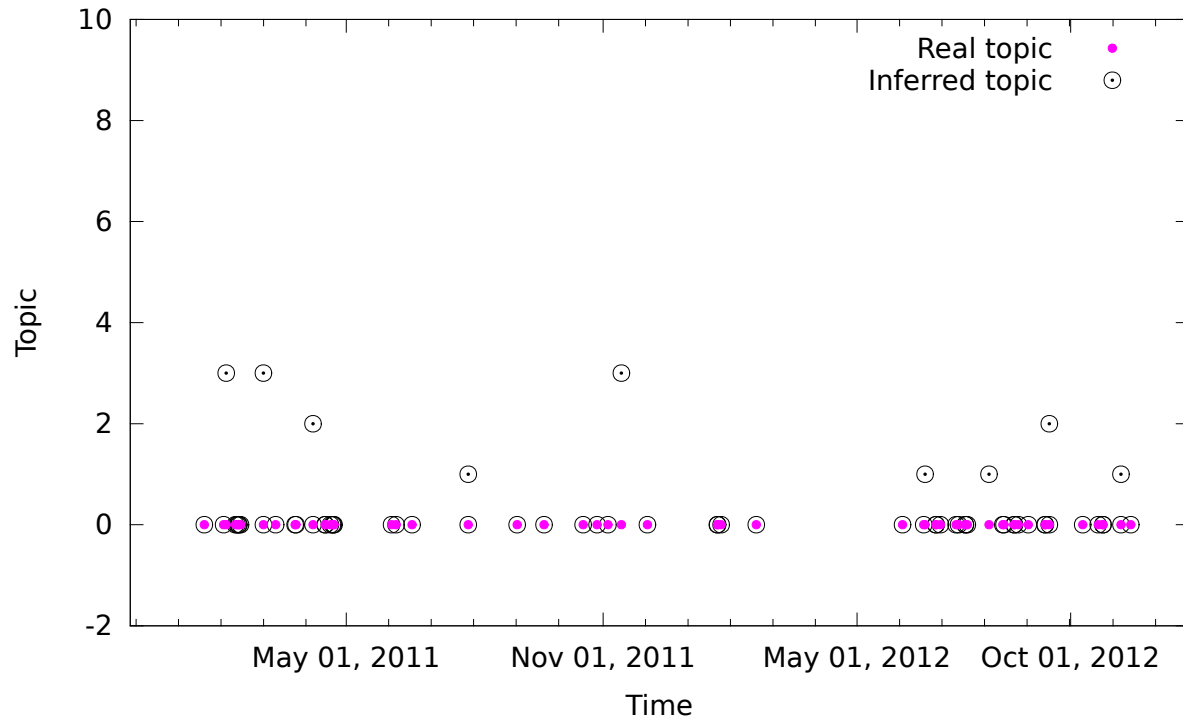


Figure 6.12: *ciDTM* timeline construction for the Arab Spring topic over 61 documents covering this topic. The independent axis is the time, while the dependent axis is the topic index. It was found that topic 0 represents the Arab Spring events topic. A pink dot (\bullet) represents the real topic that should be discovered, a black circle with a small dot (\odot) represents the topic/s that was discovered for the document. A black circle with a pink dot in the middle of it (\odot) is a real topic that was successfully discovered by the system.

Figure 6.12 shows the inferred topics for documents in the Arab Spring set using *ciDTM*. This Figure uses the same notation as the previous Figure and it is described in its caption also. Topic 0 represents the Arab Spring topic in this model. At first glance at the Figure, we notice that the *ciDTM* was able to infer the correct topic for the first document in the Arab Spring set. This can be explained by the fact that *ciDTM* trains on a batch of 256 documents. The model was able to train on a big batch of documents that included some Arab Spring set documents and learn the new topic word distribution before inferring their

topics. If that explanation is valid then this advantage is not intrinsic of the ciDTM as the oHDP based model could behave similarly by increasing its batch size.

We notice that some documents in this set were assigned multiple topics with this model. Besides topic 0, which corresponds to the Arab Spring events, topics 1, 2 or 3 sometimes appear together with topic 0 in the same document. By inspecting these topics, I found that topic 1 corresponds to economy and finance, topic 2 corresponds to health and medicine, and topic 3 corresponds to accidents and disasters. Some of the early documents in the Arab Spring set were assigned topic 3. This reflects the violence that marred the early days of the Egyptian revolution in late January 2011. The association of topic 2 with the set documents may reflect the mention of the injured protesters at that time. Later documents in this set are more associated with topic 1. This can be explained by the volume of published news stories discussing the Egyptian government efforts to recover from the economical and financial damages which the revolution inflicted on the country.

		Inferred		Accuracy
		AS	non-AS	
True	AS	57	4	0.934
	non-AS	0	13	1.000
				0.946

Table 6.4: *Confusion matrix for timeline construction using ciDTM. AS stands for Arab Spring topic, and non-AS is non Arab Spring topic. The true class is presented in rows (True AS, and non-AS), and Inferred class in columns (Inferred AS and non-AS).*

Table 6.4 shows the confusion matrix for the ciDTM topic assignment to news stories belonging to the Arab Spring timeline. ciDTM shows about 10% improvement in accuracy over the oHDP based topic model due to a 10% higher true positive rate. ciDTM achieves a recall score of 93.4% and a 100% precision. ciDTM has a higher recall power without sacrificing precision.

6.4.1 Why ciDTM is better than oHDP in timeline construction

ciDTM is better than oHDP in timeline construction because ciDTM evolves the per-topic word distribution in continuous time while oHDP only relies on document ordering and does not make use of document timestamps in evolving the per-topic word distribution.

This can be seen by comparing the timelines constructed for the Arab Spring topic using oHDP based topic model in Figure 6.11 and using ciDTM in Figure 6.12.

oHDP and ciDTM were able to detect the birth of a new topic, which is the Arab Spring topic, soon after it started in January of 2011. The true positive rate was high for both models (oHDP=0.91, ciDTM=0.94) in the first part of this timeline (from January 2011 to March 2012). In the second part of this timeline (from June 2012 to November 2012) the true positive rate for oHDP sharply dropped to a value of 0.74 while it slightly dropped to a value of 0.92 for ciDTM.

In what follows I explain why ciDTM maintained a high true positive rate value in the second part of the Arab Spring topic timeline while the true positive rate for oHDP dropped sharply.

The corpus used to train both models had no documents with the Arab Spring topic in the period of March 2012 to June 2012.

Limitations of oHDP Because oHDP evolves the per-topic word distribution based on document ordering and does not use document timestamps for that it was unaware of the long time period (3 months) that separated the last document in the first part of the timeline and the first document in the second part of the timeline. Because of that, oHDP did not evolve the Arab Spring topic word distribution to reflect the changes in vocabulary of the news stories covering this topic. The actual word distribution for the Arab Spring topic significantly changed and the oHDP did not change its Arab Spring topic word distribution

to match it. This is because oHDP was unaware of the long time period (March 2012 to June 2012 [3 months]) in which no documents covering this topic were included in the corpus. Because of the big difference between the actual Arab Spring topic word distribution and the oHDP Arab Spring topic word distribution, oHDP failed to correctly label many news stories with the Arab Spring topic in the second part of the timeline after the long period of topic dormancy. This mislabeling resulted in a sharp drop in the true positive rate value for oHDP in the second part of the timeline when compared to the true positive rate value it achieved in the first part.

Advantage of ciDTM On the other hand ciDTM uses document timestamps to evolve the per-topic word distribution. When ciDTM infers the topic mixture for the first document in the second part of the Arab Spring timeline (June 2012 to November 2012) it evolves the Arab Spring topic word distribution to reflect the long period (March 2012 to June 2012 [3 months]) of dormancy of this topic. The word distribution of the Arab Spring topic evolved by ciDTM to reflect this long time period of dormancy is closer to the actual Arab Spring topic. Because of this closeness, ciDTM was able to correctly label news stories with the Arab Spring topic in the second part of the timeline. This successful labeling resulted in a slight drop in the true positive rate for ciDTM in the second part of the Arab Spring timeline when compared to the true positive rate value it achieved in the first part of the timeline.

Bibliography

- [1] Amr Ahmed and Eric P. Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *SDM*, pages 219–230. SIAM, 2008. URL http://www.siam.org/proceedings/datamining/2008/dm08_20_Ahmed.pdf.
- [2] Amr Ahmed and Eric P. Xing. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. In Peter Grünwald and Peter Spirtes, editors, *UAI*, pages 20–29, Corvallis, Oregon, 2010. AUAI Press. ISBN 978-0-9749039-6-5. URL <http://uai.sis.pitt.edu/papers/10/p20-ahmed.pdf>.
- [3] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d’Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. Mach. Learn. Res.*, 9:485–516, June 2008. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1390681.1390696>.
- [4] A.L. Barabási, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. *Physica A: Statistical Mechanics and its Applications*, 272(1): 173–187, 1999. URL http://www.barabasilab.com/pubs/CCNR-ALB_Publications/199910-01_PhysA-Meanfield/199910-01_PhysA-Meanfield.pdf.
- [5] Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta, editors. *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*, 2009. Curran Associates, Inc. ISBN 9781615679119. URL <http://nips2009.topicmodels.net>.
- [6] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Sci-*

- ence and Statistics*). Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007. ISBN 0387310738. URL <http://www.worldcat.org/isbn/0387310738>.
- [7] D. Blackwell and J. B. Macqueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1:353–355, 1973. URL <http://www.stat.ucla.edu/~macqueen/PP12.pdf>.
- [8] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL <http://www.cs.princeton.edu/~blei/papers/Blei2012.pdf>.
- [9] David M. Blei and Michael I. Jordan. Variational methods for the dirichlet process. In *Proceedings of the twenty-first international conference on Machine learning, ICML '04*, pages 12–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015439. URL www.cs.berkeley.edu/~jordan/papers/vdp-icml.pdf.
- [10] David M. Blei and John D. Lafferty. Correlated topic models. In *NIPS*, 2005. URL http://books.nips.cc/papers/files/nips18/NIPS2005_0774.pdf.
- [11] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning, ICML '06*, pages 113–120, New York, NY, USA, 2006. ACM. URL <http://www.cs.princeton.edu/~blei/papers/BleiLafferty2006a.pdf>.
- [12] David M. Blei and John D. Lafferty. Topic models. In Mehran Sahami and Ashok Srivastava, editors, *Text Mining: Classification, Clustering, and Applications*. Taylor and Francis Group, 2009. URL <http://www.cs.princeton.edu/~blei/papers/BleiLafferty2009.pdf>.
- [13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. ISSN 1532-4435. URL <http://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf>.

- [14] N. E. Breslow and D. G. Clayton. Approximate Inference in Generalized Linear Mixed Models. *Journal of the American Statistical Association*, 88(421):9–25, 1993. URL <http://www.public.iastate.edu/~alicia/stat544/Breslow%20and%20Clayton%201993.pdf>.
- [15] Wray L. Buntine. Estimating likelihoods for topic models. In *Asian Conference on Machine Learning*, 2009. URL http://www.nicta.com.au/___data/assets/pdf_file/0019/20746/sdca-0202.pdf.
- [16] Wray L. Buntine and Aleks Jakulin. Discrete component analysis. In *SLSFS*, volume 3940 of *Lecture Notes in Computer Science*, pages 1–33. Springer, 2005. ISBN 3-540-34137-4. URL <http://cosco.hiit.fi/Articles/buntineBohinj.pdf>.
- [17] Mike Cafarella and Doug Cutting. Building nutch: Open source search. *Queue*, 2(2):54–61, April 2004. ISSN 1542-7730. doi: 10.1145/988392.988408. URL <http://doi.acm.org/10.1145/988392.988408>.
- [18] Asli Celikyilmaz. A semantic question / answering system using topic models. In Bengio et al. [5]. ISBN 9781615679119. URL http://www.umiacs.umd.edu/~jbg/nips_tm_workshop/16.pdf.
- [19] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 2009. URL <http://www.umiacs.umd.edu/~jbg/docs/nips2009-rtl.pdf>.
- [20] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, 1990. ISSN 0004-3702. doi: DOI:10.1016/0004-3702(90)90060-D. URL http://bmir.stanford.edu/file_asset/index.php/644/BMIR-1990-0317.pdf.
- [21] Robert Darnton. Google and the future of books. Online, 2009.

- [22] Pradipto Das and Rohini Srihari. Learning to summarize using coherence. In Bengio et al. [5]. ISBN 9781615679119. URL <http://www.cedar.buffalo.edu/~rohini/Papers/pdasNIPS09Wkshp.pdf>.
- [23] P.A.M. Dirac. *The principles of quantum mechanics*. Clarendon press, 1992. ISBN 9780198520115. URL <http://books.google.com/books?id=svhAAQAIAAJ>.
- [24] Gabriel Doyle and Charles Elkan. Financial topic models. In Bengio et al. [5]. ISBN 9781615679119. URL <http://idiom.ucsd.edu/~gdoyle/papers/doyle-elkan-2009-nips-paper.pdf>.
- [25] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA, 2005. Association for Computational Linguistics. doi: <http://dx.doi.org/10.3115/1219840.1219885>. URL <http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf>.
- [26] Eugene C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *AAAI'90: Proceedings of the eighth National conference on Artificial intelligence*, pages 4–9. AAAI Press, 1990. ISBN 0-262-51057-X. URL [http://rna-informatics.uga.edu/Readings/tertiary/Complexity%20of%20K-Tree%20Structured%20Constraint%20Satisfaction%20Problems%20\(Freuder1990\).pdf](http://rna-informatics.uga.edu/Readings/tertiary/Complexity%20of%20K-Tree%20Structured%20Constraint%20Satisfaction%20Problems%20(Freuder1990).pdf).
- [27] Subir Ghosh, Kenneth P. Burnham, Nico F. Laubscher, Gerard E. Dallal, Leland Wilkinson, Donald F. Morrison, Milton W. Loyer, Bennett Eisenberg, Solomon Kullback, Ian T. Jolliffe, and Jeffrey S. Simonoff. Letters to the editor. *The American Statistician*, 41(4):pp. 338–341, 1987. ISSN 00031305. URL <http://www.jstor.org/stable/2684769>.

- [28] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. Citeseer: an automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, DL '98, pages 89–98, New York, NY, USA, 1998. ACM. ISBN 0-89791-965-3. doi: 10.1145/276675.276685. URL <http://clgiles.ist.psu.edu/papers/DL-1998-citeseer.pdf>.
- [29] W.R. Gilks. Markov chain monte carlo. *Encyclopedia of Biostatistics*, 2005.
- [30] M. Girolami. A variational method for learning sparse and overcomplete representations. *Neural computation*, 13(11):2517–2532, 2001.
- [31] Stefan Gradmann. Knowledge = information in context: on the importance of sematic contextualisation in europeana. Europeana, 2010. URL http://www.europeanconnect.eu/europeanatech/documents/Gradmann_Stefan_LinkedOpenEuropeana.pdf.
- [32] Heather Green. A library as big as the world. online, 2002.
- [33] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *PNAS*, 101(suppl. 1):5228–5235, 2004. URL <http://www.pnas.org/content/101/suppl.1/5228.full.pdf>.
- [34] David Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft research, Redmond, WA, March 1995. URL <http://research.microsoft.com/pubs/69588/tr-95-06.pdf>.
- [35] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM. ISBN 1-58113-096-1. doi: 10.1145/312624.312649. URL <http://www.cs.brown.edu/~th/papers/Hofmann-UAI99.pdf>.

- [36] Diane Hu and Lawrence Saul. A probabilistic topic model for music analysis. In Bengio et al. [5]. ISBN 9781615679119. URL http://cseweb.ucsd.edu/~dhu/docs/nips09_abstract.pdf.
- [37] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999. URL <http://www.cs.berkeley.edu/~jordan/papers/variational-intro.pdf>.
- [38] Kalman, Rudolph, and Emil. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960. URL <http://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf>.
- [39] Samuel Kim, Shiva Sundaram, Panayiotis Georgiou, and Shrikanth Narayanan. Audio scene understanding using topic models. In Bengio et al. [5]. ISBN 9781615679119. URL <http://sail.usc.edu/aigaion2/index.php/attachments/single/372>.
- [40] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7(4):373–397, October 2003. ISSN 1384-5810. doi: 10.1023/A:1024940629314. URL <http://www.cs.cornell.edu/home/kleinber/bhs.pdf>.
- [41] D. Koller, U. Lerner, and D. Angelov. A general algorithm for approximate inference and its application to hybrid bayes nets. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 324–333. Morgan Kaufmann Publishers Inc., 1999. URL <http://robotics.stanford.edu/~uri/Papers/uai99.ps>.
- [42] D. Koller, N. Friedman, L. Getoor, and B. Taskar. Graphical models in a nutshell. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.
- [43] S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. URL http://www.csee.wvu.edu/~xinl/library/papers/math/statistics/Kullback_Leibler_1951.pdf.

- [44] Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed variational dirichlet process mixture models. In *IJCAI*, pages 2796–2801, 2007. URL <http://dli.iiit.ac.in/ijcai/IJCAI-2007/PDF/IJCAI07-449.pdf>.
- [45] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *ICML*, pages 282–289. Morgan Kaufmann, 2001. ISBN 1-55860-778-1. URL <http://www.cis.upenn.edu/~pereira/papers/crf.pdf>.
- [46] Thomas K Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284, 1998. doi: 10.1080/01638539809545028. URL <http://lsa.colorado.edu/papers/dp1.LSAintro.pdf>.
- [47] David D. Lewis. Reuters-21578.
- [48] Michael Ley. The dblp computer science bibliography: Evolution, research issues, perspectives. In *Proceedings of the 9th International Symposium on String Processing and Information Retrieval*, SPIRE 2002, pages 1–10, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44158-1. URL <http://dl.acm.org/citation.cfm?id=646491.694954>.
- [49] Erik Linstead, Lindsey Hughes, Cristina Lopes, and Pierre Baldi. Software analysis with unsupervised topic models. In Bengio et al. [5]. ISBN 9781615679119. URL http://www.umiacs.umd.edu/~jbg/nips_tm_workshop/5.pdf.
- [50] David J. C. Mackay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 1st edition, June 2003. ISBN 0521642981. URL <http://www.cs.toronto.edu/~mackay/itprnn/book.pdf>.
- [51] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceed-*

- ings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 188–191, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: <http://dx.doi.org/10.3115/1119176.1119206>. URL <http://www.cs.umass.edu/~mccallum/papers/mccallum-conll2003.pdf>.
- [52] Andrew McCallum and Charles Sutton. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006. URL <http://www.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf>.
- [53] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [54] Peter Mörters and Yuval Peres. *Brownian motion*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2010. ISBN 978-0-521-76018-8. URL <http://www.stat.berkeley.edu/~peres/bmbook.pdf>. With an appendix by Oded Schramm and Wendelin Werner.
- [55] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, UAI’99, pages 467–475, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-614-9. URL http://www.cs.ubc.ca/~murphyk/Papers/loopy_uai99.pdf.
- [56] U. Nodelman, C.R. Shelton, and D. Koller. Continuous time bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 378–387, 2002. URL <http://www.cs.ucr.edu/~cshelton/papers/docs/ctbn.pdf>.
- [57] Tamara Polajnar and Mark Girolami. Application of lexical topic models to protein

- interaction sentence prediction. In Bengio et al. [5]. ISBN 9781615679119. URL <http://eprints.pascal-network.org/archive/00005815/01/9.pdf>.
- [58] John W Pratt. on the efficiency of maximum likelihood estimation. *Annals of Statistics*, 4:501–514, 1976. ISSN 3. URL http://projecteuclid.org/DPubS/Repository/1.0/Disseminate?view=body&id=pdf_1&handle=euclid.aos/1176343457.
- [59] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-015157-2.
- [60] John A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 3 edition, April 2001. ISBN 0534399428.
- [61] Purnamrita Sarkar and Andrew W. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7(2):31–40, December 2005. ISSN 1931-0145. doi: 10.1145/1117454.1117459. URL http://www.cs.cmu.edu/~psarkar/NIPS2005_0724.pdf.
- [62] Masa-Aki Sato. Online model selection based on the variational bayes. *Neural Comput.*, 13(7):1649–1681, July 2001. ISSN 0899-7667. URL http://www.robots.ox.ac.uk/~parg/mlrg/papers/online_variational.pdf.
- [63] D.F. Shanno et al. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970. URL <http://www.ams.org/journals/mcom/1970-24-111/S0025-5718-1970-0274029-X/S0025-5718-1970-0274029-X.pdf>.
- [64] Xiaodan Song, Ching-Yung Lin, Belle L. Tseng, and Ming-Ting Sun. Modeling and predicting personal information dissemination behavior. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, The 11th ACM SIGKDD International conference on Knowledge Discovery and Data Min-

- ing (KDD), pages 479–488, New York, NY, USA, 2005. ACM. ISBN 1-59593-135-X. doi: 10.1145/1081870.1081925. URL <http://wing.comp.nus.edu.sg/downloads/keyphraseCorpus/138/138.pdf>.
- [65] Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *J. Mach. Learn. Res.*, 8:693–723, 2007. ISSN 1532-4435. URL <http://people.cs.umass.edu/~mccallum/papers/dcrf-icml04.pdf>.
- [66] Russell Swan and David Jensen. Timemines: Constructing timelines with statistical models of word usage. In *The sixth ACM SIGKDD international conference on Knowledge Discovery and Datamining*, pages 73–80, 2000. URL http://www.cs.cmu.edu/~dunja/KDDpapers/Swan_TM.pdf.
- [67] J. Taylor. Jstor: an electronic archive from 1665. *Notes and Records of the Royal Society of London*, 55(1):179–181, 2001. doi: 10.1098/rsnr.2001.0135. URL <http://rsnr.royalsocietypublishing.org/content/55/1/179.abstract>.
- [68] 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. URL <http://www.cs.berkeley.edu/~jordan/papers/hdp.pdf>.
- [69] Yee Whye Teh, Kenichi Kurihara, and Max Welling. Collapsed variational inference for hdp. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *NIPS*. MIT Press, 2007. URL http://books.nips.cc/papers/files/nips20/NIPS2007_0763.pdf.
- [70] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008. ISSN 1935-8237. doi: 10.1561/22000000001. URL http://www.eecs.berkeley.edu/~wainwrig/Papers/WaiJor08_FTML.pdf.

- [71] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 1105–1112, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553515. URL <http://people.cs.umass.edu/~wallach/talks/evaluation.pdf>.
- [72] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. In David A. Mcallester and Petri Myllymäki, editors, *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI)*, 2008. URL http://uai2008.cs.helsinki.fi/UAI_camera_ready/wang.pdf.
- [73] Chong Wang, John William Paisley, and David M. Blei. Online variational inference for the hierarchical dirichlet process. *Journal of Machine Learning Research - Proceedings Track*, 15:752–760, 2011. URL <http://www.jmlr.org/proceedings/papers/v15/wang11a/wang11a.pdf>.
- [74] Xuerui Wang and Andrew 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, KDD '06*, pages 424–433, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. doi: 10.1145/1150402.1150450. URL http://www.braddock.com/Topics_over_Time_A_Non_Markov_Continuous_Time_Model_of_Topical_Trends.pdf.
- [75] Xuerui Wang, Natasha Mohanty, and Andrew McCallum. Group and topic discovery from relations and text. In *Proceedings of the 3rd international workshop on Link discovery, LinkKDD '05*, pages 28–35, New York, NY, USA, 2005. ACM. ISBN 1-59593-215-1. doi: 10.1145/1134271.1134276. URL http://www.cs.umass.edu/~mccallum/papers/grouptopic_linkkdd05.pdf.
- [76] E.P. Xing, M.I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the Nineteenth conference*

on *Uncertainty in Artificial Intelligence*, pages 583–591. Morgan Kaufmann Publishers Inc., 2002. URL <http://www.gatsby.ucl.ac.uk/publications/papers/06-2000.pdf>.

[77] Eric P Xing. On topic evolution. Technical Report CMU-CALD-05-115, Carnegie Mellon University, Pittsburgh, PA, USA, Dec 2005. URL <http://www.cs.cmu.edu/~epxing/papers/TR-115.pdf>. TDT 2004 workshop notes.

[78] Raymond W. Yeung. *Information Theory and Network Coding*. Springer Publishing Company, Incorporated, 1 edition, 2008. ISBN 0387792333, 9780387792330.

Appendix A

Probability distributions

In this Appendix, I briefly introduce some of the most important probability distributions in Bayesian data modeling.

A.1 Uniform

This is a simple distribution for a continuous time variable x . The probability that this variable takes any value in its finite domain $([a, b])$ is constant and can be defined by:

$$U(x|a, b) = \frac{1}{b - a} \tag{A.1}$$

A.2 Bernoulli

This distribution is for a single binary variable $x \in \{0, 1\}$, can be thought of as representing the result of flipping a coin. The parameter $\mu \in [0, 1]$ gives the probability of success, which is having $x = 1$. More formally:

$$\text{Bern}(x|\mu) = \mu^x(1 - \mu)^{1-x} \tag{A.2}$$

A.3 Binomial

If a Bernoulli trial (such as coin flipping) is repeated N times, where the probability of success is $\mu \in [0, 1]$, then the probability distribution of the number of successes (x) is given by:

$$\text{Bin}(x|\mu, N) = \binom{N}{x} \mu^x (1 - \mu)^{N-x} \quad (\text{A.3})$$

A.4 Multinomial

The multinomial distribution is a generalization of the binomial distribution where the Bernoulli distribution giving the probability of success is replaced with a categorical distribution. In this case, there are K possible outcomes for the discrete variable $\mathbf{x} = \{x_1 \dots, x_K\}$ with probabilities $\mu = \{\mu_1, \dots, \mu_K\}$ such that $\sum_k \mu_k = 1$.

For N observations, if $\mathbf{m} = \{m_1, \dots, m_K\}$, where m_k is the number of times outcome x_k is observed. Then, \mathbf{m} follows a multinomial distribution with parameters μ and N . This distribution can be defined as follows:

$$\text{Mult}(\mathbf{m}|\mu, N) = \frac{N!}{m_1! \dots m_K!} \prod_{k=1}^K \mu_k^{m_k} \quad (\text{A.4})$$

The categorical and multinomial distributions are often conflated in computer science literature. The outcome of a categorical distribution is sometimes represented as a binary 1-of- K vector, indicating which one of the K outcomes was observed. In this notation, which is adopted by Bishop [6, p. 690], the categorical distribution represents a multinomial distribution over a single observation.

A.5 Beta

This is a distribution over a continuous variable $x \in [0, 1]$ and is often used to represent the probability of a binary event with two parameters $a > 0$ and $b > 0$. This Beta distribution

is the conjugate prior for the Bernoulli distribution where a and b can be interpreted as the effective prior number of success and failure observations, respectively. This distribution can be defined by:

$$\text{Beta}(x|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} \quad (\text{A.5})$$

Where $\Gamma(\cdot)$ is the Gamma function given by:

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du \quad (\text{A.6})$$

$$= (x-1)!, \quad \text{for integer } x \quad (\text{A.7})$$

A.6 Dirichlet

The Dirichlet distribution is a multivariate distribution over K random variables $0 \leq \mu_k \leq 1$, where $k = 1, \dots, K$,

$$\text{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = C(\boldsymbol{\alpha}) \prod_{k=1}^K \mu_k^{\alpha_k-1} \quad (\text{A.8})$$

where

$$C(\boldsymbol{\alpha}) = \frac{\Gamma(\hat{\alpha})}{\Gamma(\alpha_1) \dots \Gamma(\alpha_K)} \quad (\text{A.9})$$

and

$$\hat{\alpha} = \sum_{k=1}^K \alpha_k. \quad (\text{A.10})$$

Subject to the constraints:

$$0 \leq \mu_k \leq 1 \quad (\text{A.11})$$

$$\sum_{k=1}^K \mu_k = 1. \quad (\text{A.12})$$

The Dirichlet distribution forms a conjugate prior with the multinomial distribution and is a generalization of the Beta distribution. The parameters α_k can be interpreted as the effective number of observations of the corresponding values of the K-dimensional observation vector \mathbf{x} .

Appendix B

Dirichlet Process

The Dirichlet Process (DP) is a distribution over distributions, meaning that a draw from the process gives us a Dirichlet distribution. A DP denoted by $DP(G_0, \alpha)$ is parametrized by a base measure G_0 and a concentration parameter α which is analogous to the mean and variance of a Gaussian distribution, respectively. By integrating out G , θ follows a Polya urn distribution [7] or a Chinese Restaurant Process [2]

$$\theta_i | \theta_{1:i-1}, G_0, \alpha \sim \sum_k \frac{m_k}{i-1+\alpha} \delta(\phi_k) + \frac{\alpha}{i-1+\alpha} G_0 \quad (\text{B.1})$$

where, $\phi_{1:k}$ are the distinct values of θ , and m_k is the number of parameter θ with ϕ_k value. A Dirichlet process mixture model (DPM) can be built using the given DP on top of a hierarchical Bayesian model.

Instead of having a fixed Dirichlet distribution to draw a parameter vector θ from in a generative model, as shown in Figure 1.3, that vector, which represents a topic proportions vector in a topic model, can be drawn from a Dirichlet distribution G_d that was drawn from a DP. Such a model can accommodate an infinite number of topics. To share topics among different documents, the document-specific Dirichlet distributions G_d can be tied together by drawing their base measure G_0 from another DP. More formally: $G_0 \sim DP(H, \gamma)$. This final construction is known as the Hierarchical Dirichlet Process (HDP). By integrating G_d

out of this model, we get the Chinese restaurant franchise process (CRFP) [11];

$$\theta_{di} | \theta_{d,1:i-1}, \alpha, \boldsymbol{\psi} \sim \sum_{b=1}^{b=B_d} \frac{n_{db}}{i-1+\alpha} \delta_{\psi_{db}} + \frac{\alpha}{i-1+\alpha} \delta_{\psi_{db^{new}}} \quad (\text{B.2})$$

$$\psi_{db^{new}} | \boldsymbol{\psi}, \gamma \sim \sum_{k=1}^K \frac{m_k}{\sum_{l=1}^K m_l + \gamma} \delta_{\phi_k} + \frac{\gamma}{\sum_{l=1}^K m_l + \gamma} H \quad (\text{B.3})$$

where ψ_{db} is topic b for document d , n_{db} is the number of words sampled from it, $\psi_{db^{new}}$ is a new topic, B_d is the number of topics in document d , and m_k is the number of documents sharing topic ϕ_k .

The generative process of this model follows a Chinese restaurant model. In this metaphor, a restaurant represents a document, a customer represents a word, and a dish represents a topic. Customers sitting at the same table in a Chinese restaurant share the same dish served on that table. When customer i arrives at a restaurant, she is assigned a table to sit at. This table could be an empty table with probability $\frac{\alpha}{i-1+\alpha}$, or it could be a table occupied by n_{db} customers with probability $\frac{n_{db}}{i-1+\alpha}$. If an empty table is assigned, the dish served on this table could either be a new dish that has not been served in any restaurant before with probability $\frac{\gamma}{\sum_{l=1}^K m_l + \gamma}$ where m_l is the number of restaurants that have served dish l before, or the dish served on the table could be a dish that has been served on m_k tables across all restaurants with probability $\frac{m_k}{\sum_{l=1}^K m_l + \gamma}$. Table assignment and dish selection follows a richer scheme.

To add temporal dependence in our model, we can use the temporal Dirichlet process mixture model (TDPM) proposed in [1] which allows unlimited number of mixture compo-

nents. In this model, G evolves as follows [2]:

$$G_t | \phi_{1:k}, G_o, \alpha \sim DP(\zeta, D) \quad (\text{B.4})$$

$$\zeta = \alpha + \sum_k m'_{kt} \quad (\text{B.5})$$

$$D = \sum_k \frac{m'_{kt}}{\sum_l m'_{lt} + \alpha} \delta(\phi_k) + \frac{\alpha}{\sum_l m'_{lt} + \alpha} G_o \quad (\text{B.6})$$

$$m'_{kt} = \sum_{\delta=1}^{\Delta} \exp^{-\delta/\lambda} m_{k,t-\delta} \quad (\text{B.7})$$

where m'_{kt} is the prior weight of component k at time t , Δ , λ are the width and decay factor of the time decaying kernel. For $\Delta = 0$, this TDPM will represent a set of independent DPMs at each time step, and a global DPM when $\Delta = T$ and $\lambda = \infty$. As we did with the time-independent DPM, we can integrate out $G_{1:T}$ from our model to get a set of parameters $\theta_{1:t}$ that follows a Poly-urn distribution:

$$\theta_{ti} | \theta_{t-1:t-\Delta}, \theta_{t,1:i-1}, G_o, \alpha \propto \sum_k (m'_{kt} + m_{kt}) \delta(\phi_{kt}) + \alpha G_o \quad (\text{B.8})$$

We can make word distributions and topic trends evolve over time if we tie together all hyper-parameter base measures G_o through time. The model will now take the following form:

$$\theta_{tdi} | \theta_{td,1:i-1}, \alpha, \psi_{t-\Delta:t} \sim \sum_{b=1}^{b=B_d} \frac{n_{tdb}}{i-1+\alpha} \delta_{\psi_{tdb}} + \frac{\alpha}{i-1+\alpha} \delta_{\psi_{tdb}^{new}} \quad (\text{B.9})$$

$$\begin{aligned} \psi_{tdb}^{new} | \psi, \gamma \sim & \sum_{k:m_{kt}>0} \frac{m_{kt} + m'_{kt}}{\sum_{l=1}^{K_t} m_{lt} + m'_{lt} + \gamma} \delta_{\phi_{kt}} \\ & + \sum_{k:m_{kt}=0} \frac{m_{kt} + m'_{kt}}{\sum_{l=1}^{K_t} m_{lt} + m'_{lt} + \gamma} P(\cdot | \phi_{k,t-1}) \\ & + \frac{\gamma}{\sum_{l=1}^{K_t} m_{lt} + m'_{lt} + \gamma} H \end{aligned} \quad (\text{B.10})$$

where ϕ_{kt} evolves using a random walk kernel like in [11]:

$$H = N(0, \sigma I) \tag{B.11}$$

$$\phi_{k,t} | \phi_{k,t-1} \sim N(\phi_{k,t-1}, \rho I) \tag{B.12}$$

$$w_{tdi} | \phi_{kt} \sim \mathcal{M}(L(\phi_{kt})) \tag{B.13}$$

$$L(\phi_{kt}) = \frac{\exp(\phi_{kt})}{\sum_{w=1}^W \exp(\phi_{ktw})} \tag{B.14}$$

We can see from (3.9), (3.10), (3.11) the non-conjugacy between the base measure and the likelihood.

Appendix C

Graphical models

Probabilistic graphical models are tools that allow us to visualize the independence relationship between random variables in a model and use available graph manipulation algorithms to model complex systems, tune the model parameters, and infer the likelihood of events.

Probabilistic graphical models, or simply graphical models, can be categorized into *directed* graphical models (otherwise known as Bayesian networks), and *undirected* graphical models (called Markov networks, or Markov Random Fields [MRF] as well).

An example of a directed graphical model is given in Figure C.1. This graph models the independence/dependence relationship between seven random variables. The structure of the graph and the relationship between the variables were created using human judgment. Algorithms for learning the graph structure and parameters do exist [34] but were not used in this simple example. The nodes in the graph represent random variables and the directed edges show the flow of influence of the random variables. In this model, the priors *Genes*, *Training*, *Drugs*, define the probability that an athlete would have good genes, train well, and take performance-enhancing drugs, respectively. These three random variables affect the athlete's *Performance*, which along with the probability that the *Other athletes* would perform well or not affect the probability that our athlete would win a *Medal*. Drug doping can be tested separately by its effect on the *Drug test*.

An example of an undirected graphical model is given in Figure C.2 where the direct interaction between market stocks' values is represented by the graph edges and the nodes

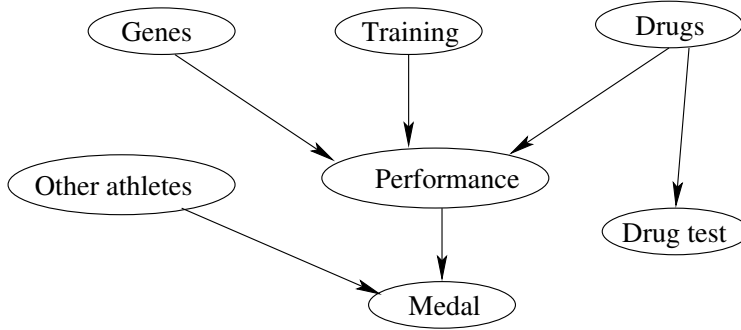


Figure C.1: *Directed graphical model.*

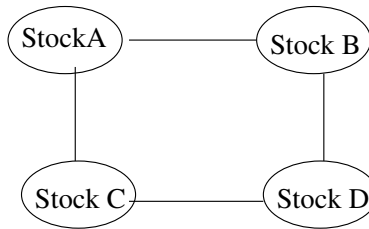


Figure C.2: *Undirected graphical model.*

themselves represent the value of the market stock. Since *Stock A* and *Stock D* are not neighbors in the graph they do not affect each other directly.

Simply speaking, a graphical model (directed or undirected) represents a joint distribution P over a set of random variables \mathcal{X} . For inference on a set of variables, we can exhaustively marginalize over the rest of the variables in the model. However, the inference complexity will increase exponentially with the number of variables and the problem would become intractable. This problem can be avoided in Bayesian networks (directed graphs) by making use of the independence relationships between the variables as follows:

Definition 1. Given a Bayesian network \mathcal{G} defined over a set of random variables $\mathcal{X} = \{X_1, \dots, X_n\}$ and a probability distribution $P_{\mathcal{X}}$ over the same set of variables. Then we say that $P_{\mathcal{X}}$ factorizes according to \mathcal{G} if

$$P_{\mathcal{X}} = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (\text{C.1})$$

where $Pa(X_i)$ is the set of parent nodes of X_i . This follows from the effect of the blanket variables that separate variables as follows:

Definition 2. Given a Bayesian network \mathcal{G} defined over a set of random variables $\mathcal{X} = \{X_1, \dots, X_n\}$. A variable X_i is independent of its non-descendants given its parents:

$$\forall X_i, (X_i \perp \text{nonDescendants} | \text{Pa}(X_i)) \quad (\text{C.2})$$

Analogously, for Markov networks:

Definition 3. Given a Markov network structure \mathcal{H} defined over the set of random variables $\mathcal{X} = \{X_1, \dots, X_n\}$ and a distribution $P_{\mathcal{H}}$ defined over the same space. We say that the distribution $P_{\mathcal{H}}$ factorizes over \mathcal{H} if there is a set of factors $\pi_1[D_1], \dots, \pi_m[D_m]$ where D_i is a clique in \mathcal{H} such that:

$$P_{\mathcal{H}} = \frac{1}{Z} P'_{\mathcal{H}}(X_1, \dots, X_n) \quad (\text{C.3})$$

$$P'_{\mathcal{H}} = \prod_{i=1}^m \pi_i[D_i] \quad (\text{C.4})$$

$$Z = \sum_{X_1, \dots, X_n} P'_{\mathcal{H}}(X_1, \dots, X_n) \quad (\text{C.5})$$

where $P'_{\mathcal{H}}$ is an unnormalized measure, and Z is a normalizing constant.

Independency in Markov networks can be identified using blankets of variables [42]:

Definition 4. Given a Markov network \mathcal{H} defined over a set of random variables $\mathcal{X} = \{X_1, \dots, X_n\}$. Let $\mathcal{N}_{\mathcal{H}}(X_i)$ be the set of X_i neighbors in the graph. Then, the *local Markov independencies* associated with \mathcal{H} can be defined as:

$$\mathcal{I}(\mathcal{H}) = \{(X \perp \mathcal{X} - \mathcal{N}_{\mathcal{H}}(X) - X | \mathcal{N}_{\mathcal{H}}(\mathcal{X})) : X \in \mathcal{X}\} \quad (\text{C.6})$$

C.1 Inference algorithms

Inference on graphical models could be a *conditional independence query*, or a *Maximum a Priori (MAP)* or its special (and important) case the *Most Probable Explanation (MPE)* among other types of queries.

A conditional independence query finds the probability of assignment of a random variable to a set of values given some evidence:

$$P(Y|E = e) = \frac{P(Y, e)}{P(e)} \quad (\text{C.7})$$

The MPE query finds the most likely assignment of values to all non-evidence variables. In other words, it finds the value of w that maximizes $P(w, e)$, where w is the values assigned to $W = \mathcal{X} - E$.

As it turns out, the problem of inference on graphical models is \mathcal{NP} -hard at best [26], if no assumptions are made about the structure of the underlying graphical model [20]. However, exact inference could be done efficiently on low treewidth graphical models. In many other important models, approximate inference could yield good results.

The problem of inference could be dealt with as an optimization problem, or sampling problem. The first alternative tries to find a distribution Q that is similar enough to the desired distribution P by searching a set of easy candidate distributions \mathbf{Q} trying to find the one that minimizes the distance measure between P and Q . Among the distance measures that could be used are Euclidean distance, which would require doing hard inference on P taking us back to the main problem, and the entropy based Kullback-Leibler (KL)-divergence measure [27].

C.1.1 Example

Conditional Random Field (CRF) [45] models have replaced hidden Markov models (HMMs) and maximum entropy Markov models (MEMM) in many NLP applications such as named-entity recognition [51] and part-of-speech tagging (POS) [65] due to their ability to relax strong independence assumptions made in these models.

Figure C.3 shows a graphical representation for a skip-chain CRF adapted from [52]. Only the x nodes are observed, and identical words are connected (with *skip edges*) because they are likely to have the same label. Strong evidence at one end can affect the label at the other endpoint of the edge. Long range dependencies in a model like this one would be

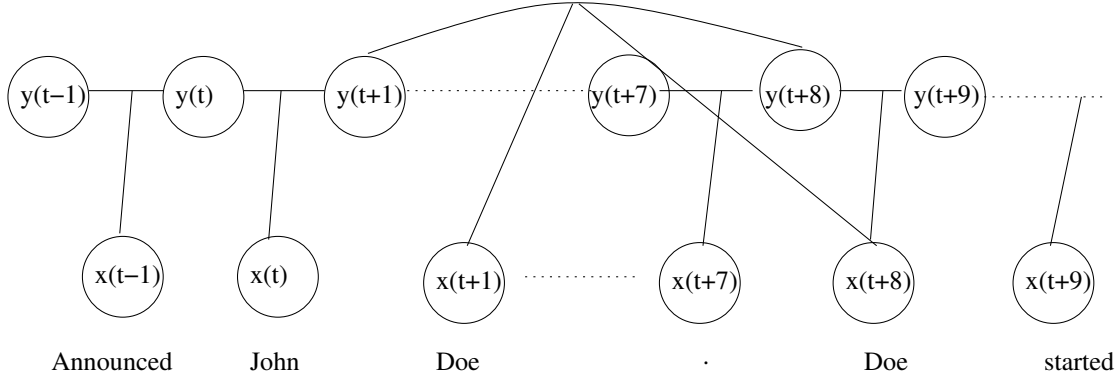


Figure C.3: A skip-chain CRF model for an NER task.

hard to represent using n -gram models because they tend to have too many parameters if n is large.

In a skip-chain model, skip-edge creation should depend on the application of the model. In a POS tagging task, this edge creation could be based on a similarity measure that uses stemming and word roots or edit distance. In the current example, we are connecting identical capitalized words for the NER task. Adding so many edges would complicate the model and make approximate inference harder. In our model, the probability of a label sequence y given an input x could be evaluated by:

$$p_{\theta}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \Psi_t(y_t, y_{t-1}, \mathbf{x}) \prod_{(u,v) \in \mathcal{I}} \Psi_{uv}(y_u, y_v, \mathbf{x}) \quad (\text{C.8})$$

where $\mathcal{I} = \{(u, v)\}$ is the set of all skip-edges sequence positions, Ψ_t are the factors for the linear-chain edges, and Ψ_{uv} are the factors over skip edges (like the one connecting y_{t+1} and y_{t+7} in Figure C.3). And we have:

$$\Psi_t(y_t, y_{t-1}, \mathbf{x}) = \exp \left\{ \sum_k \lambda_{1k} f_{1k}(y_t, y_{t-1}, \mathbf{x}, t) \right\} \quad (\text{C.9})$$

$$\Psi_{uv}(y_u, y_v, \mathbf{x}) = \exp \left\{ \sum_k \lambda_{2k} f_{2k}(y_u, y_v, \mathbf{x}, u, v) \right\}, \quad (\text{C.10})$$

where $\theta_1 = \{\lambda_{1k}\}_{k=1}^{K_1}$ are linear-chain template parameters, and $\theta_2 = \{\lambda_{2k}\}_{k=1}^{K_2}$ are the parameters of the skip-template. The observation functions $q_k(\mathbf{x}, t)$ can depend on arbitrary

positions of the input string. They factorize to:

$$f'_k(y_u, y_v, \mathbf{x}, u, v) = 1_{\{y_u=\tilde{y}_u\}}1_{\{y_v=\tilde{y}_v\}}q'_k(\mathbf{x}, u, v) \quad (\text{C.11})$$

Skip-chain CRFs have been introduced and tested by McCallum and Sutton on a problem of extracting speaker names from seminar announcements [52]. They reported an improved performance over linear-chain CRFs, which in turn outperforms HMM due to the relaxed independence assumptions between the states and the current observation. However, skip-chain CRFs fall into the trap of the implicit assumption of independence between current observations \mathbf{x} and the neighbors of the states on both ends of the skip-edge. For example, in one document, the word *China* representing the country would be linked with a skip-edge to the word *China* in *The China Daily*, and both could be labeled as COUNTRY, or ORGANIZATION in a NER task, even though they are different entities. A solution to this problem has been proposed by Finkel et al. [25]. They added richer long-distance factors to the original pairs of words factors. These factors are useful for adding exceptions to the rule that identical words have the same entity in text. However, due to the sparsity of the training data this model may fail.

Augmenting the features with neighbors of the states at the ends of the skip-edges may remedy this problem. This could take the form:

$$\Psi_{ef}(y_e, y_f, \mathbf{x}) = \exp \left\{ \sum_k \lambda_{3k} f_{3k}(y_e, y_f, \mathbf{x}, e, f) \right\} \quad (\text{C.12})$$

$$\Psi_{gh}(y_g, y_h, \mathbf{x}) = \exp \left\{ \sum_k \lambda_{4k} f_{4k}(y_g, y_h, \mathbf{x}, g, h) \right\}, \quad (\text{C.13})$$

Each factor of a skip-edge will be a function of four more states. This would require the use of different parameter estimation techniques on two stages [25]: First, we estimate the parameters of the linear-chain CRF, ignoring the skip-edges; then we heuristically select the parameters for the long-distance factors and neighboring states.

Another point worth considering is the precision of the probabilities of the forward-backward belief propagation algorithm. Their values become too small to be represented

within numeric precision. To solve this problem we could use the logarithmic domain for computation. Therefore,

$$\log(ab) = \oplus(a + b) \tag{C.14}$$

where

$$a \oplus b = \log(e^a + e^b), \tag{C.15}$$

or

$$a \oplus b = a + \log(1 + e^{b-a}) \tag{C.16}$$

Even though graphical models give us flexibility in modeling complex systems, these models could easily become intractable if the system input involved a set of rich nonindependent features. This is typical in many real world Natural Language Processing applications.

Since graphical models are usually used to represent a joint probability distribution $p(\mathbf{y}, \mathbf{x})$, where \mathbf{y} is the attributes of the entity we want to predict, and \mathbf{x} is our observation about this entity. Modeling the joint distribution for a system with rich setof features is complicated by the fact that it requires modeling the distribution $p(\mathbf{x})$ with its complex dependencies, causing our model to become intractable. This problem can be solved by modeling the conditional distribution $p(\mathbf{y}|\mathbf{x})$ instead. This approach in representation is known as Conditional Random Fields (CRFs) [45].

Appendix D

Information theory

Many definitions and equations used in this thesis can be better understood in the context of information theory. The average amount of information contained in a random variable and the relative entropy measures are useful in understanding how two probability distributions diverge from each other.

D.1 Entropy

The entropy $H(X)$ of a random variable X is defined by [78]:

$$H(X) = - \sum_x p(x) \log p(x) \tag{D.1}$$

The base for this logarithm can be chosen to be any number greater than 1. If the value chosen is 2, the unit for this entropy measure is the *bit*. If the base is Euler's number e , the unit of this entropy measure is *nat*, for natural logarithm. In this thesis, the base of the logarithm used is the size of the code alphabet, which is 2. It is to be noted that in computer science a bit is an entity that can take one of two values, 1 or 0. In information theory, a bit is a unit of measurement for the entropy of a random variable.

It is usually common in information theory to express entropy of a random variable in terms of its expected value. An alternative definition of the entropy could then given by

$$H(X) = -E \log p(X) \tag{D.2}$$

where

$$Eg(x) = \sum_x p(x)g(x) \tag{D.3}$$

The entropy $H(X)$ of a random variable X is a functional of the probability distribution $p(x)$. It measures the average amount of information contained in X , or the amount of *uncertainty* removed from X by revealing its outcome. The value of the entropy depends on $p(x)$ and not the values in its alphabet \mathcal{X} . In information theory, an alphabet \mathcal{X} of a random variable is the set of all values a random variable may take.

To maximize the entropy of random variable we want to maximize the uncertainty of the outcome of that variable. For a binary random variable X with $\mathcal{X} = \{0, 1\}$, this is achieved when both of its outcomes are equally likely to occur, i.e. when $p(0) = p(1) = 0.5$.

The joint entropy of two random variables X and Y is defined by

$$H(X, Y) = - \sum_{x,y} p(x, y) \log p(x, y) \tag{D.4}$$

$$= -E \log p(X, Y) \tag{D.5}$$

The joint entropy $H(X, Y)$ of two random variables X and Y is given by

$$H(X, Y) = - \sum_{x,y} p(x, y) \log p(x, y) \tag{D.6}$$

$$= -E \log p(X, Y) \tag{D.7}$$

In some cases, the outcome of one random variable might give a clue about the outcome of another variable. In that case, the definition of conditional entropy would be of high interest to us.

For two random variables X and Y the conditional entropy of Y given X is

$$H(Y|X) = - \sum_{x,y} p(x,y) \log p(y|x) \quad (\text{D.8})$$

$$= -E \log p(Y|X) \quad (\text{D.9})$$

D.2 Mutual information

For two random variables X and Y , the mutual information between X and Y is defined as

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (\text{D.10})$$

$$= E \log \frac{p(X,Y)}{p(X)p(Y)} \quad (\text{D.11})$$

From this definition, we can see that mutual information is symmetrical in X and Y , i.e. $I(X;Y) = I(Y;X)$.

D.3 Information divergence

In many communication applications we want to measure how a probability distribution p differs from another probability distribution q which is defined over the same alphabet. This measure is usually desired to be a non-negative one. The *information divergence* measure from p to q can be used in this case and it is defined as

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \quad (\text{D.12})$$

$$= E_p \log \frac{p(X)}{q(X)} \quad (\text{D.13})$$

where E_p is the expectation with respect to p .

In information theory, information divergence is also known as relative entropy. This makes sense as [D.12](#) could be interpreted as the entropy of p relative to the entropy of q as defined by the entropy measure in [D.1](#) and [D.2](#). Information divergence is also known as *Kullback-Leibler distance* as well. However, care must be taken when using this term as the information divergence measure is not symmetric in p and q , and does not satisfy the triangular inequality. i.e. $D(p||q) \neq D(q||p)$. Whereas the word distance in many situations imply symmetry.

Appendix E

Table of notations

Symbol	Definition
$DP(\cdot)$	Dirichlet process
G_0	base measure
α	concentration parameter
θ	Dirichlet distribution parameter space
$\phi_{1:k}$	Distinct values of θ
m_k	number of parameter θ with ϕ_k value
m'_{kt}	prior weight of component k at time t
Δ	width of time decaying kernel
λ	decay factor of a time decaying kernel
w	word in a document
H	base measure of the DP generating G_0
γ	concentration parameter for the DP generating G_0
ψ_{tdb}	topic b for document d at time t
n_{tdb}	number of words sampled from ψ_{tdb}
$\psi_{tdb^{new}}$	a new topic
B_d	number of topics in document d
$\mathcal{N}(m, v_0)$	a Gaussian distribution with mean m and variance v_0^2
$\beta_{i,k,w}$	distribution of words over topic k at time i for word w
s_i	time stamp for time index i
Δ_{s_j, s_i}	time duration between s_i and s_j
$Dir(\cdot)$	Dirichlet distribution
$z_{t,n}$	topic n sampled at time t
I	identity matrix
$Mult(\cdot)$	multinomial distribution
$\pi(\cdot)$	mapping function

Table E.1: *Table of notations*