Continuous-time Infinite Dynamic Topic Models

The Dim Sum Process for Simultaneous Topic Enumeration and Formation

Wesam Elshamy and William H. Hsu

Kansas State University

1. Dynamic Topic Modeling and Event Stream Mining

In this chapter, we discuss the problem of simultaneous topic enumeration and tracking (STEF): that of maintaining both the number of topics and a parametric representation of their changing semantics as estimated from a dynamically changing document collection, or *corpus*. We develop a continuous-time dynamic topic model for addressing the STEF problem. We extend the existing Bayesian representation of evolving topic models with two temporal components: the word distribution per topic, and the number of topics. Both evolve in continuous time.

1.1 Problem Overview: Goals of Static and Dynamic Topic Modeling

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

The predominant topic models in current use are extensions of probabilistic topic models such as latent Dirichlet allocation (LDA). Key technical strengths of these models include: their scalability to large *corpora*, or document collections; their speed of inference for new documents; their ability to estimate a topic membership distribution or other finite mixture for a new document, given previously observed training corpora, which may be continuously evolving for online applications of incremental topic modeling. Models such as LDA are traditionally but not intrinsically atemporal: there exist similar temporal dynamic topic models. Ahmed and Xing 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. 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.* presented a model where the word distribution per topic evolves in continuous time. They used variational Bayes methods for fast, scalable inference. A major limitation of their model is that it uses a predefined and fixed number of topics that does not evolve over time.

In this chapter, we first survey the mathematical foundations and belief updating (inference) tools for such models. Next we look at previous advances that incorporate time, but only allow STEF for discrete time: that is, they only allow the cardinality of topics to change when the time quantum is fixed. This presents a key technical challenge for dynamic topic modeling (DTM) as applied to monitoring of current events, where topics may emerge asynchronously and in bursty or non-bursty modes, depending on their scope and the time scale of the events. We then develop and use a topic model that is a mixture of these two models where the word distribution per topic and the number of topics evolve in continuous time. Finally, we report experiments on topic detection and tracking (TDT) using documents collected from single news media sources, in two cases using simple web crawlers and scrapers with basic content extraction. The need for the dynamic continuous-time topic model such as the type we are developing is evident in the mass media business, where news stories are published around the clock. We show that STEF is a basic task requirement of even some single-source news monitoring tasks. We hypothesize that this problem is worse for heterogeneous document streams, such as print media (magazines, newspapers), broadcast media, the web, and social media, which admit more highly variable rates of publication and commentary, and thus more variance in overall document stream rates.

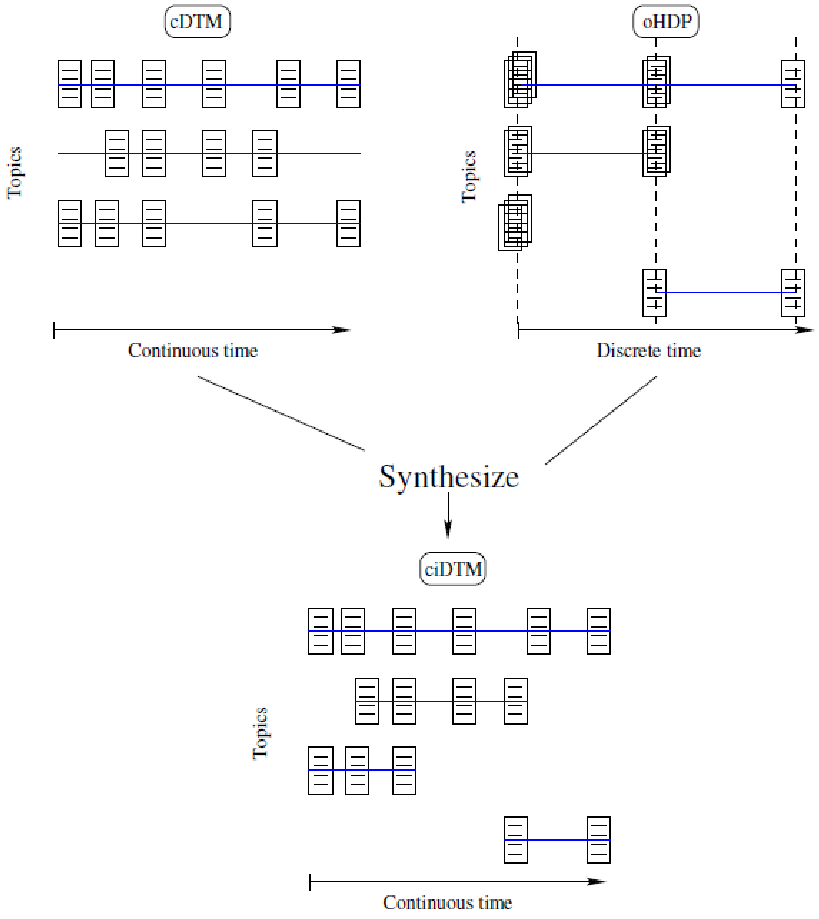

Figure 1. Top left: the continuous-time dynamic topic model (cDTM) has a continuous-time domain. Word and topic distributions evolve in continuous time, but the number of topics in this model is fixed. This may lead to having two separate topics being merged into one topic which was the case in the first 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. The continuous-time infinite dynamic topic model (ciDTM) is a mixture of oHDP and cDTM. It has a continuous-time domain like cDTM, and its number of topics evolves over time as in oHDP. It overcomes limitations of both models regarding evolution of the number of topics and time-step granularity.

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.

1.1.1 Central Hypothesis

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. 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 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 with a very small time unit (epoch), and more accurate than using a continuous-time bounded-topic model. This is illustrated in Figure 1.

1.1.2 Technical Objectives

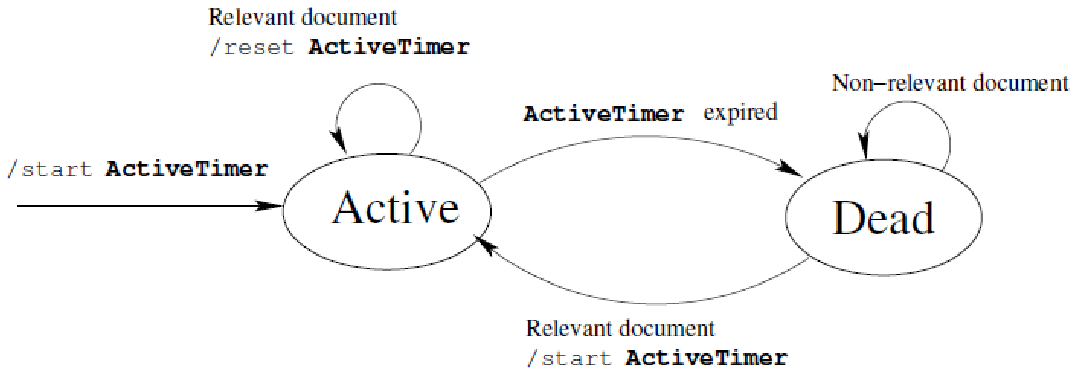

Figure 2. Topic state-transition diagram: Oval shaped nodes represent states with their labels written on the inside. Arrows represent transitions from one state to another, or to itself. Each arrow has an Event / Action describing the event that triggered the transition, and the action taken by the system in response to that transition. The start state is represented by an arrow with no origin pointing to the state.

A news timeline is characterized by a mixture of topics. When a news story arrives in 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 from 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 modeling system that receives text streams, a topic has two states that represent its level of dormancy in some monitoring context: **`Active`** and **`Dead`**. Transition between these two states is illustrated in a state-transition diagram in Figure 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 transition to the **`Dead`** state is taken, designating the current topic as inactive. 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 back to the **`Active`** state, and the timer (**`ActiveTimer`**) is started.

1.2 Static Task: Atemporal Topic Formation

Since the emergence of the World Wide Web in the early 1990s, the volume of digitized knowledge has been increasing at an unprecedented exponential rate. The past decade has witnessed the rise of projects such as JSTOR, Google Books, the Internet Archive and Europeana 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.

Novel and innovative techniques have been developed for categorizing, searching and presenting the digital material. The usefulness of these services to the end user, which are free in most cases, is evident by their ever increasing application in the fields of computer-assisted instruction, entertainment, and decision support.

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 generally 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. Most of these existing models are atemporal and perform badly in finding old relevant stories because the word collections that identify 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), probabilistic LSA (pLSA), and latent Dirichlet allocation (LDA).

1.3 Significance of Dynamic Tasks: Topic Detection and Tracking

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. 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.

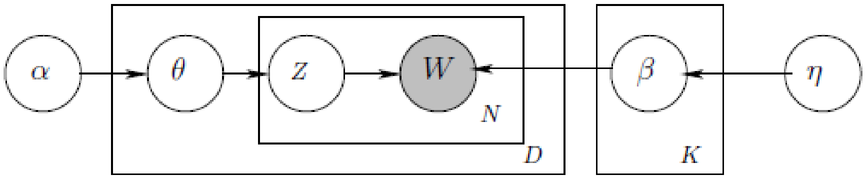

Figure 3. Latent Dirichlet allocation (LDA) graphical model. The rectangles, known as plates in this notation, represent replication. For nested plates, an inner plate has multiplicity equal to the replication value shown on its lower-right corner times the multiplicity of its parent, or just the corner value if it is at the top level. Thus, the 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*. The plate with multiplicity *K* represents a general catalog, or domain-specific term lexicon, consisting of *K* topic labels. Nodes are labeled using the variable they represent. 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.

A graphical model for latent Dirichlet allocation is shown in Figure 3. 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 is because the Dirichlet distribution is convenient distribution on the simplex, it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution. These properties are useful in developing inference procedures and parameter estimation methods, as pointed out by Blei *et al*.

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. This is done to limit the number of relevant stories, but meanwhile casts doubt over the precision, recall, and relevance 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.

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, and how different topics are correlated with each other, 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 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". In the field of signal processing, topic models were used in audio scene understanding, where audio signals were assumed to contain latent topics that generate acoustic words describing an audio scene. Topic modes have also been used extensively in text summarization, building semantic question answering systems, stock market modeling, and music analysis.

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. This may also be useful for adaptive multi-time scale event tracking and inference tasks.

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. For a detailed example, we refer the interested reader to (Hsu, Abduljabbar, Osuga, et al., 2012).

2. Background and Related Work

In this section, we survey two relevant aspects of previous work: the common probabilistic foundations of the prevailing generative, variational, and Markov chain Monte Carlo methods for topic modeling; and the specific variational clustering algorithms upon which the work reported in this chapter, and its building blocks, are built. These specifically include variational inference in *online hierarchical Dirichlet process* (*oHDP*) models, which we cover in Section 2.2.2, and a different variational method, variational Kalman filtering, for *continuous-time dynamic topic models* (*CDTM*), which we cover in Section 2.2.3.

2.1 Bayesian Models and Inference Algorithms

Evaluation of the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{X})$ of the set of latent variables $\boldsymbol{Z}$ given the observed data variable set $\boldsymbol{X}$ is essential in topic modeling applications. This evaluation is infeasible in real-life applications of practical interest due to the high number of latent variables we need to deal with. 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 is analytically intractable.

As in the case of many engineering problems, when finding an exact solution is computationally intractable or too expensive in practice, 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. Approximate inference methods fall broadly into two categories: stochastic and deterministic.

Stochastic methods, such as Markov Chain Monte Carlo (MCMC) methods, can theoretically reach exact solutions in limited time, given arbitrarily high processing power for parallel random sampling. 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 regarding the form of the posterior distribution or the way the nodes of the graphical model can be factorized. These methods therefore cannot reach an exact solution, even with unlimited computational resources.

2.1.1 Probabilistic Representation

The main problem in graphical model applications is finding an approximation for the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{X})$ and the model evidence $p(\boldsymbol{X})$, where $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_N}\}$ is the set of all observed variables and $\boldsymbol{Z} = \{\boldsymbol{z_1}, \boldsymbol{z_2}, \cdots, \boldsymbol{z_N}\}$ is the set of all latent variables and model parameters. $p(\boldsymbol{X})$ can be decomposed using:

$$\ln p(\boldsymbol{X}) = \mathcal{L}(q) + D(q \parallel p)$$

where:

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

$$D(q \parallel p) = -\int q(\boldsymbol{Z}) \ln\left\{\frac{p(\boldsymbol{Z} \mid \boldsymbol{X})}{q(\boldsymbol{Z})}\right\} d\boldsymbol{Z}$$

Our goal is to find a distribution $q(\boldsymbol{Z})$ that is as close as possible to $p(\boldsymbol{Z}|\boldsymbol{X})$. To do so, we need to minimize the Kullback-Leibler (KL) distance $D(q \parallel p)$ between them. 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.

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

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

$$q(\boldsymbol{Z}) = \prod_{i=1}^{M} q_i(\boldsymbol{Z_i})$$

This approximation method is known in the physics domain as an application of *mean field theory*.

To maximize the lower bound $\mathcal{L}(q)$, we need to minimize each of the factors $q_i\boldsymbol{Z_i}$. We can do so by substituting the equation for $q(\boldsymbol{Z})$ into that for $\mathcal{L}(q)$ to get the following:

$$\begin{aligned}\mathcal{L}(q) &= \int \prod_{i=1}^{M} q_i \left\{ \ln p(\boldsymbol{X}, \boldsymbol{Z}) - \sum_i \ln q_i \right\} d\boldsymbol{Z} \\ &= \int q_j \left\{ \int \ln p(\boldsymbol{X}, \boldsymbol{Q}) \prod_{i \neq j} q_i d\boldsymbol{Z_i} \right\} - \int q_j \ln q_i \, d\boldsymbol{Z_i} + c \\ &= \int q_j \ln \bar{p}(\boldsymbol{X}, \boldsymbol{Z_j}) d\boldsymbol{Z_j} - \int q_j \ln q_i \, d\boldsymbol{Z_i} + c\end{aligned}$$

where

$$\ln \bar{p}(\boldsymbol{X}, \boldsymbol{Z_j}) = \mathbb{E}_{i \neq j}[\ln p(\boldsymbol{X}, \boldsymbol{Q})] + c$$

and

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

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

Since the final expanded form of $\mathcal{L}(q)$ is a negative KL divergence between $q_j(\boldsymbol{Z_j})$ and $\bar{p}(\boldsymbol{X}, \boldsymbol{Z_j})$, we can maximize $\mathcal{L}(q)$ with respect to $q_j(\boldsymbol{Z_j})$ and obtain:

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

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

$$q_j^*(\boldsymbol{Z_j}) = \frac{\exp\left(\mathbb{E}_{i \neq j}[\ln p(\boldsymbol{X}, \boldsymbol{Z})]\right)}{\int \exp\left(\mathbb{E}_{i \neq j}[\ln p(\boldsymbol{X}, \boldsymbol{Q})]\right) d\boldsymbol{Z_j}}$$

For tutorial examples of how this factorized distribution can be used for variational approximation (*e.g.*, of univariate Gaussians) and for more concrete details of its adaptation to variational inference in hierarchical Dirichlet processes, we refer the interested reader to (Elshamy, 2012).

2.1.2 Brief Survey of Variational Inference

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:

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

Here we have introduced a variational parameter λ, for which we seek the value that minimizes the overall function $\log(x)$.

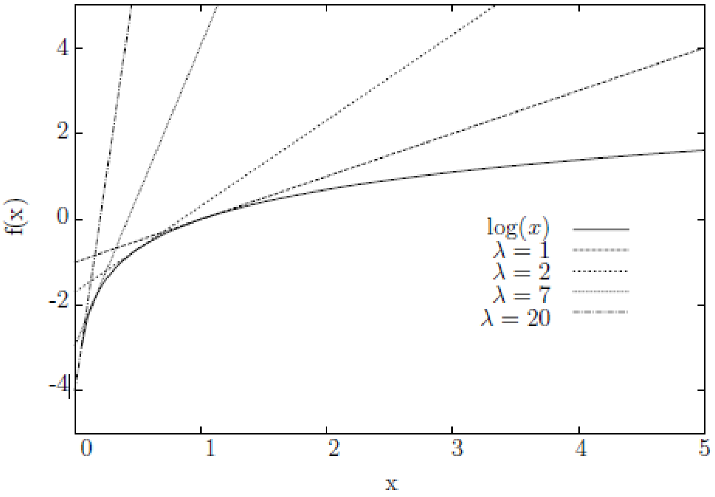

Figure 4. Variational transformation of the logarithmic function. The transformed function ($\lambda x - \log\lambda - 1$) forms an upper bound for the original function.

As we can see in Figure 4, 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:

$$\forall\lambda, \log(x) \leq \lambda x - \log\lambda - 1$$

We use variational inference to find an approximation to the true posterior of the latent topic structure, consisting of: the topic distribution per word, the topic distribution per document, and the word distribution over topics. Specifically, we use variational Kalman filtering in continuous time for this problem. The variational distribution over the latent variables can be factorized as follows:

$$q\left(\beta_{\mathbf{1}:\boldsymbol{T};\boldsymbol{z}_{\mathbf{1}:\boldsymbol{T}};\mathbf{1}:\boldsymbol{N}}, \boldsymbol{\theta}_{\mathbf{1}:\boldsymbol{T}} | \widehat{\beta}, \phi, \gamma\right) =$$
$$\prod_{k=1}^{K} q\left(\beta_{(1,k)}, \beta_{(2,k)}, \cdots, \beta_{(T,k)} | \hat{\beta}_{(1,k)}, \hat{\beta}_{(2,k)}, \cdots, \hat{\beta}_{(T,k)}\right) \times$$
$$\prod_{t=1}^{T} q(\theta_t|\gamma_t) \prod_{n=1}^{N_t} q\left(z_{(t,n)}|\phi_{(t,n)}\right)$$

where β is the word distribution over topics and $\boldsymbol{\beta_{1:T;z_{1:T};1:N}}$ is the word distribution over topics for time $1 \cdots T$, topic $z_1 \cdots z_T$, and word index $1 \cdots N$, where N is the size of the dictionary.

In the equation above, 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 $z_{(t,n)}$. $\left\{\hat{\beta}_{(1,k)}, \hat{\beta}_{(2,k)}, \cdots, \hat{\beta}_{(T,k)}\right\}$ are Gaussian variational observations for the Kalman filter.

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 especially when the time granularity gets finer. In our model, we use sparse variational inference 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:

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

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

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

where

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

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

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

where

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

$$= V_{(t-1,w)} + \left(\frac{V_{t-1}}{V_{t-1} + v\Delta_{s_t}}\right)^2 \left(\tilde{V}_t + V_{t-1} + v\Delta_{s_t}\right)$$

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

$$\mathcal{L}(\hat{\beta}) \geq \sum_{t=1}^{T} \mathbb{E}_q\left[\log\ p(w_t|\beta_t) - \ \log\ p(\hat{\beta}_t|\beta_t)\ \right]$$

$$+ \sum_{t=1}^{T} \log\ q(\hat{\beta}_t|\hat{\beta}_{1:t-1})$$

where

$$\begin{aligned}
\mathbb{E}_q[\log\ p(w_t|\beta_t)] &= \sum_w n_{(t,w)} \mathbb{E}_q\left(\beta_{(t,w)} - \log \sum_w \exp\left(\beta_{(t,w)}\right)\right) \\
&\geq \sum_w n_{(t,w)}\, \tilde{m}_{t,w} \\
&\quad - n_t \log \sum_w \exp\left(\tilde{m}_{t,w} + \frac{\tilde{V}_{(t,w)}}{2}\right) \\
\mathbb{E}_q\left[\log\ p\left(\hat{\beta}_t|\beta_t\right)\right] &= \sum_w \delta_{(t,w)} \mathbb{E}_q \log\ q\left(\hat{\beta}_{(t,w)}|\beta_{(t,w)}\right) \\
\log\ q\left(\hat{\beta}_t|\beta_{1:t-1}\right) &= \sum_w \delta_{(t,w)} \log\ q\left(\hat{\beta}_{(t,w)}|\beta_{(1:t-1,w)}\right)
\end{aligned}$$

where $\delta_{(t,w)}$ is the Dirac delta function 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 $n_t\ = \sum_w n_{(t,w)}$.

2.2 Existing Dynamic Topic Models and Their Limitations

Several studies have been done to account for the changing latent variables of the topic model. Xing presented a dynamic logistic-normal-multinomial and logistic-normal-Poisson models that he used later as building blocks for his models. Wang and McCallum presented a non-Markovian continuous-time topic model in which each topic is associated with a continuous distribution over timestamps. Blei and Lafferty proposed a dynamic topic model in which the topic's word distribution and popularity are linked over time, though the number of topics is fixed. This work was continued by other researchers who extended this model. In the following we describe some of these extended 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 continuously. Using a fixed word distribution for such topics and for many other topics that typically evolve in time may result in errors in both document classification and topic inference. The change in meaning of the topic over time, or topic drift, can be measured by taking 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.

As the topic continues to evolve and the distance between the original and new topic becomes greater, document classification and topic inference errors can grow. Therefore, there is a strong need to extend such topic models with time and time-dependent parameters to reflect the changing word distribution for topics in a collection of documents. We hypothesize that this will improve topic inference in a dynamic collection of documents.

2.2.1 Non-Markov Process Models

Several topic models were suggested that add a temporal component to the model. We will refer to them in this chapter as temporal topic models. These temporal topic models include Topics over Time (TOT), a generative model developed by Wang and McCallum. 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 is based upon a generative process in which for each document a multinomial distribution θ is sampled from a Dirichlet prior α.

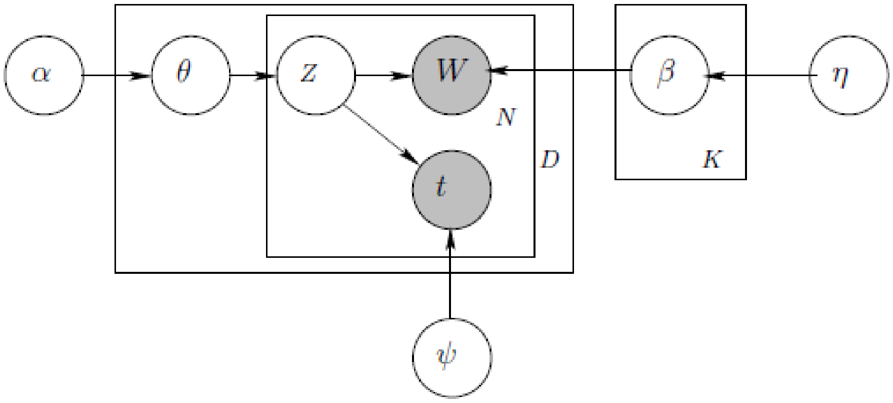

Figure 5. Topics Over Time (TOT) graphical model. In this view, one timestamp *t* is sampled from a Beta distribution per word in a document. All words in a single document have the same timestamp. All the symbols in this figure follow the notation given in Figure 3 and is the Beta distribution prior.

Conditioned on the value of this multinomial parameter 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.

Meanwhile, 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 seems natural to document collections where each document has a single timestamp, contrasts with the first view that the authors presented. It is depicted in Figure 5. In this version, which the authors ultimately adopted and implemented 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.

We note, however, that the word distribution per topic in this TOT model is fixed over time. "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." 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 time step. Picking a large time step leads to the problem of using documents that cover a large time period for learning these distributions or inferencing only one fixed distribution for them for that entire period of time. During this period, the word distributions for the topics covered in these documents evolved significantly. Picking a small time step complicates inference and learning, as the number of parameters explodes as the time step 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 time step 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 time step at that point will make inference and learning slower as it increases the number of model parameters. On the other hand, having a coarse time step at one point in time when a topic does not evolve quickly may be suitable for that time; however, the time step would then be too big if and when the topic starts evolving faster in time. In that case, documents that fall within one time step are inferenced using the fixed word distributions for the topics, not reflecting the change that happened to these topics. To take this case to the extreme, consider a very large time step covering then entire time period over which the document collection exists; this would be equivalent to using a classical latent Dirichlet allocation (LDA) model that has no notion of time at all.

We note further 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 are born, die, and are 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 greater than the actual number extant in a 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 article 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 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. This is what we would naturally expect. However, in the generative graphical model presented in Figure 5 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 timestamps. 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 5 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 mode 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 used an 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. 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. 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 numbers 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 fewer 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, benefitted 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.

In their *TimeMines* system, Swan and Jensen 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.

2.2.2 Discrete-Time Markov Process Models with Fixed or Variable Numbers of Topics

The topics over time (TOT) topic model surveyed in Section 2.2.1 is a temporal model but not a Markov process model. 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 analyze the dynamic social network of friends as it evolves over time using a Markovity assumption. Kleinberg 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.

Allowing a variable number of topics. Ahmed and Xing proposed a solution that overcomes the problem of having a fixed number of topics. This design, called an infinite Dynamic Topic Model (iDTM), allows for an unbounded number of topics and an evolving representation of topics according to Markov dynamics. Ahmed and Xing 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 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. For the mathematical model, an expanded discussion of the statistical parameters and updating algorithm, we refer the reader to (Ahmed & Xing, 2010) and (Elshamy, 2012). These references also explain the underlying mathematical model for the arrival process and selection process for iDTM: the recurrent Chinese restaurant franchise (RCRF) or Polya Urn model – and discuss the need to use hierarchical Dirichlet process models (HDP) in implementing online learning.

Online updating for incremental handling of large corpora and document streams. 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 especially 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 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.

Incrementality, the property of adaptive or learning systems that allows new data to be incorporated without reversing or resetting the results of previous adaptation, is a general open problem area in computer science. In machine learning, it is closely related to, and considered synonymous with, *online learning*, which allows handling of new input in a running system – usually taking a single instance or set of instances at a time, and often with real-time (*i.e.*, time-bounded) constraints. Wang et al. (2011) adapted the HDP to allow online learning, using a variant called an online Hierarchical Dirichlet Process (oHDP) model. Elshamy (2012) discusses this work in the context of document streams and gives an elaboration of the stick-breaking model for oHDP.

In summary, in streaming text topic modeling applications, the discrete-time models given above are brittle. In particular, they are susceptible to changes in temporal granularity. Extending such models to handle continuous time, when possible, gives them the necessary flexibility to account for such changes.

2.2.3 Continuous-Time Markov Process Models with a Fixed Number of Topics

Many Markov process models have been developed for use outside the area of topic modeling and the specific application of probabilistic modeling of topics from corpora. However, some of these models are general enough to admit use in topic modeling. For example, Nodelman *et al.* 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 Markovity assumption.

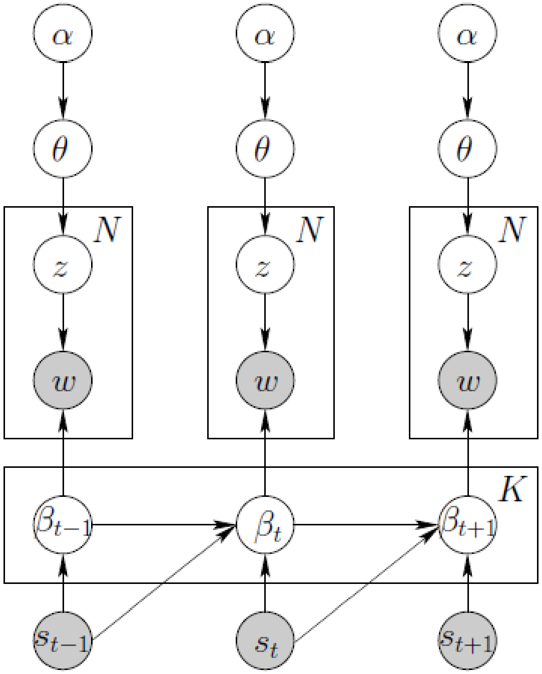

Figure 6. Graphical model representation of the continuous-time dynamic topic (CDTM) model.

Wang et al. (2008) proposed a continuous-time dynamic topic model (CDTM) that uses Brownian motion to model the evolution of topics over time. As shown in Figure 6, it adds timestamps s_i. To make the topics evolve over time, we define a Wiener process and sample β_t from it. The posterior $p(\beta_{(t,k)} | \beta_{(t-1,k)}, s)$, which is the distribution of the latent topic structure given the observed documents and their waiting times (i.e., latency, time delay, or spacing), is intractable. We resort to approximate inference. For this model, sparse variational inference can be used.

Even though CDTM 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 fewer 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-tuned number of topics, topics and storylines will get conflated.

This is related to the very similar problems of cluster cohesion, centrality, and separation being dependent on the number of centroids or medoids in partitioning-based clustering algorithms such as k-means and k-medoids. As previously discussed, it is further exacerbated when topics can actually emerge or become obsolete during the operating lifetime of the topic modeling system. A second problem discussed in Section 2.2 above is topic drift, related to the general problem of concept drift in machine learning, where a topic remains active but shifts gradually in meaning. If drift occurs because the position within a statistically-based projection is fluid, there may be some driving trend related to semantics of the topic relative to an unchanging (or negligibly changing) domain lexicon. A third final challenge is that topics can split and merge to absorb, recruit, or eject documents based on current relevance. This fungibility of topics is seen as grouping of documents “under” a topic heading – i.e., which new articles belong to which topic – but neither all topic models (even dynamic ones) necessarily account for this type of change.

2.2.4 The Challenge: Continuous Time with a Variable Number of Topics

In the first section, we defined the problem of simultaneous topic enumeration (counting the number of extant topics according to some activity criterion) and formation (defining the boundary and a statistical mapping from words to topics). Just as with other generative models, and with the general challenge in clustering of finding how many cluster centers there are and where each one lies, this task includes an inherent chicken-egg problem: how can we determine how many topics there are if they are simultaneously shifting, gaining and losing members, and moving in and out of existence?

Earlier in this chapter, we reviewed the state of the field: dynamic topic models that allow either the number of topics to change over time based on newly received documents, but operate at a fixed time granularity, or those that allow for continuous time by letting timestamp resolution be arbitrarily fine but not requiring that the arrival data be represented using a fixed minimum time quantum, or minimum grain size. We saw why this tradeoff is not satisfactory for practical reasons: inability to handle variable rate or mixed-rate document streams on the discrete-time side vs. being locked into a fixed *number* of topics regardless of timeliness or drift. The simple *ad hoc* fix of oversampling or upsampling discrete-time data is computationally intractable and infeasibly wasteful.

The remaining option is to find a way to hybridize the continuous-time model with a partial model that can update the number of topics with the same corpus, in discrete time.

3. Hybrid Solution: The Dim Sum Process for Dynamic Topic Modeling

In this section we describe a new contribution to predictive analytics: a probabilistic transfer learning ensemble based upon hyperparameter sampling that hybridizes adaptive parameter estimation tasks with shared attributes for a nonstationary process. We call this ensemble the *dim sum process*. In general, dim sum processes are a hybrid solution approach designed to track changes in parameter values for dynamical systems where both the hyperparameters and their dependent parameters are nonstationary. Traditionally, this issue has been approached using generative models and Markov chain Monte Carlo inference, but where this is computationally infeasible or the available data are insufficient, another approach is needed. Our hypothesis was that a multi-phase variational inference method might break the barrier.

The general idea applies to dynamic clustering problems where the number and cluster membership of observed objects can both change. Specifically, the number of clusters and the location of clusters can vary simultaneously based on local cohesion of arriving objects. The number is one of several hyperparameters that may indirectly govern cluster location, which is directly influenced by the parameters that relate location to observable features.

By specializing the clusters in the above abstract problem framework to topics, the objects to documents, the generative processes to document streams, the hyperparameter ensembles to the plate models used in natural language processing (NLP) and information retrieval (IR), especially topic modeling, we can see that simultaneous topic enumeration and formation (STEF) tasks are a special case of the above dynamic clustering task. We refer to the general problem as *dynamic enumeration of cluster arity and formation* (*DECAF*).

This section formulates two processes that share the same observations (documents) and operate on some shared hyperparameters. On the discrete-time side, both topic formation and enumeration can be performed; however, the requirement of this component is that it be a) **able to update** the number of topics (*enumeration*) and b) **sensitive to** the parameters that govern cluster localization (*formation*). In other words, the discrete-time side can import values for (b) or update them internally, but it must be able to export values for (a). An internal update capability will not suffice for the continuous-time task, as discussed in Chapter 2. On the continuous-time side, only topic formation is permitted; this component must a) **respect** all external constraints on the number of topics (enumeration) and b) be **able to update** the parameters that govern cluster localization (formation). In other words, the continuous-time side can import the current number of topics, but must be able to update and export parameters that directly determine the cluster shape and location.

Our representation of the STEF task using a dim sum process, including our choice of inference algorithms, is driven by two specific instantiations. The discrete-time model that exports the number of topics is instantiated as an online hierarchical Dirichlet process (oHDP) updated using variational inference. That of the continuous-time model that exports the "fine-grained updates" of the topic model parameters as a variational Kalman filter. Each of these comes with its own preferred and applicable inference algorithms for implementation. The ensemble thus gains the best of both worlds: the ability to spawn and retire clusters, combined with the ability to track the enumerated clusters in asynchronous continuous time (*i.e.*, without a minimum time slice).

3.1 Hyperparameter Sampling

3.2 Dim Sum Process for Continuous-Time Topic Modeling

We present a hybrid topic model called the continuous-time infinite dynamic topic model (ciDTM) that combines the properties of the continuous-time Dynamic Topic Model (cDTM) and the online Hierarchical Dirichlet Process model (oHDP). We will refer to the stochastic process we developed to combine the properties of these two systems as a *dim sum process*.

Dim sum (点心) is a style of Cantonese 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 for the different dishes the restaurant serves, to better satisfy the customers' tastes and suit seasonal changes and availability of these ingredients.

Items are ordered *a la carte*, but the contents of serving carts are initialized by a process of selection: this is usually the chef's decision, but also includes customers' new orders. We do not differentiate among parties to this decision process, but treat them as a unified assignment process that results in a filled tray of multiple dishes. Serving sizes are small for each dish (three to four pieces), and customers can order family-style, sharing dishes among members of a dining party, each one trying a wide variety of food. Once a customer has chosen a tray (a topic and its corresponding word mixture at the current time), we do not restrict their selection of dishes or units of food from that tray. We note that it is the choice of **tray** rather than the choice of **table** that the customer orders from – therefore, the dim sum process is agnostic as to seating arrangement, unlike the Chinese restaurant franchise process. This is a cosmetic distinction, whereas other differences we note below are mathematically significant.

The generative model of a dim sum process is analogous to that of a Chinese restaurant franchise process (CFRP) in the following ways:

1. *Groups* are the generalization of the concept of a **restaurant** in the original formulation of the Chinese Restaurant Process by Aldous (1985) and in the definition of hierarchical Dirichlet processes by Teh *et al.* (2005); however, they correspond to a single <u>restaurant service</u> for the dim sum process metaphor.
2. In the CFRP formulation, there is a notion of a *collection of groups*, not named in Teh *et al.* (2005) but referred to as corresponding to a **franchise**. This corresponds to a <u>set or series of services for a restaurant</u> (which is observed once for the single restaurant) in the dim sum process. In topic modeling, it corresponds to a corpus. This allows us to represent the explicit time-dependency, and the asynchronous nature, of the observed groups, as appropriate. In other words, we can timestamp the documents of the corpus.
3. Customers are seated in the restaurant according to a sequential arrival process for groups – **discrete** for iDTM and oHDP, <u>continuous</u> for dim sum process models such as ciDTM.
4. Customers, in one-to-one correspondence with parametric *factors* θ_i, select food from the local menu and are thereby mapped to **tables** ψ_k. In clustering, this corresponds to the notion of *objects* and *clusters*; in topic modeling, these are words and topics. By contrast, in dim sum processes, customers order <u>trays</u> of dishes whose ingredients (mixture) evolve in continuous time. The dishes correspond to factors θ_i but are thus in many-to-one correspondence with customers (objects/words), allowing us to capture additional analyzed features such as sense annotation, contextual cues and constraints, or co-occurrence or as a side benefit. The CRFP metaphor supposes that a family-style shared dish is the main course at one table (ergo, dishes and tables are in one-to-one correspondence) and that cluster drift corresponds to differences in the **flavor** of the dish at that table on that day. The dim sum metaphor supposes that a tray will evolve by gaining and losing dishes (undergoing factor updating) at the behest of the stream of customers (word sequence) and that cluster drift is accounted for by the <u>current collection of dishes</u> on the tray, like a cart shelf in a real dim sum restaurant.

When the dim sum process is used in a topic modeling application, each customer corresponds to a word and a single restaurant service represents a document – a finite word sequence, also represented as a bag of words. A topic corresponds to one tray, located on an arbitrary serving cart that is accessible to all customers. It is used to group a set of customers (words) to a tray (its word distribution).

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

1. At the start of each service, the dim sum restaurant (document) is empty.
2. The local menu ("today's choices") is a service-specific subset of the global menu, which persists across all services. In real dim sum restaurants, many copies of a global menu are printed and the unavailable items, which do **not** appear on the local menu, are struck out when the menu is offered to a customer.
3. The first customer (word) to arrive sits down at a table and orders a pre-selected tray (topic) of dishes from the local menu. The dishes on this tray are initially prepared from scratch, but other customers may now see and order from (be assigned to) this tray.
4. The second customer to arrive, at any table, has two options.
    a. She can with probability $\alpha/(1+\alpha)$ request the local menu and order a tray for her party or table, which will now be seen on the cart by every subsequent customer coming in during that service.
    b. She can indicate an existing tray with probability $1/(1+\alpha)$ and be served dishes that have been already ordered for that tray. These dishes are replenished for new customers who choose that tray and the ingredient supply is again sufficiently high to last the entire restaurant service, no matter how many customers arrive.
5. When the $(n+1)^{\text{st}}$ customer enters the restaurant, he can start a new tray on a cart with probability $\alpha/(n+\alpha)$, or he can point out tray k with probability $n_k/(1+\alpha)$ where n_k is the number of customers who have currently ordered the mixture (topic) corresponding to tray k.

Higher values of α lead to higher number of trays (topics) in use and corresponding ensembles of dishes (topic parameters) sampled in one restaurant service (document). We note that a document thus consists of a mixture of topics. 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).

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.

3.3 Mathematical Model and Use Case (Topic Enumeration and Formation)

The *continuous-time infinite dynamic topic model (ciDTM)* is a mixture of the online hierarchical Dirichlet process (oHDP) model presented earlier in Section 2.2.2, and the continuous-time dynamic topic model (cDTM) presented in Section 2.2.3. Figure 1 shows the conceptual behavior of this hybrid model.

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

1. We build a two-level hierarchical Dirichlet process (HDP) such as the one presented in Section 2.2.3.
2. 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:

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

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

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.

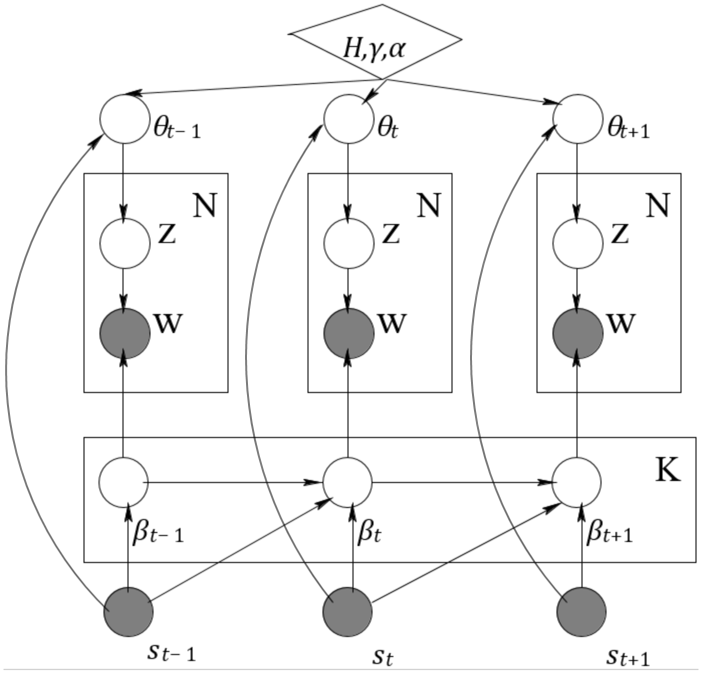

Figure 7. Continuous-time infinite dynamic topic model (ciDTM). Unshaded nodes represent unobserved latent variables, shaded nodes are observed variables, diamonds represent hyperparameters, and plates represent repetition. Symbols in this figure follow the notation in Figure 3, Figure 5, and Figure 6. *H* is the base distribution for the upper level DP, *γ* is the concentration parameter for the upper level DP, *α* is the one for the lower level DP, and s_t is the timestamp for document at time *t*.

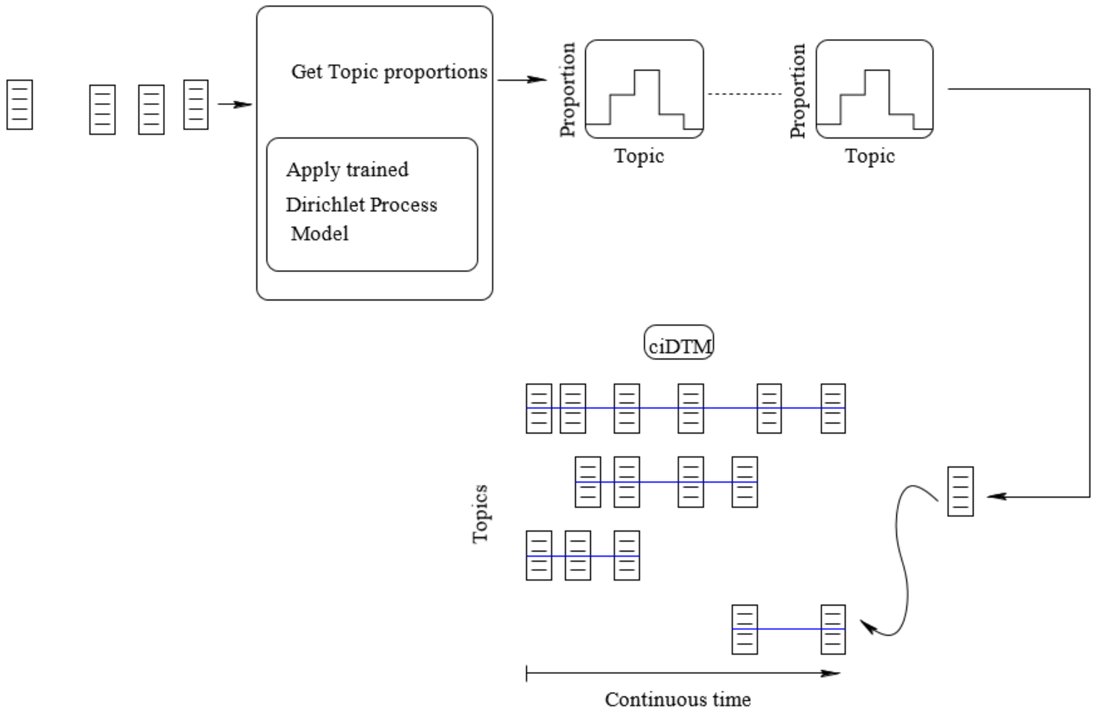

Figure 8. A data flow diagram depicting the workflow for the continuous-time infinite dynamic topic model (ciDTM) working on a single document at a time.

Figure 8 is a data flow diagram that represents the workflow and typical use case for ciDTM. 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. 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:

(4.3)
$$\beta'_k \sim Beta(1,\gamma)$$
$$\beta_k = \beta'_k \prod_{l=1}^{k-1}(1-\beta'_l)$$

(4.4)
$$\phi_k \sim H$$

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k} \qquad (4.5)$$

(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's stick-breaking construction of the DP, again as follows:

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

$$\pi'_{jt} \sim Beta(1, \alpha_0 \qquad) \qquad (4.8)\ ) \quad (4.9)$$

$$\pi_{jt} = \pi'_{jt} \prod_{l=1}^{t-1} (1 - \pi'_{jl} \qquad (4.10)$$

where ψ_{jt} is a document-level atom $G_j = \sum_{t=1}^{\infty} \pi_{jt} \delta_{\psi_{jt}}$ (topic) and π_{jt} is the weight associated with it.

This model can be simplified by introducing indicator variables c_j that are drawn from the weights β:

$$c_{jt} \sim Mult(\beta) \qquad (4.11)$$

The variational distribution is thus given by:

$$q(\beta^0,\pi^0,c,z,\varphi_0) = q(\beta^0)q(\pi^0)q(c)q(z)q(\varphi_0) \qquad (4.12)$$

$${}_{K-1}\, q(\beta^0) = {}^{\mathrm{Y}}\, q(\beta_k{}^0$$

(4.13)

$$|u_k,v_k)$$

$${}_{k=1}$$

$${}_{T-1}$$

$$q(\pi 0) = \mathrm{YY}\ q(\pi_{jt0}\,|a_{jt},b_{jt}) \qquad (4.14)$$

$${}_{j \quad t=1}$$

$$q(c) = \mathrm{YY} q(c_{jt}|\phi_{jt}) \qquad (4.15)$$

$${}_{j \quad t}$$

$$q(z) = \mathrm{YY} q(z_{jn}|\zeta_{jn}) \qquad (4.16)$$

$${}_{j \quad n}$$

$$q(\varphi_0) = {}^{\mathrm{Y}} q(\varphi_k|\lambda_k) \qquad (4.17)$$

$${}_{k}$$

where β^0 is the tuple of corpus-level stick proportions and (u_k,v_k) are parameters for a beta distribution defined over it, $\pi^0{}_j$ is the tuple of document-level stick proportions and (a_{jt},b_{jt}) are parameters for its beta distribution, c_j is the tuple of indicators, φ_0 is the tuple of topic distributions, and z is the tuple of topic indices. 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 D]:

$$\log p(\mathrm{D}|\gamma,\alpha_0,\zeta) \geq \mathrm{E}_q[\log p(D,\beta^0,\pi^0,c,z,\varphi_0)] + H(q) \tag{4.18}$$

$$= \underset{j}{X}\{\mathrm{E}_q[\log(p(w_j|c_j,z_j,\varphi_0)p(c_j|\beta^0)p(z_j|\pi_j^0)p(\pi_j^0|\alpha_0))] \tag{4.19}$$

$$+ H(q(c_j)) + H(q(z_j)) + H(q(\pi_j'))\} \tag{4.20}$$

$$+ \mathrm{E}_q[\log p(\beta^0)p(\varphi_0)] + H(q(\beta^0)) + H(q(\varphi_0)) \tag{4.21}$$

$$= \mathrm{L}(q) \tag{4.22}$$

where $H(.)$ is the entropy term for the variational distribution.

We use coordinate ascent to maximize the log-like likelihood given abpve. Next, given the per-topic word distribution φ_0, I use a Wiener motion process to make the topics evolve over time. We 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:

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

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

$$w_{t,n} \sim \mathrm{Mult}(\pi(\varphi_{t,zt,n})) \tag{4.25}$$

where $\pi(.)$ 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 therefore use approximate inference. For this model, sparse variational inference may be applied.

4. Corpora for Simultaneous Topic Enumeration and Formation (STEF)

In this section we discuss the real-world STEF tasks that necessitate a continuous-time infinite dynamic topic model and derive data requirements for a corpus-based topic modeling system that we developed as an application test bed. We also present a framework for evaluation of the implemented model and competing models.

4.1 Corpus Development

Crtiteria for development of the corpus include:

1. **Purpose:** What is the need for the corpus, in terms of the primary aims of a performance element for collecting it and delivering the results?
2. **Composition:** What kind of documents should this corpus be made of?
3. **Figures of merit:** What makes a good corpus?
4. **Means of production:** How can a good corpus be created? Does the methodology involve manual annotation or automate pre-processing?
5. **Alternative source:** Are there any existing corpora that already fit the determined needs?
6. **Alternative preparation methods:** Can one modify existing corpora to fit the needs?
7. **Technical difficulty:** What are the computational, statistical, and mathematical challenges in creating such a corpus?
8. **Benefits to community:** Who else could benefit from this corpus if it is disseminated?
9. **Restrictions:** Can one freely publish this corpus?

We now define the problem of creating timelines for different news stories which has many potential applications especially in the news media industry. Afterwards, we describe 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.

In order to test the performance of ciDTM and create news timelines we 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 our case, we can crawl newspapers websites, news agencies websites, or news websites. For one of these sources to be considered a valid source for our 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 we 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.

- **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. Such sites are more difficult to crawl and also tend to exhibit more semantic bias, especially political bias.

- **Newspaper websites** tend to fall in the middle between news agencies websites and news websites regarding the diversity, coverage and richness of. 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. However, the paper-published content usually has hard space limitations. This typically leads to an easier crawling task than for news websites.

For more details on these three categories of news sites, we refer the interested reader to Elshamy (2012).

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 our 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, such as the *Newstracker* system being used by BBC News, cannot be used as a gold standard for our 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 seconds. Such a fast-paced publication 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, we obtain its set of related stories. For each one of these stories in turn, we get a set of related stories. This process can be repeated, and we 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.

We create a news corpus by crawling news websites like the British newspaper *The Guardian* website. We 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, we first use the link as the story identifier. We then crawl the main news story text and title, its release date and time, and the set of related news stories. Finally, we 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 we can use them for our gold standard. We repeat this process until we have crawled a predefined number of news stories.

4.2 Experimental Results

For our experiments we used two news agency corpora: the popular Reuters corpus dating from 1987, and a BBC news corpus crawled from the BBC web site in 2012.

4.2.1 Topic Detection and Tracking: STEF Tasks for 1987 Reuters and 2012 BBC Corpora

Reuters corpus. The *Reuters-21578* corpus 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 the following fields:

- **Date:** The date the story was published. This date is accurate to milliseconds. *e.g.*, 26FEB-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 our experiments, we used only **Date** and **Body**.

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 is the case with oHDP, 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.

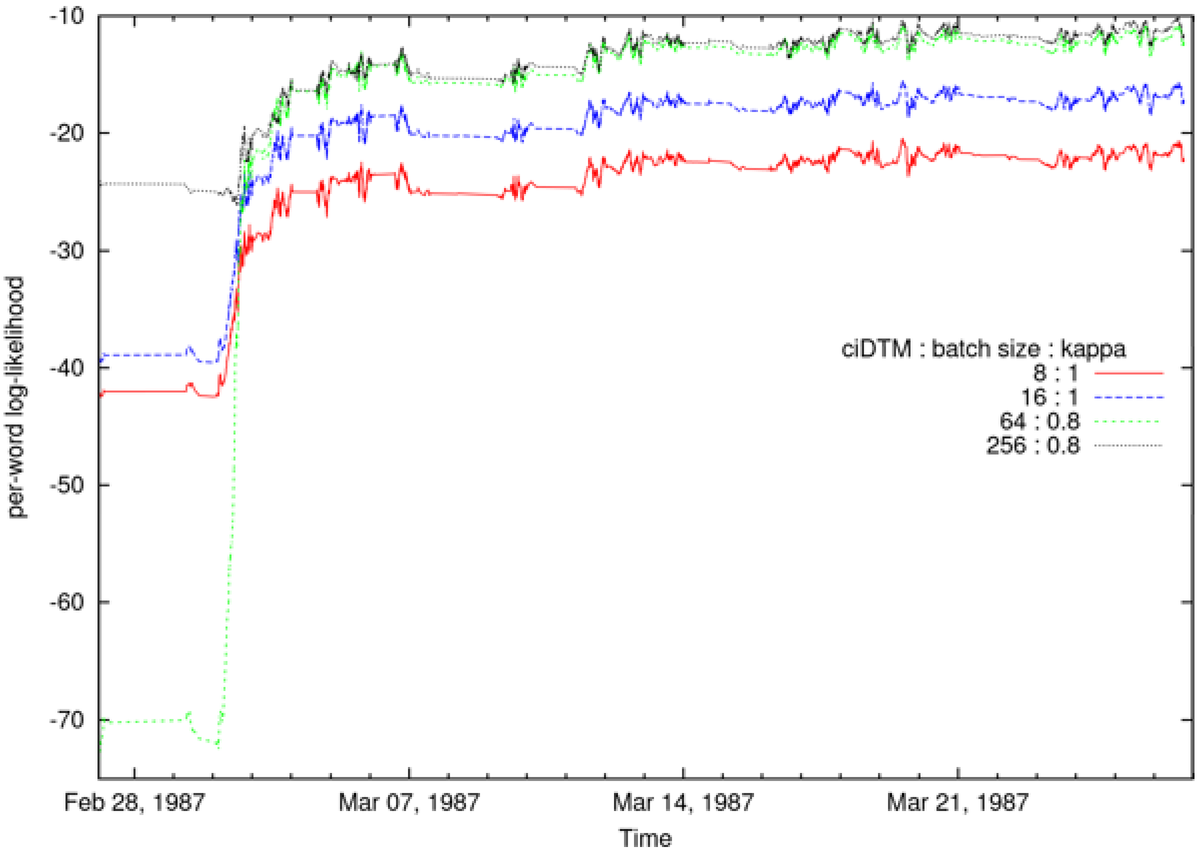

Figure 9. ciDTM per-word log-likelihood for different batch size values using the BBC news corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this 10,000 news stories corpus.

Figure 9 shows the results of an experiment in which four different batch size values were tried.

BBC corpus. The BBC news corpus is made of 10,000 news stories in English we collected by implementing and running a web crawler to scrape documents from the BBC news website[1]. The stories in the corpus cover a time period of about a two and a half years, 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 we collected:

- **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 ...”

[1] http://www.bbcnews.com

- **Related** The ID of a related news story. *e.g.*, http://www.bbc.co.uk/news/world-europe10916011

For the BBC corpus, 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.

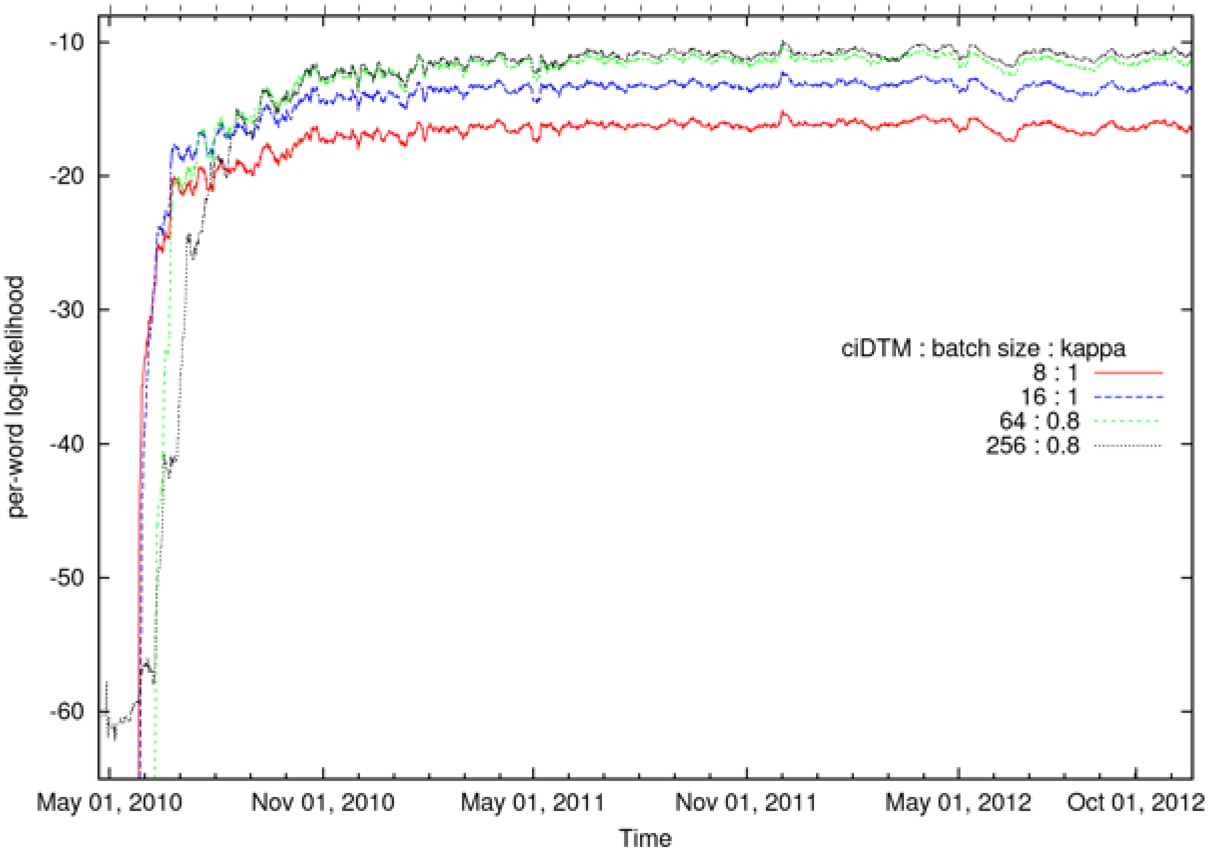

Figure 10. ciDTM per-word log-likelihood for different batch size values using the BBC news corpus. The per-word log-likelihood values shown are the moving average of a set of 100 consecutive documents. This averaging was needed to smooth out the plot. Higher values indicate better model fit on this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

Figure 10 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 outweighted by the longer periods separating model updates.

4.2.2 Timeline Reconstruction: Event Detection Task using BBC Corpus

For the timeline construction task, we manually selected 61 news stories from the BBC news corpus covering the “Arab Spring” events since the inception of the phenomenon in January 2011 until the time of our experiment (November, 2012). We 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.

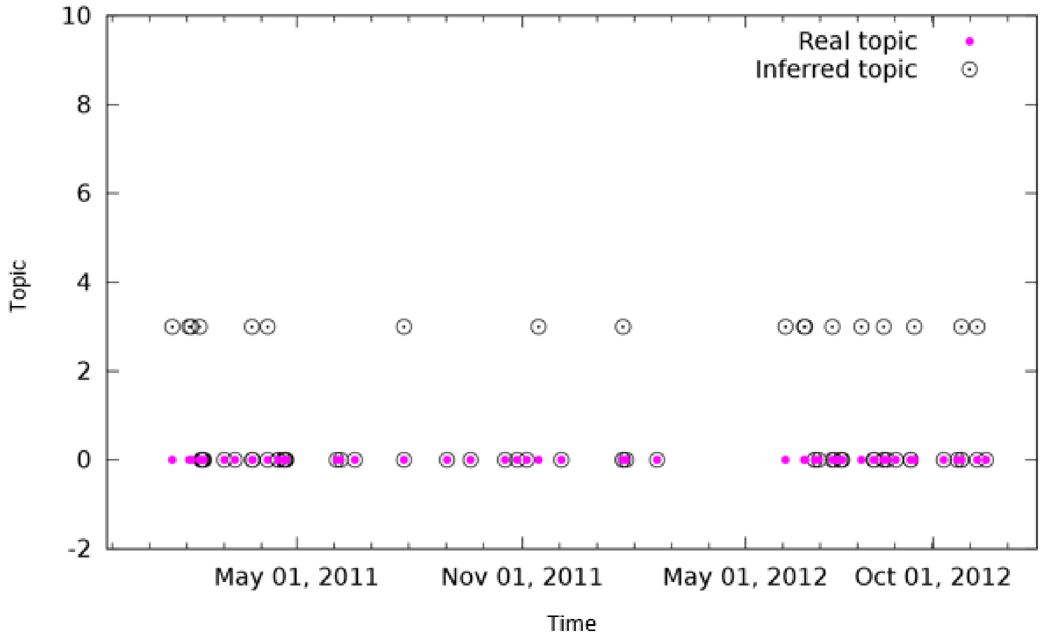

Figure 11. oHDP timeline construction for the Arab Spring topic over 61 documents covering this topic. The independent axis is the time, while the dependent axis is the topic index. It was found that topic 0 represents the Arab Spring events topic. A pink dot • represents the real topic that should be discovered, a black circle with a small dot ⊙ represents the topic/s that was/were actually 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.

We 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. We then manually inspected the inferred topics and found the one that corresponds to the Arab Spring events. Both systems where trained using their best settings found in earlier experiments on the BBC news corpus.

Figure 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 unable to infer the correct topic for three documents in this set. The documents were associated with the accidents and disasters topic instead. Overall, the model was unable to infer the correct topic for 10 out of 61 documents in the Arab Spring set (*i.e.*, it incurred 10 false negatives for class “AS”).

Table 1. Confusion matrix for timeline construction using oHDP. “AS” denotes an Arab Spring topic and “non-AS” a document (news article) whose topic is not an Arab Spring event. The true class is presented in rows (True AS, and non-AS), and Inferred class in columns (Inferred AS and non-AS).

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

Table 1 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%.

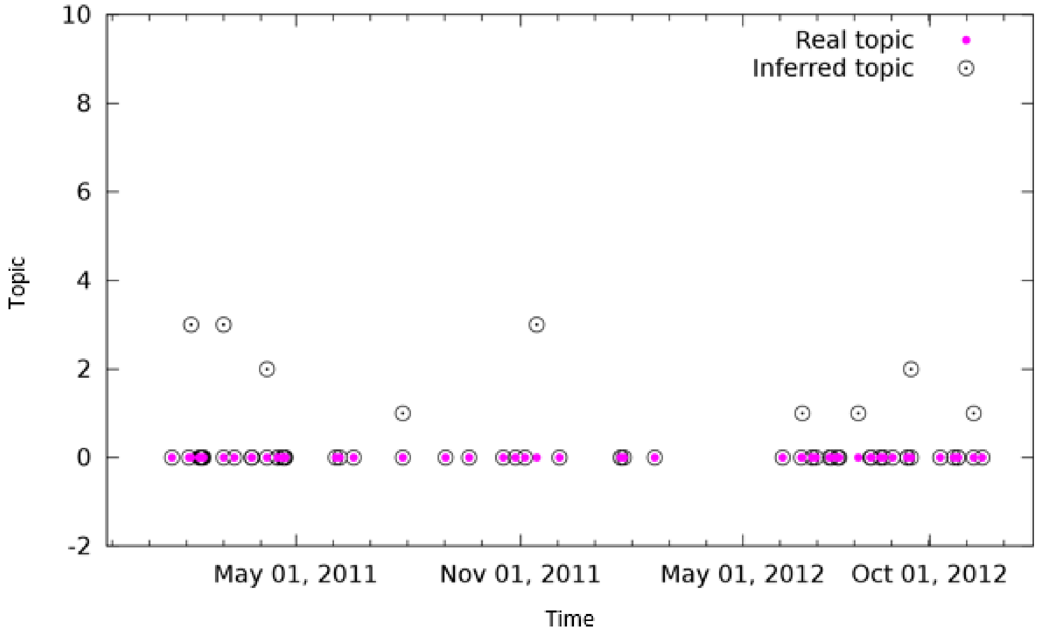

Figure 12. ciDTM timeline construction for the Arab Spring topic over 61 documents covering this topic. The independent axis is the time, while the dependent axis is the topic index. It was found that topic 0 represents the Arab Spring events topic. The same glyphs are used as in Figure 11.

Figure 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 economic and financial damages which the revolution inflicted on the country.

Table 2. Confusion matrix for timeline construction using ciDTM. The row and column label are identical to those for Table 1.

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

Table 2 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 achieves a higher rate of recall without sacrificing precision.

4.2.3 Interpretation of Results

The *per-word log likelihood* achieved by ciDTM is competitive with that of oHDP (discrete-time, unrestricted topic counts) and cDTM (continuous-time, fixed topic counts). On a *timeline construction* task defined over the BBC Corpus, cIDTM performed better than oHDP, achieving a recall boost of 6 out of 10 false negatives without any loss of precision, 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.

Bibliography

Cadez, I., Heckerman, D., Meek, C., Smyth, P., & White, S. (2000). Visualization of navigation patterns on a Web site using model-based clustering. In R. Ramakrishnan, S. J. Stolfo, R. J. Bayardo, & I. Parsa (Ed.), *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000)*, (pp. 280-284). Boston, MA, USA.

Cuzzocrea, A., Song, I.-Y., & Davis, K. C. (2011). Analytics over large-scale multidimensional data: the big data revolution! In A. Cuzzocrea, I.-Y. Song, & K. C. Davis (Ed.), *Proceedings of the ACM 14th International Workshop on Data Warehousing and On-Line Analytical Processing (DOLAP 2011)* (pp. 101-104). Glasgow, UK: ACM Press.

Elger, C. E., & Lehnertz, K. (1998, February). Seizure prediction by non-linear time series analysis of brain electrical activity. *European Journal of Neuroscience, 10*(2), 786–789.

Goldstein, J., & Roth, S. F. (1994). Using aggregation and dynamic queries for exploring large data sets. In E. Dykstra-Erickson, & M. Tscheligi (Ed.), *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)* (pp. 23-29). Boston, MA, USA: ACM Press.

Hall, M., Frank, E., Holmes, G., & Pfahringer, B. (2009, June). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter, 11*(1), pp. 10-18.

Heer, J., Kong, N., & Agrawala, M. (2009). Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI 2009)* (pp. 1303-1312). Boston, MA, USA: ACM Press.

Keim, D. A. (2006). Challenges in Visual Data Analysis. In E. Banissi, K. Börner, C. Chen, G. Clapworthy, C. Maple, A. Lobben, . . . J. Zhang (Ed.), *10th International Conference on Information Visualisation (IV 2006)* (pp. 9-16). London, UK: IEEE Press.

Kumar, N., Keogh, E., Lonardi, S., & Ratanamahatana, C. A. (2005). Time-series bitmaps: a practical visualization tool for working with large time series databases. *Proceedings of the 5th SIAM International Conference on Data Mining (SDM 2005)*, (pp. 531-535). Newport Beach, California, USA.

Mario, C., & Talia, D. (2003, January). The knowledge grid. *Communications of the Association for Computing Machinery, 46*(1), 89-93 .

Monmonier, M. (1990). Strategies For The Visualization Of Geographic Time-Series Data. *Cartographica: The International Journal for Geographic Information and Geovisualization, 27*(1), 30-45.

Steele, J., & Iliinsky, N. (Eds.). (2010). *Beautiful Visualization: Looking at Data through the Eyes of Experts.* Cambridge, MA, USA: O'Reilly Media.

Watson, H. J., & Wixom, B. H. (2007, September). The Current State of Business Intelligence. *IEEE Computer, 40*(9), 96-99 .