Probabilistic Models of Topics and Social Events

Wei Wei

CMU-ISR-16-113

December 2016

School of Computer Science
Institute for Software Research
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Kathleen M. Carley
Tom Mitchell
Alexander J. Smola
Huan Liu, Arizona State University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: Machine Learning, Topic Modeling, Graphical Models, Non-parametric Bayesian, Text Mining

Abstract

Structured probabilistic inference has shown to be useful in modeling complex latent structures of data. One successful way in which this technique has been applied is in the discovery of latent topical structures of text data, which is usually referred to as topic modeling. With the recent popularity of mobile devices and social networking, we can now easily acquire text data attached to meta information, such as geo-spatial coordinates and time stamps. This metadata can provide rich and accurate information that is helpful in answering many research questions related to spatial and temporal reasoning. However, such data must be treated differently from text data. For example, spatial data is usually organized in terms of a two dimensional region while temporal information can exhibit periodicities. While some work existing in the topic modeling community that utilizes some of the meta information, these models largely focused on incorporating metadata into text analysis, rather than providing models that make full use of the joint distribution of meta-information and text.

In this thesis, I propose the event detection problem, which is a multi-dimensional latent clustering problem on spatial, temporal and topical data. I start with a simple parametric model to discover independent events using geo-tagged Twitter data. The model is then improved toward two directions. First, I augmented the model using Recurrent Chinese Restaurant Process (RCRP) to discover events that are dynamic in nature. Second, I studied a model that can detect events using data from multiple media sources. I studied the characteristics of different media in terms of reported event times and linguistic patterns.

The approaches studied in this thesis are largely based on Bayesian non-parametric methods to deal with steaming data and unpredictable number of clusters. The research will not only serve the event detection problem itself but also shed light into a more general structured clustering problem in spatial, temporal and textual data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis I study a clustering problem called *event discovery*, which aims to discover the latent representations of what is happening by learning from a large set of corpus with spatio-temporal meta information. There are many reasons that the event discovery problem is important. First, as a clustering algorithm to capture knowledge representations, the event discovery problem can be useful to summarize the contents of a large chunk of text data with meta information. The summary can represent the major events covered in the text and there is no need to read them one by one as what would need to be done by a human otherwise. Second, the event discovery problem can be useful to government and police department to monitor breaking events that can be harmful to the society. Examples of such events include natural disasters and terrorist attacks, which usually requires government leaders to take timely and decisively actions. The latent clusters learned in this thesis provide real time and accurate representations of the event's spatial, temporal and topical distributions.

There are couple of assumptions that have to be made in order for the algorithms in this thesis to work. I assume that there exist latent event distributions across those spatial, temporal and topical domains. Texts with spatial, temporal meta data are observations drawn from those latent event distributions. This assumption is vital for the algorithms to work and many data sources such as Twitter and newspaper exhibit properties that align with such an assumption. I study three statistical models to discover latent social events and provided scalable inference for each of them. I verified the models by illustrating both qualitative results and numerical results.

To tackle this problem, I first studied a parametric model to detect events on Twitter data to discover independent clusters in Chapter 3. In this model, event clusters are assumed to be static over time no local structures exist for different types of media. The number of clusters in this model is also fixed because of the nature of parametric graphical models. Experiments were conducted on a data set collected over the country of Egypt during the famous Arab Spring revolutions[6]. I showed that events detected using my method successfully matched the records in Wikipedia and official documents from the United Nations. I also illustrated how the learned latent events distributions can be used in supervised settings such as predicting the missing location and time information of the tweets.

To improve the original event detection model and to solve the problems of fixed number of clusters, I studied two improve models, both of which are non-parametric and the complexity of the models can be determined automatically based on the data set. Two non-parametric techniques are used: the Dirichlet Process (DP)[36] and the Recurrent Chinese Restaurant Process (RCRP) [1]. I use Sequential Monte Carlo (SMC) to infer the latent variables for both of the models and the inference algorithms are implemented in parallel. Since SMC scans one data point each time, the algorithms developed for these two models are able to handle streaming data adapt the number of clusters dynamically.

The first improved model is motivated by the fact certain social events exhibit the nature of temporal dynamics and their contents might change over time. Take Superbowl as an example, both spatial and topical concentrations of this event might change over time while they also exhibit certain common properties. In this model, I applied the RCRP framework and used Logistic-normal priors for both the spatial and temporal distributions. When the cluster exist in the previous time step, Logistic-normal prior is centered on the value of cluster at the previous time. Otherwise, the prior is zero centered and no particular dimension is favored. To solve the non-conjugacy problem between the Logistic-normal prior and the Multinomial likelihood, I come up with a solution that outperforms the ones proposed in prior work in terms of prediction results. I also illustrated that the model can generate meaningful results by detecting real world social events that are interpretable. This study of the evolutionary events will be discussed in Chapter 4.

The second improved model extracts social events from multiple data sources and assumes local structures for each of them. Here local structures include media specific temporal and language distributions. Take Twitter and newspaper as an example, the former one usually contains accurate spatial information while the latter one usually has high quality text written by professionals. By utilizing data sets from both end, we are able to generate event clusters that are of higher quality than relying on a specific data source along. We can also study the differences of different media such as the linguistic characteristics as well as the swiftness of when the event is being reported for each media. The model is built on Dirichlet Process that allows the complexity of the model to increase over time. I am able to detect real world social events on four different countries and concluded that social media trend to act faster in events that involve public participations while newspaper reports are usually faster in the case of natural disasters. This work will be discussed in Chapter 5.

Chapter 2

Background

2.1 Event Detections

As most information available on the web does not provide geospatial or temporal information, text based methods represent an important aspect of event detection methodology. Three general types of approaches are surveyed here.

Similarity-based methods are the most common means of detecting events in text. The general idea is to define a similarity metric and compare the pairwise similarity score across documents. Documents that belong to the same event should have high similarity with each other. Otherwise, a new event will be created to maintain high similarity within each event. Several approaches have been proposed. For example, [49] use cosine similarity. Other methods include Hellinger distance [22], Kullback-Leibler divergence [21] and TF-IDF similarity [74].

The second class of methods for detecting events in text are based on abnormality detection of frequent words. For example, [56] monitored the hourly frequency of disaster related keywords such as "alert". The idea was that after normalizing the keyword frequency against on the total number of tweets in each bucketed time slot, one will be able to detect sudden change on those keywords during the major event. Once a major event happened such as an earthquake, the hourly frequency distribution will appear abnormal when compared to historical data, which indicates a potential new event. The authors of [95] uses similar ideas on Twitter sport data set but focuses on the birth of sub-events.

The third type of methods utilize a supervised structured learning algorithm on text data to learn patterns toward the classifications of events. [10], for example, built a Bayesian model to classify a Twitter data set containing labeled 110 music concert events.

Beyond the extraction of events purely from text, there have also been several efforts to incorporate temporal and geospatial information. The authors of [72] analyzed the statistical correlations between earthquake events in Japan and Twitter messages that were sent during the disaster time frame. A linear dynamic system model is used to detect earth quakes. Both [68] and [65] extract events into a hierarchy of types, in part utilizing the temporal information in both the text and the timestamp of the tweet itself. However, their work does not consider the spatial information explicit in geo-spatially tagged tweets.

2.2 Topic Modeling

Topic modeling is a central problem in text mining. In topic modeling, documents are modeled to be a bag-of-words, which ignores the sequences of words and thus retains only the frequency of appearance of words in a document. The objective of topic modeling is to uncover latent representations of document clusters (topics). Several approaches have been proposed, including Latent Semantic Indexing (LSA) [35] which is based on Singular Vector Decomposition (SVD) and Latent Dirichlet Allocation (LDA) [17] which is based on probabilistic graphical models [45]. Here I focus on LDA since it is most relevant to the probabilistic approach I use in this thesis.

In LDA, topics as assumed to be Dirichlet distributed multivariate random variable over the vocabulary set. Each document is assumed to contain words drawn from a mixture of topics. LDA sees important applications in finding topics in documents such as scientific articles [41]. However, just like many statistical learning approaches, its application-agnostic nature allowed it to extend to other areas such as clustering region functions [92] and clustering check-in patterns [46]. The LDA model can be extended with additional meta-data, such as author-topic model [71], relational topic model [28], named entity topic model [62] Syntactic topic model [20], dynamic topic model [15], sentiment topic model [52] and Spatial LDA [82]. The computational intensive nature of LDA leads to many work that improves its efficiency by introducing different sampling techniques such as Gibbs Sampling [41], Sparse-LDA [91], Alias-LDA[51] and light-LDA [93]. Finally, probabilistic models that contains an LDA component but serves other purposes are also proposed. Examples include spatial topic pattern model [44], review aspect modeling and recommendation system [32, 84, 85] and event detection [83].

2.3 Bayesian Non-parametrics

Parametric Bayesian models such as LDA require a fixed number of parameters (e.g. the number of topics), which has to be determined a priori. As with all other Bayesian methods, if the priors are not set correctly, the performance of the model will suffer. Moreover, in a streaming setting where documents are arriving constantly, the dimension of model parameters must increase with the new data. Non-parametric Bayesian approaches can automatically infer an adequate complexity for the model and allow it to grow as new data comes in. There are several Bayesian non-parametric models such as Dirichlet Process [11, 36], Gaussian Process [37, 67], Infinite Hidden Markov model [8] and Polya Trees [57]. I focus on techniques related to Dirichlet Process since they are most related to this thesis.

In a Dirichlet Process (DP), data that fall into the k^{th} cluster have the same parameter β_k. For the i^{th} data point, the conditional probability for its cluster parameter θ_i follows Equation 2.1 [12].

$$\theta_i|\{\theta_{1:i-1}\}, G_0, \alpha \sim \frac{1}{i-1+\alpha} \times \left[\sum_k (n_k^{(i)}\delta(\beta_k) + \alpha G_0\right] \tag{2.1}$$

Here δ is the Dirac delta function and $n_k^{(i)}$ is the number of data points in cluster k before the

i^{th} data point. What Equation 2.1 says is that θ_i has probability proportional to $n_k^{(i)}$ to take one of the existing cluster k with parameter β_k and probability proportional to the dispersion parameter α to take a new cluster parameter generated from the base distribution G_0. The DP starts with 0 clusters and grows as the data exhibit new patterns. This interpretation of DP is known as the Chinese Restaurant Metaphor [5] in that it can be viewed as a brunch of customers (documents) walking into a restaurant with several tables (clusters). The customers can choose to sit on an existing table or create a new table according to the conditional probability in Equation 2.1.

Many non-parametric models related to LDA have been proposed. For example, the Hierarchical Dirichlet Process [77] is a non-parametric extension of LDA. In order to model the nested structures of topics, several non-parametric techniques have been proposed such as the nested Chinese Restaurant Process [40], Nested Chinese Restaurant Franchise Process [4] and Nested Hierarchical Dirichlet Process [63]. There are also several techniques to model with time and topics together in a non-parametric setting. For example, the Recurrent Chinese Restaurant Process [1] and the Dirichlet-Hawkes Process [34].

Chapter 3

Modeling Independent Events

3.1 Introduction

The perpetual availability of online content and our increasing reliance on the Internet have made social networking websites such as Twitter and Facebook an indispensable part of modern social life for many people. As of November 2014, it is estimated that roughly a half billion tweets are generated on a daily basis [1]. The content generated from these social networking/social media sites is not only voluminous; it also contains a selection of information that is new and interesting to individual users, corporate and government actors and researchers alike. This information is useful for many types of analysis, such as sentiment analysis [64] and abnormality detection [79].

One particularly interesting line of work that draws on social media content is the problem of detecting events. In event detection, we wish to uncover abnormal subsets of content that may be referring to a particular occurrence of interest. A significant amount of this work focuses purely on the analysis of the textual content of social media messages [10, 49]. While the inference of topical focus is an interesting problem in its own right, the idea that topical coherence is a signal for an "event" is slightly misleading. Such algorithms are essentially detecting topics, which are words that clustered together, rather than any coherent subset of content that has a unique geo-temporal realization, one we would expect of a typical event. For example, topics uncovered that are broadly related to online games and jokes have little or no link to the physical world and thus are difficult to consider events.

Having realized this, recent work has begun to focus on the geo-temporal aspects of event detection [72]. However, much of this work fails to utilize the textual information that previous authors have capitalized on, information that is vital in interpreting the topical focus of a particular event [68]. For example, events that occur in a residence and a nearby night club at the same time will contain the same geospatial and temporal information but are, of course, different in important ways. A good definition of an event should thus contain a geographical approximation of where the event is happening, a temporal range over which the event lasts and also a specific set of words and/or phrases that can be used to describe what the event is about.

In this chapter, we develop a probabilistic graphical model that learns the existence of events

[1] http://www.internetlivestats.com/twitter-statistics

based on the location, time and text of a set of social media posts, specifically tweets. An event is described by a central geographical location and time, a variance in space and time and a set of words (a topic) that is representative of the terms that can be used to describe this event. By incorporating both a central location and time and a variance around it, we account for the fact some events are more concentrated within a specific region and time (e.g. a marathon) while others might be distributed across a broader area in time and/or space (e.g. Occupy Wallstreet). The use of a set of words that are frequently used in tweets from or about the event allows us to incorporate topic modeling to extract information from the actual tweet text, from which an understanding of the focus of the event can be derived.

Our contributions are twofold. First, we build an event detection model that successfully discovers latent events being discussed at different points in time and at different locations in a large, geo-tagged Twitter data set. We demonstrate the model's abilities by applying our method to a Twitter data set collected in an Arab country during a time period where demonstrations and social movements were frequent. Second, we build a location and time prediction tool based on our learned model that allows us to accurately predict the location or time of a tweet (when this information is held out) with considerably more accuracy than several baseline approaches.

3.2 Related Work

The problem of event detection is well studied. Here, we provide a brief survey of relevant methods, touching on a variety of approaches that have been taken in studying the problem.

3.2.1 Events Extraction from Text

As most information available on the web does not provide geospatial or temporal information, text based methods represent an important aspect of event detection methodology. Three general types of approaches are surveyed here.

Clustering is one of most important techniques in dealing with the event detection problem using text. Clustering approaches attempt to find latent events by uncovering common patterns of texts that appear in the document set. These efforts generally fall into two distinct types of approaches: similarity based methods and statistical ones. Similarity-based methods usually compare documents by applying metrics such as cosine similarity [49]. These models are usually efficient but ignore statistical dependencies between both observable and latent underlying variables. A statistical method such as a graphical model [10] can incorporate more complicated variable dependencies and hierarchical structure to event inference.

Another type of event detection model utilizes the fact that the arrival of new events will change the distribution of the existing data. Such approaches are thus concerned with developing criterion for detecting abnormal changes in the data. For example, Matuszka et al. (2013) assumes a life cycle for each possible keyword for an event, penalizing the term if it appears consistently in the data. The result is an event defined by keywords that only appear in some specific subset of the observed data. Zubiaga et al. (2012) use techniques such as outlier detection to detect abnormalities in the data set which is considered a potential consequence of a new event.

The third type of work defines events indirectly by linking documents together. Models such as the one proposed by Štajner and Grobelnik (2009) define each document as a node in a graph and then build connections between them once they are classified as being a part of the same event. Finally, there is also a large amount of work focusing on using information retrieval techniques such as TF-IDF as features to extract events[22].

3.2.2 Events Extractions from Space and Time

Beyond the extraction of events purely from text, there have also been several efforts to incorporate temporal and geospatial information. Sakaki et al. (2010) analyzed the statistical correlations between earthquake events in Japan and Twitter messages that were sent during the distaster time frame. An abrupt change of volume of tweets in a specific geo region indicated a potential disaster in that area. Hong et al. (2012) constructed a probabilistic graphical model that contains both a geographical component and a topical component to discover latent regions from Twitter data. Their efforts, however, are not strictly focused on event detection, as they do not consider the temporal domain. In contrast, Ritter et al. (2012) and Panisson et al. (2014) extract events into a hierarchy of types, in part utilizing the temporal information in both the text and the timestamp of the tweet itself. However, their work does not consider the spatial information explicit in geospatially tagged tweets.

3.2.3 Graphical Models and Sampling Techniques

Graphical models are powerful tools that can be used to model and estimate complex statistical dependencies among variables. For a general overview, we refer the reader to [45], which contains a much richer discussion than is possible here. By constructing statistical dependencies among both observed and latent variables, graphical models can be used to infer latent representations that are not observed in the data. Latent Dirichlet allocation [17], used to discover such latent topics/events from text, is perhaps the most widely known example in this area.

One issue often raised in graphical models is the difficulty in estimation. As the complexity of the model increases, exact inference become difficult or even impossible. Various sampling strategies such as Gibbs sampling [26] has thus been developed to find approximate solutions.

3.3 Model

We use a probabilistic graphical model to characterize the relationship between events and tweets (referred to here as documents). Using plate notation, Figure 3.1 illustrates the structure of the model. Note that there are D documents and E events, where E is a value pre-determined by the researcher. The model has three major components. First, an **event model** contains information about a specific event, such as the parameters that characterize its spatial and temporal distributions. Second, a **document model** contains the location, time and event index of each document. Third, there is a **language model**, which contains information about the topical content of the documents. Table 3.1 gives a summary of all notation that will be used as we describe the model in this section.

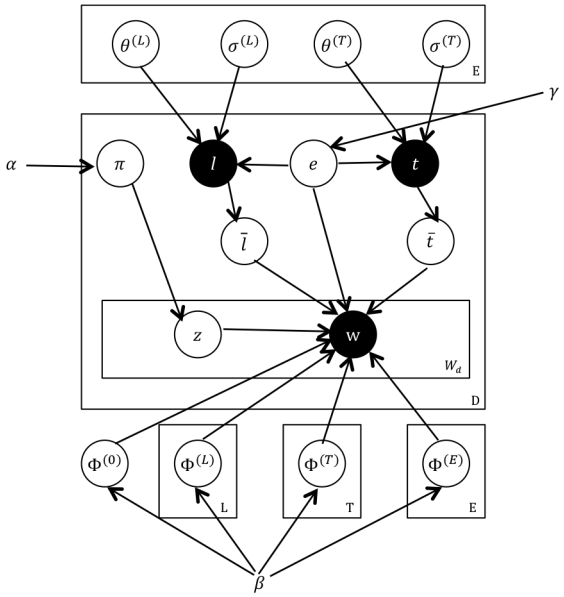


Figure 3.1: Illustrations of the model in plate notations

3.3.1 Event Model

An important observation incorporated into our model is that events are in many ways natural extensions of topics; events have a topical focus but also include a spatial and temporal region in which they are likely to occur. We thus assume events are defined by three things. First, each event has a geographical center $\theta_e^{(L)}$ as well as a geographical variance controled by a diagonal covariance matrix with each value defined by $\sigma_e^{(L)}$. The location of a document that belongs to event e is assumed to be drawn from a two dimensional Gaussian distribution governed by these parameters.

$$l \sim \mathrm{N}(\theta_e^{(L)}, I \cdot \sigma_e^{(L)}) \tag{3.1}$$

Second, each event is defined by a temporal domain. Similar to the spatial distribution of an event, event time is also modeled as a Gaussian distribution, except with mean $\theta_e^{(T)}$ and a variance of $\sigma_e^{(T)}$:

$$t \sim \mathrm{N}(\theta_e^{(T)}, I \cdot \sigma_e^{(T)}) \tag{3.2}$$

The mean and standard deviations of both Gaussian distributions are latent variables and will need to be inferred by the model. Finally, events are determined by a topic (or distribution over words) that characterizes the event. The details of this are implemented within the document model and Language model, discussed later in this section.

3.3.2 Document Model

A document contains the information we obtain for a specific tweet. In our model, we only consider tweets that have both a geo-location tag (latitude/longitude pair) l and a time stamp t.

Tweets also consist of a word array w which contains the actual words that appear in the tweet.

Several latent variables are also present in the document model. First, an event identity e defines which single event out of the E possible events in the event model that this specific document belongs to. We assume a multinomial prior γ for each e in each document.

$$e \sim \text{Mult}(\gamma) \tag{3.3}$$

Second, each word w_i in the document has a corresponding category variable z_i that determines which of 4 categories of topics this word has been drawn from. Category "0" is a global category, which represents global topics that frequently occur across all tweets. Category "L" defines a set of regionally specific topics that are specific to particular geospatial subareas within the data. Category "T" represents a set of temporally aligned topics that contain words occurring within different temporal factions of the data. Category "E" defines topics that are representative of a particular event e, distinct from both other events and more specific to the event than topics in the other categories. By controlling for global, temporal and spatial topics, these event-specific topics allow us to uncover the defining terms of this particular event beyond those specific to a general spatial or temporal region. The variable z is controlled by a multinomial distribution whose parameter is a per document category distribution π:

$$z \sim \text{Mult}(\pi) \tag{3.4}$$

For each document a π is generated by a prior α from a Dirichlet distribution:

$$\pi \sim \text{Dir}(\alpha) \tag{3.5}$$

To index into the topics of the location and time categories, each location l and time t is converted into a location index $\bar{l}$ and a time index $\bar{t}$, respectively. These conversions are conducted by applying two functions $f(l)$ and $g(t)$. These resulting indices are used for the language model to retrieve the corresponding topics from these categories in a manner that will be introduced later.

$$\bar{l} = f(l) \tag{3.6}$$

$$\bar{t} = g(t) \tag{3.7}$$

3.3.3 Language Model

The language model defines how words within a document are drawn from topics (within specific categories) based on the full set of parameters associated with the document. Topic distributions for each category are generated using a Dirichlet prior β:

$$\Phi_*^{(*)} \sim \text{Dir}(\beta) \tag{3.8}$$

Each topic contains the probability of each word in the vocabulary occurring within it. While this is the traditional representation of LDA, note that our approach is a generalization of the original model [17], since now topics are also hierarchically organized by the four different

categories. For a model with one global topic, L location topics, T time topics and E event topics, the total number of topics across the four categories is thus $K = 1 + L + T + E$.

Each word w_i is chosen from a corresponding topic based on its category variable z and the corresponding geo, temporal and event indices $\bar{l}$, $\bar{t}$ and e, respectively, depending on which category is being used. This is represented mathematically in Equation 3.9 below:

$$\begin{aligned} &P(w_i|\bar{l}, \bar{t}, e, z_i, \Phi^{(0)}, \Phi^{(L)}, \Phi^{(T)}, \Phi^{(E)}) \\ &\quad = P(w_i|\Phi^{(0)})^{I(z_i=0)} \cdot P(w_i|\Phi^{(L)}, \bar{l})^{I(z_i=L)} \cdot \\ &\quad\quad P(w_i|\Phi^{(T)}, \bar{t})^{I(z_i=T)} \cdot P(w_i|\Phi^{(E)}, e)^{I(z_i=E)} \end{aligned} \tag{3.9}$$

3.3.4 Spatial and Temporal Boundaries

To generate the location index (i.e. $\bar{l}$) and time index (i.e. $\bar{t}$), we need to define two transformation functions that map from a real vector space to an integer space. To do so, we first divide the geographical and temporal space into a lattice within a pre-determined boundary. For geospace, a preset boundary $B^L = (x_{low}, x_{hi}, y_{low}, y_{hi})$ is determined based on the data. The geoarea is then divided evenly by the number of locations L to form a $\sqrt{L} \times \sqrt{L}$ square lattice. Each cell in the lattice has a unit length of $U^L = (x, y)$, with $U^L_x = (B^L_{x_{hi}} - B^L_{x_{low}})/\sqrt{L}$ and $U^L_y = (B^L_{y_{hi}} - B^L_{y_{low}})/\sqrt{L}$ respectively. The transformation function for location data $f(l)$ is then defined in Equation3.10:

$$f(l) = \lfloor (l_x - B^L_{x_{low}})/U^L_x \rfloor * \sqrt{L} + \lfloor (l_y - B^L_{y_{low}})/U^L_x \rfloor \tag{3.10}$$

Similar to the way that l is mapped to $\bar{l}$, a function that maps t into an index space $\bar{t}$ is also defined in equation 3.11. Here we treat t as a real valued scalar bounded in range from B^T_x to B^T_y. A unit length U^T is also calculated to be the unit length of each time cell in the lattice, which is $(B^T_{hi} - B^T_{low})/T$.

$$g(t) = \lfloor (l - B^T_{low})/U^L \rfloor \tag{3.11}$$

In our model we treat the timestamp of a document as a real-valued variable by dividing the UNIX time by the number of seconds in a month. By doing this we converted the information so that tweets are represented by a real-valued variable that defines the month and year in which they occur. This meets the requirement of the Gaussian distribution in which we used to model the temporal span of a particular event.

3.3.5 Generative Model

The graphical model we defined above can be used as a generative model that produces new tweets that have a geo coordinate, a time stamp and a set of words constituting the text of the message. The generative process is as follows:

- Pick an event $e \sim \mathrm{Mult}(\gamma)$.
- Pick a location $l \sim \mathrm{N}(\theta^{(L)}_e, \sigma^{(L)}_e)$
- Pick a time $t \sim \mathrm{N}(\theta^{(T)}_e, \sigma^{(T)}_e)$
- Pick a category distribution $\pi \sim \mathrm{Dir}(\alpha)$
- For each word w_i, first pick $z_i \sim \mathrm{Mult}(\pi)$ then pick $w_i \sim \Phi^{(*)}$

Table 3.1: Notations

Symbol	Size	Comments
D	1	number of documents
L	1	number of location plates
T	1	number of time plates
E	1	number of events
Z	1	number of topic categories
K	1	number of topics
V	1	number of vocabularies
W_d	1	number of words in document
l	D $\times 2$	location lat and lon
t	D	timestamps
e	D	event index
w	W_d	word in a document
$\bar{l}$	D	location index of a document
$\bar{t}$	D	time index of a document
$\theta^{(L)}, \sigma^{(L)}$	E	mean and sd of event locations
$\theta^{(T)}, \sigma^{(T)}$	E	mean and sd of event time
z	W_d	topic category of word
π	D $\times$ Z	category distribution
Φ	K $\times$ V	word distribution for topics
α	Z	dirichlet prior for π
β	V	dirichlet prior for Φ
γ	E	multinomial prior for e
O	-	Observed variables
Ω	-	latent variables solved in E step
Θ	-	latent variables solved in M step

3.4 Model Inference

Given the number of hidden variables as well as the hierarchical structure of the model, exact inference is intractable. Instead, we use a Gibbs-EM algorithm [7, 81] to infer the model parameters. Before we detail the inference procedure, we clarify three pieces of notation, O, Ω and Θ, that define the sets of variables we are concerned with during the inference procedure. The set $O = \{l, t, w\}$ defines the set of observed variables. The set $\Omega = \{e, z, \pi, \Phi^{(0)}, \Phi^{(L)}, \Phi^{(T)}, \Phi^{(E)}\}$ defines variables that will be solved during the E stage of the algorithm. Variables falling into this set are mainly those related to the language model. The variable $\Theta = \{\theta_L, \theta_T, \sigma_L, \sigma_T\}$ is a set of parameters that will be estimated during the M step. Note that we do not perform inference on the Bayesian hyper parameters $\{\alpha, \beta, \gamma\}$, treating them as static constants to be defined by the researcher. To avoid confusions, we have omitted all the Bayesian hyper parameters in our equations and we will follow this convention in the rest of the chapter.

3.4.1 E Step

During the Expectation ("E") step, we assume that parameters in Θ are already known as the result of a previous Maximization ("M") step. We then use Gibbs sampling to generate samples for the parameters in Ω over a number of Gibbs iterations and use the average of these samples to approximate the expectation of the E step. Before we do this, however, we first integrate out $\Phi^{(*)}$ and π, resulting in a more efficient collapsed Gibbs sampling problem. Equation 3.12 gives the collapsed distribution we are interested in sampling from. Here Γ is the gamma function and $n_{d,r}^{z,k}$ denotes the number of times that a document d has a word r that falls into topic k of category z. If any of d, r, k or z are replaced by "*", the value should be interpreted as one which takes the sum over this particular variable. Note again that in contrast to the standard LDA model, here

we need to pay attention to both topic k and the category z.

$$
\begin{aligned}
P(z, e|\Theta, O) = \int_{\phi^{(0)}} \int_{\phi^{(L)}} \int_{\phi^{(T)}} \int_{\phi^{(E)}} \int_{\pi} & \\
& P(z, e, \pi, \Phi^{(0)}, \Phi^{(L)}, \Phi^{(T)}, \Phi^{(E)}|\Theta, O) \\
= \prod_{d=1}^{D} \frac{\Gamma(\sum_{z=1}^{Z} \alpha_z)}{\prod_{z=1}^{Z} \Gamma(\alpha_z)} \frac{\prod_{z=1}^{Z} \Gamma(n_{d,*}^{z,*} + \alpha_z)}{\Gamma(\sum_{z=1}^{Z} n_{d,*}^{z,*} + \alpha_z)} \times & \\
\frac{\Gamma(\sum_{r=1}^{V} \beta_r)}{\sum_{r=1}^{V} \Gamma(\beta_r)} \frac{\sum_{r=1}^{V} \Gamma(n_{*,r}^{0,1} + \beta_r)}{\Gamma(\sum_{r=1}^{V} n_{*,r}^{0,1} + \beta_r)} \times & \\
\prod_{\bar{l}=1}^{L} \frac{\Gamma(\sum_{r=1}^{V} \beta_r)}{\sum_{r=1}^{V} \Gamma(\beta_r)} \frac{\sum_{r=1}^{V} \Gamma(n_{*,r}^{L,\bar{l}} + \beta_r)}{\Gamma(\sum_{r=1}^{V} n_{*,r}^{L,\bar{l}} + \beta_r)} \times & \\
\prod_{\bar{t}=1}^{T} \frac{\Gamma(\sum_{r=1}^{V} \beta_r)}{\sum_{r=1}^{V} \Gamma(\beta_r)} \frac{\sum_{r=1}^{V} \Gamma(n_{*,r}^{T,\bar{t}} + \beta_r)}{\Gamma(\sum_{r=1}^{V} n_{*,r}^{T,\bar{t}} + \beta_r)} \times & \\
\prod_{e=1}^{E} \frac{\Gamma(\sum_{r=1}^{V} \beta_r)}{\sum_{r=1}^{V} \Gamma(\beta_r)} \frac{\sum_{r=1}^{V} \Gamma(n_{*,r}^{E,e} + \beta_r)}{\Gamma(\sum_{r=1}^{V} n_{*,r}^{E,e} + \beta_r)} &
\end{aligned}
\tag{3.12}
$$

Word Category

The word category variable z is sampled for each word in each document. The conditional probability of a specific category for word n in document d given all the other variables is proportional to the conditional probability given in Equation 3.13. While space constraints do not allow us to present the full derivation of the conditional probability, ideas utilized in the proofs of the original LDA algorithm in Griffiths and Steyvers (2004) can be directly applied to our efforts to derive the equation.

$$
\begin{aligned}
& P(z_{(d,n)} = z)|z_{\text{-}(d,n)}, \Theta, O) \\
& \quad \propto P(z_{(d,n)} = z), w_{\text{-}(d,n)}, w|\Theta, O) \\
& \quad \propto (n_{d,*}^{z,*-(d,n)} + \alpha_k) \frac{n_{*,r}^{z,*-(d,n)} + \beta_r}{\sum_{r=1}^{V} n_{*,r}^{z,*-(d,n)} + \beta_r}
\end{aligned}
\tag{3.13}
$$

Category and Word Distribution

After the category variable z is sampled for each word in each document in the data, we update all word distributions $\Phi^{(*)}$ as well as the category distribution π for each document according to Equation 3.14 and Equation 3.15. Again, while proofs are omitted, similar proofs can be found in Griffiths and Steyvers (2004). One thing worth noticing, however, is that $\pi_{d,z}$ is a bit different from its counterpart $\theta_{d,k}$ in the classic LDA model because of the second dimension k, which is a topic index in the classic LDA. In the present model, this value is changed to z, thus representing

a draw from a category rather than a topic.

$$\Phi_{k,v}^{(i)} = \frac{n_{*,v}^{i,k} + \beta_v}{\sum_{v=1}^{V} n_{*,v}^{i,k} + \beta_v} \tag{3.14}$$

$$\pi_{d,z} = \frac{n_{d,*}^{z,*} + \alpha_z}{\sum_{d=1}^{D} n_{d,*}^{z,*} + \alpha_z} \tag{3.15}$$

Event Index

In addition to sampling the category variables and distributions over the categories, we also must sample the event index e for each document d. The conditional probability for sampling the event index for a specific document based on all other variables is given in Equation 3.16. It is determined by three terms: a prior multinomial distribution on e, two Gaussian distributions, one each on location and time, and a term defining the joint likelihood of each word in the tweet. Observing that this expression can be further simplified and only those words w_i with $z_{w_i} = E$ are actually affecting the probability of sampling e, we are left with Equation 3.16

$$\begin{aligned} & P(e_d|\Omega \backslash e_d, \Theta, O) \\ & \quad \propto \prod_{i;z_i=e} P(w_i|z_i, \Phi^{(z_i)}) \cdot P(l|\theta_e^{(L)}, \sigma_e^{(L)}) \cdot \\ & \quad P(t|\theta_e^{(T)}, \sigma_e^{(T)}) \cdot P(e_d|\gamma) \\ & \quad \propto \frac{1}{\sigma_e^{(L)} \sigma_e^{(T)}} \cdot \gamma(E=e) \cdot \prod_{i;z_i=e} \Phi^{(e)}(w=w_i) \cdot \\ & \quad e^{-\frac{1}{2}[\frac{(L-\theta_e^{(L)})^T (L-\theta_e^{(L)})}{{\sigma_e^{(L)}}^2} + \frac{(T-\theta_e^{(T)})^T (T-\theta_e^{(T)})}{{\sigma_e^{(T)}}^2}]} \end{aligned} \tag{3.16}$$

3.4.2 M step

In the M step, we treat all the variables in Θ as parameters and estimate them by maximizing the likelihood function. Since we use Gibbs sampling in the E step, the likelihood function is an average over all samples drawn from the E step.

For each Gibbs step s we use a superscript to annotate the variables that are drawn from this specific step. The objective function of the M step $Q(\Theta)$ can be written in Equation 3.17. The goal of this M step is to find the latent variables in Θ that maximize this objective function. To achieve better optimization results, we add an L2 penalty term to the location and time deviations in our objective function in addition to the log likelihood. The penalty term has a factor $(1 + r_e)$, where r_e is the ratio of documents that belong to event e. If the ratio r_e for a specific event is high, it will receive a stronger penalty in the size of its spatial and temporal deviations, causing

these variances to be restricted.

$$
\begin{aligned}
Q(\Theta) =& \frac{1}{S}\sum_{s=1}^{S} log(P(O, \Omega^{(s)}|\Theta^{(t)})) \\
&+ \frac{1}{2}\lambda((||\sigma^{(L)}||_2^2 + ||\sigma^{(T)}||_2^2)(1 + r_e)) \\
\propto& \frac{1}{S}\sum_{s=1}^{S}\sum_{d=1}^{D}\Big[- (log(\sigma^{(L)}_{e_d^{(s)}}) + log(\sigma^{(T)}_{e_d^{(s)}})) \\
&- 0.5(\frac{||l_d - \theta^{(L)}_{e_d^{(s)}}||}{\sigma^{(L)^2}_{e_d^{(s)}}} + \frac{||t_d - \theta^{(T)}_{e_d^{(s)}}||}{\sigma^{(T)^2}_{e_d^{(s)}}})\Big] \\
&- \frac{1}{2}\lambda((||\sigma^{(L)}||_2^2 + ||\sigma^{(T)}||_2^2)(1 + r_e))
\end{aligned}
\tag{3.17}
$$

Event Centers

Event centers for both location and time can be estimated in a straightforward manner by maximizing the objective function.

$$
\hat{\theta_e^{(L)}} = \frac{\sum_s \sum_{d;e_d^{(s)}=e} l_d}{\sum_s \sum_{d;e_d^{(s)}=e}} \tag{3.18}
$$

Similarly, we can also acquire a MLE estimation for $\hat{\theta_e^{(T)}}$:

$$
\hat{\theta_e^{(T)}} = \frac{\sum_s \sum_{d;e_d^{(s)}=e} t_d}{\sum_s \sum_{d;e_d^{(s)}=e}} \tag{3.19}
$$

Event Variance

In the estimation of the variance in space and time for each event, the penalty term we have introduced means that we can no longer use the MLE to find an optimal value for them. While this complicates inference, the penalty term is an important part of the model. It is introduced because in model development, we observed that as the number of EM steps increased, larger events tended to rapidly acquire more documents during training. This, in turn, increases the variance of these events to a value larger than we would expect to see for a spatially constraint event. This situation becomes worse over time and eventually these events come to dominate the analysis. The introduced L2 penalty restricts this from occurring.

To solve for the variances, we use a gradient descent approach to find the optimal value. In order to do so, we take the derivative of the EM objective function and acquire the gradient of the event deviations in Equation 3.20 and Equation 3.21. We then apply a standard gradient descent

algorithm.

$$\frac{\partial Q(\Theta)}{\partial \sigma_e^{(L)}} = \frac{\sum_s \sum_{d:e_d=e} \frac{-1}{\sigma_e^{(L)}} + \frac{||l-\theta_e^{(L)}||}{{\sigma_e^{(L)}}^3} - \lambda \sigma_e^{(L)}(1+r_e)}{S} \tag{3.20}$$

$$\frac{\partial Q(\Theta)}{\partial \sigma_e^{(T)}} = \frac{\sum_s \sum_{d:e_d=e} \frac{-1}{\sigma_e^{(T)}} + \frac{||t-\theta_e^{(T)}||}{{\sigma_e^{(T)}}^3} - \lambda \sigma_e^{(T)}(1+r_e)}{S} \tag{3.21}$$

Initializations

Several variables need to be properly initialized in order for the EM algorithm to converge to the correct distribution. The parameters z and e are initialized randomly within their domains. The variables $\theta^{(L)}$ and $\theta^{(T)}$ are initialized by learning a kernel density estimator from the data first and then drawing e samples from it. This initialization gives areas in space and time where tweets are concentrated a higher chance of becoming centers in location or time, respectively. Finally, the variables $\sigma^{(L)}$ and $\sigma^{(T)}$ are generated from a uniform distribution from 0 to 1.

3.4.3 Prediction

One of the most important applications of the model proposed here is to predict the location and time of tweets based on the words contained within them. To achieve this goal, we use another EM algorithm again to infer the hidden variables as well as the variable(s) we are interested in predicting. In the prediction setting, event specific parameters θ and σ and topic categories $\Phi^{(*)}$ are already trained and our goal is to infer z,e and either l, t or w given some or all of the other variables.

Category Variable and Event Index

In our prediction EM algorithm, we estimate the category variable z and the event index e in the E step. This is almost the same process as the one in the training, as all other variables are again fixed. The only difference is that during the training stage, $n_{d,i}^{z,k}$ is initialized according to a randomly generated z and e while in the prediction stage these variables are the result of a trained model.

Predict Location and Time

To predict location and time, we use the samples generated from the E step to make a point inference on one or both, depending on the task at hand. As opposed to the M step in the training stage, in our prediction task all event variables have already been learned and our goal is to estimate l and t instead. Equation 3.22 is the objective function for both l and t. Utilizing the fact that the addition of several Gaussian distributions is proportional to another Gaussian distribution, the summation term for the location and time distributions can each be absorbed into a single Gaussian distribution. The part of the likelihood function that contains the summation

of word probabilities can also be simplified to consider only those words with topics related to either L or T. This results in an objective function that has a location component and a time component, each of which contains a Gaussian term and a grid density term.

$$
\begin{aligned}
Q'(l,t) &\propto \frac{1}{S}\sum_{s=1}^{S}[logP(l|\theta^{(L)}_{e^{(s)}},\sigma^{(L)}_{e^{(s)}}) + logP(t|\theta^{(T)}_{e^{(s)}},\sigma^{(T)}_{e^{(s)}}) \\
&\quad + \sum_{i}^{W} logP(w_i|\Phi^{(z_i)},\bar{l},\bar{t},e^{(s)})] \\
&\propto logP(l|\theta^{(L)}_{*},\sigma^{(L)}_{*}) + \frac{1}{S}\sum_{s=1}^{S}\sum_{i;z_i=L} log\Phi^{(L)}_{\bar{l}}(w_i) \\
&\quad + \underbrace{logP(t|\theta^{(T)}_{*},\sigma^{(T)}_{*})}_{\text{Gaussian Term}} + \underbrace{\frac{1}{S}\sum_{s=1}^{S}\sum_{i;z_i=T} log\Phi^{(T)}_{\bar{t}}(w_i)}_{\text{Grid Density Term}} \qquad (3.22)
\end{aligned}
$$

$$
\text{Where} \quad \theta^{(L)}_{*} = \frac{\sum_s \frac{\theta^{(L)}_{e^{(s)}}}{\sigma^{(L)\,2}_{e^{(s)}}}}{\sum_s \frac{1}{\sigma^{(L)\,2}_{e^{(s)}}}}, \quad \theta^{(T)}_{*} = \frac{\sum_s \frac{\theta^{(T)}_{e^{(s)}}}{\sigma^{(T)\,2}_{e^{(s)}}}}{\sum_s \frac{1}{\sigma^{(T)\,2}_{e^{(s)}}}},
$$

$$
\sigma^{(L)2}_{*} = \frac{S}{\sum_s \frac{1}{\sigma^{(L)\,2}_{e^{(s)}}}} \quad \text{and} \quad \sigma^{(T)2}_{*} = \frac{S}{\sum_s \frac{1}{\sigma^{(T)\,2}_{e^{(s)}}}}
$$

Speeding Up the Optimization

From Equation 3.22, we observe that the estimation of l and t can be done independently, as the objective functions of each entity are absolved of terms from the other. However, to infer either l or t based on the objective function is difficult using conventional optimization methods such as gradient descent since it involves optimizing an objective function that is not continuous. This occurs because the transformation from l and t to $\bar{l}$ and $\bar{t}$ makes the objective function no longer differentiable. Search based optimization techniques can still be applied but are exceedingly slow.

We thus develop a method particular to our specific issue that can estimate l and t rapidly. To see how we can speed up the optimization, observe that the grid density term in Equation 3.22 is fixed when variables fall within a single grid cell. For example for all l such that $\bar{l}$ are the same, these l will fall into the same cell. For all variables falling into the same cell, it is up to the Gaussian term to determine the optimal value. For each grid cell, if the Gaussian center falls outside of it, the optimal point within the cell is the point along the cell boundary that is closest to the Gaussian center. If the Gaussian center falls inside of the grid cell, the optimal point will be the Gaussian center. Using the fact, we can effectively reduce the complexity of the optimization to a linear time algorithm in the number of squares in the location lattice, L when evaluating l or linear to the number of elements in the temporal lattice, T when evaluating t.

Table 3.2: Basic Statistics of the Data Set

Geo Boundary	(21.89,24.84),(32.16,37.70)
Time Covered	from Oct,2009 to Nov,2013
Num.Tweets	1,436,186
Num.Words	183,478

3.5 Experimental Results

In order to show the value of our approach in analyzing real-world data, we ran our model on a Twitter data set collected within the geographical boundary of Egypt from October 2009 to November 2013. We are particularly interested in this data set because social movements were frequent in Egypt at this time[6] and Twitter has been considered by many to have played at least some role in both planning and promoting of these demonstrations and gatherings [30, 55]. We examine two aspects of the model in our experiment. First, we provide a qualitative interpretation of several events uncovered from a trained model to illustrate our ability to discover major events that can match reports from newspaper and online sources. Second, we provide a quantitative analysis of the prediction accuracies of location and time in a held out testing data set. In all cases, experiments are run with 400 Gibbs sampling steps, by fixing $L = 100$ and $T = 100$ and varying the number of events E unless otherwise noted. We set hyperparameters to be the following values: $\alpha = 0.05$, $\beta = 0.05$, $\gamma = 1.0$.

3.5.1 Data Set

We pre-processed the data so that only tweets written in Arabic remained, having observed that nearly all tweets utilizing the English character set were use a non-standard language that is phonetically similar to Arabic but was largely uninterpretable. For example, while with the help of a native speaker we were able to discern that "tab3an" means "of course", large portions of these tweets were not interpretable. We filter out all tweets that are composed of less than 95% of Arabic characters[2]. After these preprocessing steps, we are left with roughly 1.4 million tweeets over with a vocabulary size of approximately 180K words. The geo-boundary we use is defined by the latitude/longitude point (21.89, 24.84) in the lower right corner and the point (32.16, 37.70) in the upper right corner. This covers the entirety of the area of Egypt. Table 3.2 is a summary of basic statistics in our data set.

3.5.2 Qualitative Analysis of Events

We believed that looking for real life interpretations of the events we have detected was an intuitive first step for model validation. To do so, we selected five events from the output of our trained model that spanned different geographical regions and time periods. The events discovered by the algorithm are summarized in Table 3.3. Please note that all event geo-centers

[2]This percentage excludes English punctuations and Twitter mentions which usually fall into the English character sets. For more details on the data as part of a larger set, we refer the reader to [25]

Table 3.3: Spatial and temporal parameters of each event

E	Geo Center	G SD	Start Time	End Time
E1	30.86,29.87	0.43	2011-01-30	2011-03-21
E2	31.23,30.93	0.24	2013-09-10	2013-09-26
E3	31.77,30.84	0.32	2012-01-29	2012-03-22
E4	29.98,31.05	0.37	2012-10-15	2012-11-22
E5	31.20,29.57	0.37	2013-09-09	2013-10-13

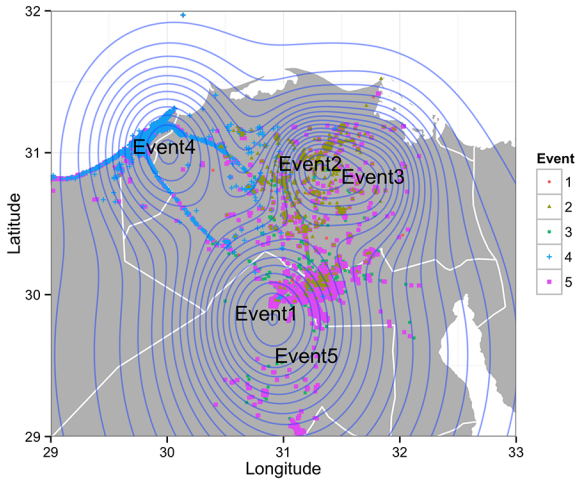


Figure 3.2: Geographical visualizations of the events and tweets belong to these events

are in the format of (Lat,lon) pair and the start date and end date are determined by $\theta_e^{(T)} - \sigma_e^{(T)}$ and $\theta_e^{(T)} + \sigma_e^{(T)}$. The spatial distribution of the five events is illustrated in Figure 3.2, where each point represents a tweet and a particular event being ascribed to by the color and shape. The figure displays up to 20,000 randomly sampled tweets that the model associated with these five events. Figure 3.2 also overlays a contour graph for all points in the graph. The contour plot is constructed using a mixture Gaussian distribution. To construct such a mixture Gaussian distribution, we use γ to serve as the mixture weight and use the event geographical centers and deviations for each Gaussian component. The result is a single distribution on a two-dimensional space that represents latitude and longitude. Curved circles in the contour plot represent the probability density of the distribution. Regions with multiple such curves are the ones that have steep change in their mixture Gaussian distributions. The contour plot shows three clear geographical clusters that correspond to three large cities in Egypt: Alexandria (left), Cairo (bottom right) and El-Mahalla El-Kubra (top right). As is also clear, certain events are located within the same cities. Without the temporal and lexical dimensions of the model, it would thus be difficult to discern differences between these events. However, exploring these distributions makes it relatively easy to observe the very different focus of each of these sets of tweets.

Figure 3.3 displays the temporal distributions of the five events of interest. Though we have analyzed each event independently in validating the model, we focus here on the most relevant event, labeled Event 1(E1). This event's tweets were heavily centered in Cairo and took place

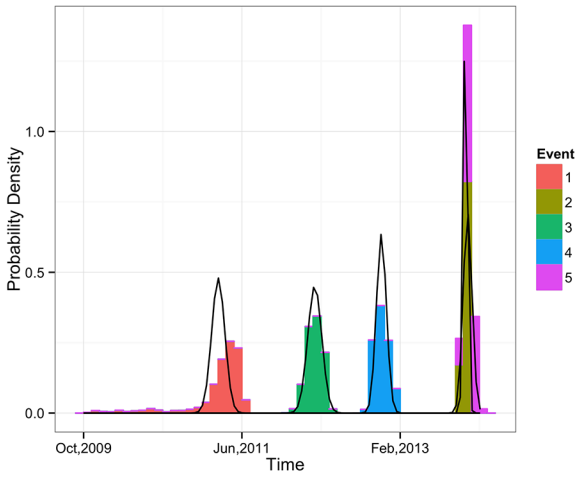


Figure 3.3: Temporal visualizations of the events

during the earlier portion of 2011. Without considering the topical focus of the event, these clues suggest that it corresponds to the initial protests that spurred the rapid spread of the social movement generally referred to as the Arab Spring [6]. The protests were held largely in Tahrir Square, located within Cairo. Additionally, the central date associated with the protests was January 25 and start from January 28 the government started to force the protestors to leave. Nevertheless, the main protest lasted for approximately three weeks with continuous demonstrations continued after that. The model's inferred start date for Event 1 was January 30th, extending to an end date of March 21st.

The topic for Event 1 in the event category in Table 3.4 supports the idea that Event 1 uncovers the protests in Tahrir Square. Here we see words such "burn", "arrested", "honor", "injustice", "tortured", all of which match what we would expect to have seen and have expected to be protested during the demonstrations. Indeed, the focal date of the protests occurred on January 25 and we correspondingly observe that the popular term "jan25" appear frequently in our data set. The most representative words in Event 1's topic also include the name "ghonim", referring to the activist Wael Ghonim who played a central role in the protests.

While we focus here on Event 1, we note that the other events in our dataset do appear to have a qualitative realization in the real world. For example, Event 3 describes a (comparatively) minor event related to an outbreak of hand and foot disease in Egypt around February of 2012 [3].

3.5.3 Quantitative Analysis

While our qualitative analysis shows the real-world relevance of model output, it does not provide an illustration of how well the model fits the data, nor how it performs in a predictive setting. In this section, we compare three variants of the model and use each for three different prediction tasks given varying amounts of information about the test data. We train each model on a training data set composed of a randomly selected set of 90% of the data, leaving 10% of the data for

[3] http://www.fao.org/news/story/en/item/129919/icode/

Table 3.4: Top words for each event

E1	jan25	arrested	Egypt	Ghonim
	burn	injustice	Libya	tortured
E2	guilt	minimum	death	hurts
	Arif	home	pulse	lord of
E3	scar	pharmacist	disease	immediately
	eye	urticaria	evil	transplantation
E4	live	promise	tireless	condensed
	need	granulate	thanks	traipse
E5	end	voice	winter	lord, thou
	god	I want	lord	to god

testing. We explain the models used, the prediction tasks and the level of information we use from the test data in turn below.

Model variants

The first model variant we consider is the full model proposed in Figure 3.1, marked as **M=L+T**. Second, we use a model with only the location component, ignoring information on time and thus ignoring $\bar{t}$, and $\Phi^{(T)}$. We denote this as **M=L**. Finally, we use a model that does not utilize location information, eliminating the location variables l, $\bar{l}$ and $Phi^{(L)}$. This is denoted as **M=T**.

Prediction tasks

In the first task, we use each model and the information given to us in the test data to predict the words in each tweet. We evaluate this by using perplexity. Second, we use each model to predict the time of each tweet in the test data. Finally, we use each model to predict the location of each tweet in the test data.

Utilization of test data

For all of the three prediction tasks, we vary the level of information we use from the test data in order to make the specified prediction. When analyzing perplexity, we vary whether or not we provide the model with time information, location information, neither or both. Giving the full model temporal or location information should naturally improve its ability to predict the words used in the tweet. Note that when we give the model neither time nor location, the full model reduces to an LDA-like one. For predicting location, we vary whether or not the full model is given time, while for predicting time we vary whether or not the full model is given location. In both cases, all models are given the words in each document in the test data.

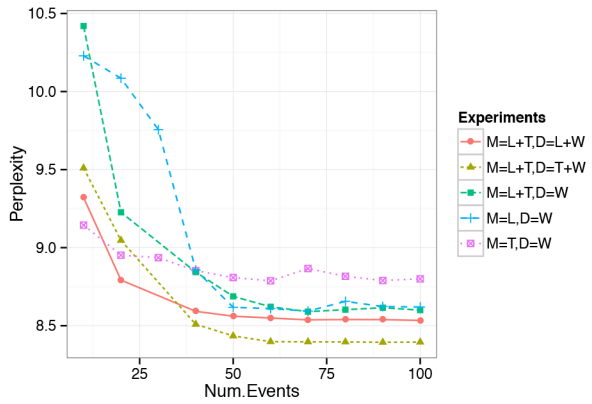


Figure 3.4: Perplexity over the number of events

3.5.4 Perplexity analysis

We define the log perplexity of a document D_{test} in Equation 3.23. The value is equal to the negative sum of the log probability of all words appearing in our test data set. The higher the probability of each word in the model, the lower the perplexity.

$$log(PPX(D_{test})) = -\frac{1}{N_W}\sum_{d\in D}\sum_{w\in W_{d,*}} log(p(W_{d,w})) \tag{3.23}$$

Experimental results for perplexity are illustrated in Figure 3.4, where each colored line represents a different model/test data combination. For example, the line marked with "M=L+T,D=L+W" represents the results with Model M=L+T trained on a data set where both location and text information are given for training while "M=L+T,D=W" represents the same model where only text is given during training. On the x-axis we vary the number of events the model is trained with. Two important observations can be made about the plot. First, the figure shows that up to a point, model performance improves with an increasing number of events regardless of the model and test data used. When the number of events becomes large enough (e.g. 50) the decrease in perplexity is not as substantial as before, suggesting that the number of events is large enough to capture the major event information in our data set. Second, and more importantly, Figure 3.4 shows that the full model performs significantly better than all other models when given temporal and text information about the test data and when trained with a large enough number of events.

3.5.5 Prediction of location and time

The prediction of location and time shows similar pattern to perplexity, indicating that with certain number of events approaches, the full model performs better than the alternative models. And the more data we provide in training, the better prediction results we will achieve. This is illustrated in Figure 3.5 and Figure 3.6. Results thus indicate that the model is able to make good use of the provided information and improves on models that do not take into account location or time.

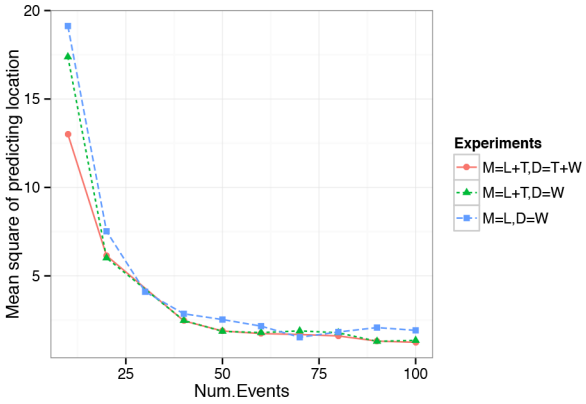


Figure 3.5: Mean square error (MSE) of predicting location over the number of events

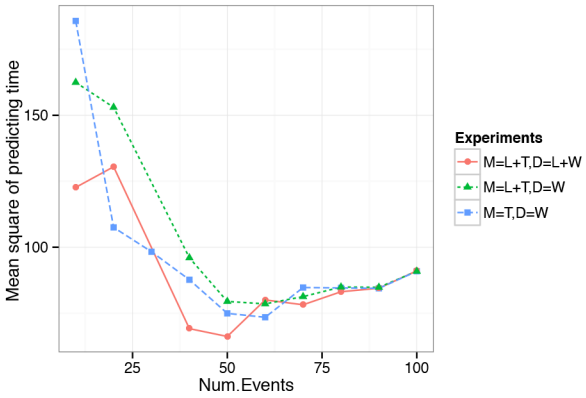


Figure 3.6: Mean square error (MSE) of predicting time over the number of events

3.6 Discussion

In this chapter we proposed a probabilistic graphical model to discover latent events that are clustered in the spatial, temporal and lexical dimensions. Both the qualitative analysis and quantitative analysis we present justified our model on a large Twitter data set. Results show that our model improved over baseline approaches on a variety of prediction tasks. These qualitative efforts show that our work can be used in a variety of application areas where event extraction and location/time prediction of social media data is of interest, like in the detection of protests and demonstrations as shown here but also in detecting, for example, important local sporting events that may be relevant to different users.

One important component of the model is the Gaussian assumptions on the distributions of both the geo-spatial coordinates and the time stamps of the events. These assumptions ensure the existence of event location and time centers which are represented by the density mass in the Gaussian distribution. They also enable the model to discover events ranging from geo-spatially/temporally constrained to those that are more universal. The assumptions of using Gaussian to model location and time are also validated in prior work such as Hong et al. (2012)

and Sakaki et al. (2010). Still, it may be interesting to explore other options for the structure of the geospatial and temporal distribution of events in the future.

There are several ways in which the present work can be further extended. First, both location and time are converted into an index through an evenly distributed selection function. There may be better approaches in cases where geo-temporal distributions are uneven, as is frequently the case in real-world data. Second, a control on granularity of the event should be added so that when tweaking the granularity of the variables, one can generate (or discover) events that are more localized or globalized. Finally, the assumption that a spatial and temporal related topic is allocated on an evenly spaced grid requires further investigation. One immediate solution is to use techniques such as k-d tree to generate topics on regions of different sizes.

Chapter 4

Modeling Temporal Evolutionary Events

4.1 Introduction

Clustering techniques have become increasingly important to the community of machine learning with the increasing amount of unlabeled data sets that can be easily acquired. Thanks to the growing amount of social media and social networking applications, publicly available text data has grown at a massive, exponential rate. As the amount of data produced has rapidly surpassed human capacity for interpretation, one of the most important questions we face today is how we can effectively organize this data together to form clusters in the data that are meaningful for human.

Techniques such as Latent Dirichlet Allocation (LDA) [18], or otherwise known as topic models, have been one of the most popular clustering methods to deal with this task for text data. In topic models, latent representations of clusters referred to as "topics" are learned by scanning a large text corpus. When meta information such as spatial coordinates or timestamps are present, extensions of topic models can be developed. Examples include the geographical topic model[44], the dynamic topic model [15] and the event detection model [83]. In those models, clusters usually contain distributions that describes meta data in addition to topic distributions found in traditional topic models.

In many situations, clusters may change or evolve over time. For example, the topic about presidential elections on 2012 and 2016 might focus on different aspects but share certain similarities. A temporal evolutionary model such as the dynamic topic model [15] can identify the subtle changes of such evolutions and adapt the topic clusters dynamically other than identifying them as a completely new cluster. To allow the number clusters to be automatically determined by the data set rather than setting a fixed value, the non-parametric version of the evolutionary dynamic models is proposed by utilizing Recurrent Chinese Restaurant Process (RCRP)[1]. Using RCRP, we are able to construct evolutionary models with infinite number of clusters. With the help of inference techniques such as Sequential Monte Carlo, massively paralleled online algorithms can be developed to deal with streaming data sets.

One of the issues for evolutionary dynamic models is the problem of non-conjugacy between the data likelihood and the evolutionary prior. In such models, cluster evolutionary priors are usually chosen to be logistic-normal distributions [1, 15], which is not conjugate with the

Multinomial likelihood used in topic modeling. Non-conjugacy put significant computational limitations to the evaluation of marginal likelihood, which is usually required for the inference of such statistical models. The usual solution to this issue is to utilize Laplace Approximations to approximate the marginal likelihoods. In this approximation, Taylor expansion up to the second order is used to approximate the integral around a point that maximizes that original function. The particular form of the evolutionary dynamic model makes it difficult to solve this maximum point. Based on Bayesian theory, prior work choose the point to maximize the data likelihood along instead of the posterior in order to get a solution that is much easier to solve [1]. This solution, however, ignored information on the prior, which contains historical clusters on previous time steps.

Another issue with the evolutionary dynamic models for clustering is the difficulties involved in inference in general. Prior work[1] uses RTS smoothing[39, 47] to solve the model, which is only feasible when the emission functions are in the form of strictly Gaussian. In situations where emission functions can not be expressed as a single Gaussian, new inference technique has to be developed.

In this paper, we study inference techniques to solve the evolutionary dynamic clustering problem. To illustrate how our technique work, we apply it onto the *Evolutionary Social Event Discovery (ESED)* problem Based on prior work on event detection[83]. The ESED task is to discover evolutionary latent clusters of documents that characterize distinct social events by monitoring an evolving set of documents with spatiotemporal meta-data that contain text about social events. Our experimental results suggest that we are able to detect major evolutionary social events on a set of Twitter data. Although the methods are illustrated through a model to solve a specific problem, we note that our inference technique can be used to solve latent evolutionary clustering models in general that are not restricted to the ESED problem.

4.2 Background

4.2.1 Topic Modeling

Topic modeling has become a popular approach to discover latent topics in large collections of text data[18, 41]. Over the past decade, many work have been done to extend topic modeling by incorporating meta information[58, 83] by improving its sampling efficiencies[3, 51, 91], and by improving the generalizability of the model[14, 15]. Within the topic modeling literature, perhaps the most relevant work for our purposes are those models dealing with temporal dynamics. For example, the dynamic topic model [15] uses a parametric model to characterize changes of topics over time by assuming logistic-normally distributed topics. The RCRP model [1] takes care of temporal dynamics in a non-parametric fashion but doesn't include spatial as a dimension in its model as we do here.

4.2.2 Non-parametric Bayesian

There exist a wide range of Bayesian non-parametric techniques that are relevant to topic modeling, most of which are based on the idea of Dirichlet Process (DP) [36]. Since topic modeling

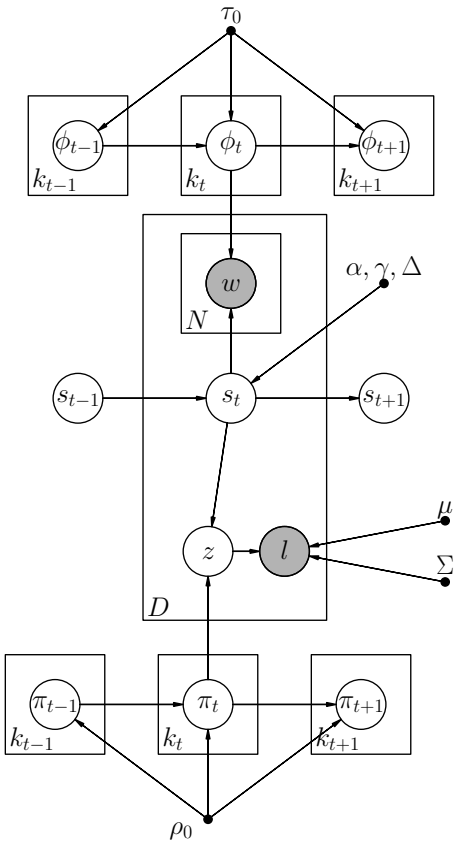


Figure 4.1: Graphical Model

usually assumes a hierarchical structure on its Dirichlet distributions, the DP cannot be directly applied unless simplifications to the models are made (e.g. making documents to have only one topic). Instead, hierarchical models, such as the Hierarchical Dirichlet Process (HDP)[78], nested Chinese Restaurant Process (nCRP) [19] and nested Chinese Restaurant Franchise Process(nCRFP) [4] have been proposed to develop non-parametric version of topic models.

Another strand of research addresses the temporal dynamics of non-parametric clustering or topic modeling specifically. For example, the recurrent Chinese Restaurant Process (RCRP) [1] divides data into epochs and the process of choosing a specific cluster membership for the d^{th} document at time (epoch) t, $s_{t,d}$, is given by Equation 4.1. Here $s_{1:(t,d)-1}$ denotes the set of all documents before (and excluding) the d^{th} document at time t. Documents can either create a new cluster with probability proportional to the dispersion parameter γ or reuse the existing cluster k with probability proportional to $\sum_{\delta=0}^{\Delta} e^{-\frac{\delta}{\alpha}} m_{t-\delta,k}^{-d}$. Here $m_{t-\delta,k}^{-d}$ is the number of documents belong to cluster k at time $t-\delta$ that includes all the documents before (and excluding) d. $e^{-\frac{\delta}{\alpha}}$ here is a decay factor that put more weights on recent time steps rather than historical ones. By using Gaussian transiting distributions, we are able to develop evolutionary document clustering algorithms such as the one in [1].

$$P(s_{t,d} = k|s_{1:(t,d)-1}) \propto \begin{cases} \sum_{\delta=0}^{\Delta} e^{-\frac{\delta}{\alpha}} m_{t-\delta,k}^{-d} & \text{k exists} \\ \gamma, & \text{k is new} \end{cases} \tag{4.1}$$

4.2.3 Non-conjugacy on Logistic-Normal Prior with Multinomial Likelihood

Temporal dynamic models with a topic modeling component [1, 15] rely on the logistic normal distribution to provide the ability to model topic evolutions. A logistic normally distributed variable $L(X)$ can be acquired by applying a logistic function $L(\cdot)$ onto the normally distributed variable X. Unfortunately, the non-conjugacy between logistic normal prior and multinomial likelihood makes it difficult to integrate the topic variable out, which is essential for efficient and effective inference in practice. Many solutions have been proposed to address this issue, such auxiliary sampling using the Polya Gamma distribution [29] and Laplace approximation [1]. In this paper, we favor the later approach since the auxiliary sampling method still needs to sample each dimension of the latent variable. In the Bayesian setting of Laplace approximation, our goal is to come up with an approximation to the marginal likelihood, denoted as M. The basic idea of Laplace approximation is to use a single point $\widehat{\theta}$ to approximate the whole integral mass. Here we let $h(\theta) = -\frac{1}{N}(\log P(X|\theta) + \log \pi(\theta))$ with N being the number samples, d being the dimension of the data, and $\Sigma = (D^2 h(\widehat{\theta}))^{-1}$.

$$M = \int P(X|\theta)\pi(\theta) \approx P(X|\widehat{\theta})\pi(\widehat{\theta})(2\pi)^{d/2}|\Sigma|^{1/2}N^{-d/2} \tag{4.2}$$

A Laplace approximation solution that is similar to the problem we are studying in this paper has been proposed in [1]. However, their solution ignored the historical data and, for reasons described below, makes too many simplifying assumptions. We will remedy this issue here by providing a better solution to the approximation that is efficient at the same time.

4.2.4 Sequential Monte Carlo

Sequential Monte Carlo (SMC) methods, otherwise known as particle filtering [33] methods, are widely used in the inference of Bayesian models [2, 23, 34]. SMC algorithm keeps track of several sets of instances, known as "particles" and update them sequentially. For each instance, an SMC algorithm maintains the posterior distribution of latent variables given the data. In our case, since documents are organized into epochs, SMC maintains the posterior $P(z_{1,(t,d)}, s_{1:(t,d)}|x_{1:(t,d)})$. Here $z_{1,(t,d)}$ is the set of latent variables up to the d^{th} document at time t. Similar notations apply to $s_{1:(t,d)}$ and $x_{1:(t,d)}$, which are cluster indicators and the data, respectively.

An SMC algorithm updates this posterior to $P(z_{1,(t,d+1)}, s_{1:(t,d+1)}|x_{1:(t,d+1)})$ after scanning another piece of data $x_{(t,d+1)}$ by sampling a proposal distribution in the form of $Q(z_{(t,d+1)}, s_{(t,d+1)}|x_{1:(t,d+1)}, z_{1,(t,d)}, s_{1:(t,d)})$. Here, like in all SMC algorithms, we maintain several sets of those particles and calculate "particle weights" to evaluate how good of a representation of the true posterior distribution they are. Once the weights in the particles become

Table 4.1: The notation used in the construction of our statistical model

Symbol	Description
(t, d)	index of document d^{th} document at time t
$1 : (t, d)$	a collection of documents up to the d^{th} document at time t
K_t	num. of events at time t
D_t	num. of documents at time t
$N_{t,d}$	num. of words belongs to document (t, d)
M	num. of Gaussian distributed location centers
F	num. of particles in Sequential Monte Carlo
$s_{t,d}$	event index of document (t, d)
$\pi_{t,k}$	mixture weight (before logistic transform) of location centers of event k at time t
$\phi_{t,k}$	topic distribution (before logistic transform) of event k at time t
μ_m	mean parameter for location m
Σ_m	co-variance matrix of component m
$l_{t,d}$	location of document (t, d)
$w_{t,d}$	text that belongs to document (t, d)
α	decay factor for RCRP
γ	dispersion parameter for RCRP
Δ	temporal width for RCRP
τ_0	parameter for topic transition Gaussian co-variance matrix
ρ_0	parameter for location weight Gaussian co-variance matrix
$L(\cdot)$	logistic function
τ_k	the first time step when cluster k presents.

unbalanced, we eliminate low particles and duplicate high weight ones. This process is referred to as resampling in the SMC literature.

There are several benefits of using SMC in the inference of Bayesian models. First, SMC framework makes it easy to develop online algorithms that deal with streaming data, which is a very important property for document clustering. Second, SMC algorithms can be naturally parallelized and the computational load of the algorithms can therefore be evenly distributed on each particle.

4.3 Statistical Model

Rooted in prior work on event discovery [83], our model characterizes a social event as a collection of distributions on text and location that change with time. Figure 4.1 displays a probabilistic graphical model representation of our model and Table 4.1 provides an overview of notation used. Our model can roughly be characterized as follows: we assume that a cluster at a particular time step is characterized by a spatial distribution and a topical distribution over words. Importantly, these distributions are allowed to evolve over time. Within a given time step, each document is characterized by the cluster it belongs to. The cluster to which it belongs

informs the set of words the document is likely to have, as well as the location the document is likely to be sent from. On the latter point, each document is characterized by a location represented by a latitude, longitude pair. In our model, this latitude and longitude is generated by selecting a specific pre-defined *region*, described below.

A key component of the model we develop is that we discretize the time stamps of tweets (referred to generically here as documents) and organize them into $epochs$. For example, if we chose to discretize our data into month-long time periods, all documents with a timestamp in January, 2016 would fall into the same epoch, while February 2016 will be another epoch, etc. The d^{th} document at time step (or synonymously, epoch) t is labeled with the subscript (t, d). More specifically, each document has a unique *event index* $s_{t,d}$ generated from a RCRP with dispersion parameter γ, temporal width Δ and decay factor α [1]. Here $m_{t,k} = \sum_{i=1}^{D_t} \mathbb{I}(s_{t,i} = k)$ represents the number of documents that belong to cluster k at time t and $m_{t,k}^{-d}$ represents this same quantity up to document d. Compared to Dirichlet Process [36], the RCRP considers the temporal dynamics of clusters in the history. Specifically, the hyper-parameter Δ controls the amount of history information to be taken into account. From Equation 4.1, we can see that recent data will receive much higher weight and that weight decays exponentially over time. The parameter α controls the speed of such decay. As a result of RCRP, new events can be "born" and old events can "die out" once the weight becomes zero, as the event will therefore not be able to attract subsequent documents.

Within each cluster k in our model, there exists a topical component $\phi_{t,l}$ and a spatial component $\pi_{t,l}$ for each time t that are initially generated by a Gaussian centered on 0 with diagonal covariance $\tau_0 I$ and $\rho_0 I$ respectively.

$$\phi_{t,k} \sim \mathcal{N}(0, \tau_0 I) \tag{4.3}$$

$$\pi_{t,k} \sim \mathcal{N}(0, \rho_0 I) \tag{4.4}$$

For a given document, the probability of generating the words in the document, $w_{(t,d),i}$ and the region index of the document $z_{t,d}$ are determined using a multinomial distribution. By applying a logistic function $L(*)$, the parameters $\phi_{t,l}$ and $\pi_{t,l}$ serve as the natural parameter of these distributions. Hence, w and z follows a logistic normal distribution. Such structure is not new to the community of topic modeling and has been explored by many prior work such as the Correlated Topic Model [13].

$$w_{(t,d),i} \sim Multi(L(\phi_{t,s_{t,d}})) \tag{4.5}$$

$$z_{t,d} \sim Multi(L(\pi_{t,s_{t,d}})) \tag{4.6}$$

Once the region index $z_{t,d}$ of a document is determined, the actual document location $l_{t,d}$, which contains a two-dimensional vector representing latitude and longitude, can be generated by using the Gaussian prior μ and Σ generated from each region.

$$l_{(t,d)} \sim \mathcal{N}(\mu_z, \Sigma_z) \tag{4.7}$$

One unique characteristic of our model is to allow both the topical parameter $\phi_{t,k}$ and the spatial parameter $\pi_{t,k}$ to evolve with time. This can be achieved by using another Gaussian evolutionary prior on existing events for the current time step that is centered on but that can deviate from the value of the last time step. This idea has been explored in [1]. However, as we mentioned, the authors tried to approximate the emission function using a single Gaussian, which is a reasonable assumption in that model but no longer holds in our scenario since we are modeling spatial component as well.

$$\phi_{t,k} \sim \mathcal{N}(\phi_{t-1,k}, \tau_0 I) \tag{4.8}$$

$$\pi_{t,k} \sim \mathcal{N}(\pi_{t-1,k}, \rho_0 I) \tag{4.9}$$

The model can be summarized with a description of its generative process, which is as follows:

1. For each time period t:
   (a) For each existing event k
      i. Draw $\pi_{t,k} \sim \mathcal{N}(\pi_{t-1,k}, \rho_0 I)$
      ii. Draw $\phi_{t,k} \sim \mathcal{N}(\phi_{t-1,k}, \tau_0 I)$
   (b) For each document d
      i. Draw event index $s_{t,d}$ from $RCRP(\gamma, \alpha, \Delta)$
      ii. If $s_{t,d} = k$ is a new event
         A. Draw $\pi_{t,k} \sim \mathcal{N}(0, \rho_0 I)$
         B. Draw $\phi_{t,k} \sim \mathcal{N}(0, \tau_0 I)$
      iii. Draw $w_{(t,d),i}$, zt, d and $l_{t,d}$ according to Eq.4.5, Eq.4.6, and Eq.4.7

4.4 Scalable Inference

4.4.1 Integrating Variables

We start with the joint probability of the model and seek a collapsed version of it, $P(s, z, w, l|\mu, \sigma, \gamma, \Delta, \alpha, \rho_0, \tau_0)$ by integrating out the natural parameters $\phi_{t,k}$ and $\pi_{t,k}$. In the following derivations, we will omit hyper-parameters and use "$\cdot$" to annotate them for cleaner notation. We also define $g(\cdot)$ to be the likelihood function. The location likelihood $g(\pi_{t,k})$ and the text likelihood $g(\phi_{t,k})$ are defined in Equation 4.10 and Equation 4.11, respectively. Here $n^{\pi}_{t,k,g}$ and $n^{\phi}_{t,k,i}$ are the number of occurrence in cluster k at time t for location component g and vocabulary i, respectively.

$$g(\pi_{t,k}) = \prod_{d=1}^{D_t} P(z_{t,d}|\pi_{t,k}, s_{t,d} = k) = \prod_{g} \Big(\frac{e^{\pi_{t,k,g}}}{\sum_j e^{\pi_{t,k,j}}}\Big)^{n^{\pi}_{t,k,g}} \tag{4.10}$$

$$g(\phi_{t,k}) = \prod_{d=1}^{D_t} P(w_{t,d}|\phi_{t,k}, s_{t,d} = k) = \prod_i \left(\frac{e^{\phi_{t,k,i}}}{\sum_j e^{\phi_{t,k,j}}}\right)^{n^{\phi}_{t,k,i}} \tag{4.11}$$

By utilizing the notations defined above, the integration can be expressed in Equation 4.12. Here we use τ_k to denote the fist time step when cluster k occurs. We also define $\psi^{\pi}_{t,k} = 0$ when $t = \tau_k$ and $\psi^{\pi}_{t,k} = \pi_{t-1,k}$ if $t > \tau_k$. Similar definition can be applied to $\psi^{\phi}_{t,k}$.

$$\begin{aligned} &P(s, z, w, l|\cdot) \\ =&\prod_{k=1}^{K}\prod_{t=\tau_k}^{T}\int_{\pi_{t,k}} g(\pi_{t,k})P(\pi_{t,k}|\psi^{\pi}_{t,k})\int_{\phi_{t,k}} g(\phi_{t,k})P(\phi_{t,k}|\psi^{\phi}_{t,k}) \\ &\prod_{t=1}^{T}\prod_{t=\tau_k}^{T}\prod_{d=1}^{D_t} P(s_{t,d}|s_{1:(t,d)-1})P(l_{t,d}|\mu_k, \Sigma_k, z_{t,d} = k) \end{aligned} \tag{4.12}$$

The key to this integrations is to correctly deal with terms involving the likelihood function $g(\cdot)$ and its priors, which is $\prod_{t=\tau_k}^{T}\int_{\pi_{t,k}} g(\pi_{t,k})P(\pi_{t,k}|\psi^{\pi}_{t,k})$. As we will see shortly, we can conduct integrations in a chain fashion from the very beginning when $t = \tau_k$ all the way to the end when $t = T$. We will get a constant term and a future term each time when an integration is done at a specific time step. The future term, which we annotate as $f_{t,k}(\pi_{t+1,k}|\theta_{t,k})$ contains the information for a future integration and will participate the integration in the next time step. The constant term, which we annotate as $D_{t,k}$ will be emitted as part of our final integration result. Here we will focus on the terms that involves π and we will omit the procedures for ϕ since it can be derived similarly.

As we mentioned above, the future term $f_{t,k}(\pi_{t+1,k}|\theta_{t,k})$ is generated as part of the integration result at time t. It contains variable $\pi_{t+1,k}$ that will participate the integration of the next time step $t + 1$ with parameter $\theta_{t,k}$ that is determined by information on the previous time steps. To illustrate how the future term $f_{t,k}(\pi_{t+1,k}|\theta_{t,k})$ interplay with the integration, we define the following relationship in Equation 4.13. We also assume that $f_{t-1,k}(\pi_{t,k}|\theta_{t-1,k})$ is in the form of Gaussian distribution with mean $\theta_{t-1,k}$ and covariance matrix $\rho_0 I$. We will prove this using mathematical induction.

$$\int_{\pi_{t,k}} g(\pi_{t,k})P(\pi_{t+1,k}|\pi_{t,k}, \rho_0 I)f_{t-1,k}(\pi_{t,k}|\theta_{t-1,k}) \tag{4.13}$$

Apparently, for the base cases, where $t = \tau_k - 1$ we define $f_{\tau_k-1,k}(\pi_{t=\tau_k,k}|\theta_{\tau_k-1,k})$ to be a zero mean Gaussian with covariance matrix $\rho_0 I$. One can validate this definition by taking $f_{\tau_k-1,k}(\pi_{t=\tau_k,k}|\theta_{\tau_k-1,k})$ into Equation 4.13 to get the expression for the first integration.

$$f_{\tau_k-1,k}(\pi_{t=\tau_k,k}|\theta_{\tau_k-1,k}) = \mathcal{N}(\pi_{t,k}|0, \rho_0 I) \tag{4.14}$$

For general case where $\tau_k \leq t < T$, we define the recursive formula of $f_{t,k}(\cdot)$ in Equation 4.16 to be the integration of $\pi_{t,k}$ divided by a constant $D_{t,k}$, which is defined in Equation 4.15

and is designed to absorb all constants that is not related to the Gaussian distribution to participate the next round of integration.

$$D_{t,k} = \mathcal{N}(\widehat{\pi_{t,k}}|\theta_{t-1,k}, \rho_0 I)(2\pi)^{d/2}|\Sigma_{t,k}|^{1/2}N_{t,k}^{-d/2}g(\widehat{\pi_{t,k}}) \tag{4.15}$$

Here we utilize the induction assumption that $f_{t-1,k}$ is a Gaussian distribution with mean $\theta_{t-1,k}$ and covariance matrix $\rho_0 I$. We also use Laplace Approximation to approximate the integral around a point $\widehat{\pi_{t,k}}$, which will be discussed in more detail in the next sub-section. After letting $D_{t,k}$ to absorb all the constants, we again get a Gaussian form of $f_{t,k}(\cdot)$ with mean value equal to $\widehat{\pi_{t,k}}$ and covariance matrix $\rho_0 I$.

$$\begin{aligned}
&f_{t,k}(\pi_{t,k}|\theta_{t-1,k}) \\
&= \frac{\int_{\pi_{t,k}} P(\pi_{t+1,k}|\pi_{t,k}, \rho_0 I) \cdot f_{t-1,k}(\pi_{t,k}|\theta_{t-1,k})g(\pi_{t,k})}{D_{t,k}} \\
&= \frac{\int_{\pi_{t,k}} \mathcal{N}(\pi_{t,k}|\frac{\pi_{t+1,k}+\theta}{2}, \rho_0 I/2)\mathcal{N}(\pi_{t+1,k}|\theta_{t-1,k}, 2\rho_0 I)g(\pi_{t,k})}{D_{t,k}} \\
&= \mathcal{N}(\widehat{\pi_{t,k}}|\frac{\pi_{t+1,k}+\theta_{t-1,k}}{2}, \rho_0 I/2)\mathcal{N}(\pi_{t+1,k}|\theta_{t-1,k}, 2\rho_0 I) \\
&\quad \frac{g(\widehat{\pi_{t,k}})(2\pi)^{d/2}|\Sigma|^{1/2}N^{-d/2}}{D_{t,k}} \\
&= \mathcal{N}(\pi_{t+1,k}|\widehat{\pi_{t,k}}, \rho_0 I)\mathcal{N}(\widehat{\pi_{t,k}}|\theta_{t-1,k}, \rho_0 I) \\
&\quad \frac{(2\pi)^{d/2}|\Sigma_{t,k}|^{1/2}N_{t,k}^{-d/2}g(\widehat{\pi_{t,k}})}{D_{t,k}} \\
&= \mathcal{N}(\pi_{t+1,k}|\widehat{\pi_{t,k}}, \rho_0 I)
\end{aligned} \tag{4.16}$$

Please note that we do not define $f_{t,k}(\cdot)$ when $t = T$ since there will be no term to contribute to future integrations. To summarize, the integration of $\pi_{t,k}$ over time equals to the $\prod_{t=\tau_k}^{T} D_{t,k}$. We can use the same technique to get the integration of $\phi_{t,k}$ to be $\prod_{\tau_k}^{T} C_{t,k}$ with $C_{t,k}$ defined below:

$$C_{t,k} = \mathcal{N}(\widehat{\phi_{t,k}}|\theta_{t,k}, \rho_0 I)(2\pi)^{d/2}|\Sigma|^{1/2}N^{-d/2}g(\widehat{\phi_{t,k}}) \tag{4.17}$$

We then use the same notation to get the joint distribution after π and ϕ are integrated out by taking the results in the previous steps into Equation 4.12:

$$\begin{aligned}
P(s, z, w, l|\cdot) = &\prod_{t=1}^{T}\prod_{k=1}^{K_t} C_{t,k}D_{t,k} \\
&\prod_{t=1}^{T}\prod_{d=1}^{D_t} P(s_{t,d}|s_{1:(t,d)-1})P(l_{t,d}|\mu, \Sigma, z_{t,d})
\end{aligned} \tag{4.18}$$

The the collapsed joint distribution leave only two variables to be inferred: $z_{t,d}$ and $s_{t,d}$ for each document (t, d). In experiments we found that the MCMC converges very quickly and only several Gibbs iteration steps are necessary for the algorithm to reach convergence.

4.4.2 Laplace Approximation to Marginal Likelihood

Although we have discussed the general form of the joint distribution after the integration, we haven't covered the details on how we conducted the Laplace Approximation when taking the integral. As seen in Equation 4.2, Laplace's method approximates the integral around a specific point where the majority of the probability mass lies on. In our case, $h(\cdot)$ takes the form of a negative log of a Multinomial likelihood function with a Gaussian prior. And ideally we should choose $\widehat{\pi_{t,k}}$ to minimize $h(\pi_{t,k})$. When we use sequential techniques to solve the model, we do not have the knowledge of cluster parameters in the next time step, $\pi_{t+1,k}$, which is required to evaluate $h(\pi_{t,k})$. Instead, we use the expectation of its prior information $\pi_{t-1,k}$ to approximate $h(\pi_{t,k})$. $h(\pi_{t,k})$ then becomes:

$$\begin{aligned} h(\pi_{t,k}) =& \frac{-\log\big(\mathcal{N}(\pi_{t,k}|\frac{\theta_{t-1,k}+\pi_{t+1,k}}{2}, \frac{\rho_0^2}{2}I)g(\pi_{t,k})\big)}{N_{t,k}} \\ =& \frac{-\log\big(\mathcal{N}(\pi_{t,k}|\theta_{t-1,k}, \frac{\rho_0^2}{2}I)g(\pi_{t,k})\big)}{N_{t,k}} \end{aligned} \tag{4.19}$$

When the sample size $N_{t,k}$ is large enough, the impact of the prior will be very small and a natural selection of $\widehat{\pi_{t,k}}$ will be the one that maximize its likelihood. This solution is illustrated in Equation 4.20 as **Solution 1** and is used by [1]. Here, we illustrate the solution by its logistic form rather than its original form, which is more useful since $g(\pi_{t,k})$ utilizes the logistic form of $\pi_{t,k}$. This solution simply normalizes the number of documents having the locations in spatial component i for each cluster k at time t, $N^{\pi}_{t,k,i}$ with the total number of documents that belong to cluster k at time t, $N^{\pi}_{t,k}$. However, this solution ignores all the historical data before time t since it ignored the prior information.

$$Solution1 : \frac{e^{\widehat{\pi_{t,k,i}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}} = \frac{N^{\pi}_{t,k,i}}{N^{\pi}_{t,k}} \tag{4.20}$$

Another solution that can be used as an natural comparison to solution 1 is to use the document count of all historical data that cluster k has on this location component i instead of the count just on this particular time step. Equation 4.21 illustrates the exact form of **Solution 2**. Here the solution is taken to be the normalized count of all the documents belong to cluster k that are located in location component i, $N^{\pi}_{k,i}$. This solution, however, ignores the temporal importance and information across all time steps are treated equally.

$$Solution2 : \frac{e^{\widehat{\pi_{t,k,i}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}} = \frac{N^{\pi}_{k,i}}{N^{\pi}_{k}} \tag{4.21}$$

However, we note that neither of the solutions above take into account of the prior information. A better approach is to solve $\widehat{\pi_{t,k}}$ to minimize the whole $h(\pi_{t,k})$ rather than only the likelihood part. In order to do this, we take the derivative of Equation 4.19 and set it to zero. After we assume that $\sum_j e^{\pi_{t,k,j}} = 1$, we got the relations in Equation 4.22.

$$\frac{2\theta_{t,k,i} + n_{t,k,i}\rho_0^2}{n_{t,k}} - \frac{2}{\rho_0^2 n_{t,k}}\pi_{t,k,i} = e^{\pi_{t,k,i}} \tag{4.22}$$

The above equation fall into the set of problems that can be solve using the notation of *Lambert's W* [31]. This solution can be expressed analytically and we illustrate it in Equation 4.23 in the logistic form. In this equation, we define $n^{'}_{t,k,i}$ such that $e^{\theta_{t,k,i}} = \frac{n^{'}_{t,k,i}}{n_{t,k}}$ to represent the pseudo counting that is introduced by the prior, which is an important trick that will be utilized later.

$$\begin{aligned}\frac{e^{\widehat{\pi_{t,k,i}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}} &= \frac{2}{\rho_0^2 n_{t,k}} W(\frac{e^{\theta_i}\rho_0^2 n_{t,k}}{2} e^{\frac{\rho_0^2 n_{t,k,i}}{2}}) \\ &= \frac{2}{\rho_0^2 n_{t,k}} W(\frac{\rho_0^2 n^{'}_{t,k,i}}{2} e^{\frac{\rho_0^2 n_{t,k,i}}{2}})\end{aligned} \tag{4.23}$$

We observed that terms inside Lambert W's function can be bounded by two quantities.

$$\begin{aligned}&\min\{\frac{\rho_0^2 n_{t,k,i}}{2} e^{\frac{\rho_0^2 n_{t,k,i}}{2}}, \frac{\rho_0^2 n^{'}_{t,k,i}}{2} e^{\frac{\rho_0^2 n^{'}_{t,k,i}}{2}}\} \\ &\leq \frac{e^{\widehat{\pi_{t,k,i}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}} \leq \\ &\max\{\frac{\rho_0^2 n_{t,k,i}}{2} e^{\frac{\rho_0^2 n_{t,k,i}}{2}}, \frac{\rho_0^2 n^{'}_{t,k,i}}{2} e^{\frac{\rho_0^2 n^{'}_{t,k,i}}{2}}\}\end{aligned} \tag{4.24}$$

By utilizing the fact that $W(xe^x) = x$, we know that the actual solution of $\frac{e^{\widehat{\pi_{t,k,i}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}}$ must lie in the linear combination of its lower and upper bounds. A good choice of the linear weight is the use the information on the variance, ρ_0. Since ρ_0 controls the amount of information that we can allow to change from one time step to the other, a natural choice of the combination weight would be $1/(1+\rho_0)$ and $\rho_0/(1+\rho_0)$. We put weight $1/(1+\rho_0)$ on the bound that contains the information about the prior while using $\rho_0/(1+\rho_0)$ on the bound that contains the information on the current time step. Our solution, which here will be referred to as **Solution 3**, takes the linear combination of the document count $n_{t,k,i}$ at time t of cluster k on component i and the pseudo document count on the prior $n^{'}_{t,k,i}$ normalized by the total number of documents that belong to cluster k on this time step t, $n_{t,k}$. And since $n^{'}_{t,k,i}$ is normalized by $n_{t,k}$, our solution takes value range from 0 to 1.

$$\begin{aligned}&Solution3 : \frac{e^{\widehat{\pi_{t,k,i}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}} \\ &= \frac{2}{\rho_0^2 n_{t,k}}\big(\frac{1}{1+\rho_0} W(\frac{\rho_0^2 n_{t,k,i}}{2} e^{\frac{n_{t,k,i}\rho_0^2}{2}}) + \frac{\rho_0}{1+\rho_0} W(\frac{\rho_0^2 n^{'}_{t,k,i}}{2} e^{\frac{n^{'}_{t,k,i}\rho_0^2}{2}})\big) \\ &= \frac{2}{\rho_0^2 n_{t,k}}(\frac{1}{1+\rho_0}\frac{\rho_0^2 n_{t,k,i}}{2} + \frac{\rho_0}{1+\rho_0}\frac{\rho_0^2 n^{'}_{t,k,i}}{2}) \\ &= \frac{\frac{1}{1+\rho_0} n^{'}_{t,k,i} + \frac{\rho_0}{1+\rho_0} n_{t,k,i}}{n_{t,k}}\end{aligned} \tag{4.25}$$

We can derive similar solution for the optimal points to be used in Laplace approximation for $\phi_{t,k}$. For the length of this paper we will omit the exact derivation since the results are highly similar.

4.4.3 Sample Cluster Index $s_{t,d}$

Starting from Equation 4.18, it is now straightforward to derive Equation 4.26 to sample the cluster index $s_{t,d}$ for each document. Here we see that the equation is linear to the number of words in the document, $N_{t,d}$. We need to choose one of the three solutions proposed in the previous section to substitute $\widehat{\phi_{t,k,n}}$ and $\widehat{\pi_{t,k,z}}$. $P(s_{t,d}|s_{1:(t,d)-1})$ is the RCRP prior defined in Equation 4.1.

$$\begin{aligned} &P(s_{t,d} = k|s_{1:(t,d)-1}, w_{t,d}, z_{t,d}) \\ &\propto \prod_{n=1}^{N_{t,d}} \left(\frac{e^{\widehat{\phi_{t,k,w_{t,d,n}}}}}{\sum_j e^{\widehat{\phi_{t,k,j}}}}\right)^{N_{d,t,i}} \left(\frac{e^{\widehat{\pi_{t,k,z_{t,d}}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}}\right) P(s_{t,d}|s_{1:(t,d)-1}) \end{aligned} \tag{4.26}$$

4.4.4 Sample Region Index $z_{t,d}$

Similarly, we can derive the equation to sample the location region index $z_{t,d}$ for each document from the joint distribution in Equation 4.18. Here we see that the probability of selecting $z_{t,d}$ is proportional to the logistic normal component $\frac{e^{\widehat{\pi_{t,k,z}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}}$ and $P(l_{t,d}|\mu_z, \Sigma_z)$, which is the Gaussian probability of location $sl_{t,d}$ on the $z_t h$ component of the Gaussian prior.

$$P(z_{t,d} = z|s_{t,d} = k, l_{t,d}) \propto \left(\frac{e^{\widehat{\pi_{t,k,z}}}}{\sum_j e^{\widehat{\pi_{t,k,j}}}}\right) P(l_{t,d}|\mu_z, \Sigma_z) \tag{4.27}$$

4.4.5 SMC Updates

As we stated in the background section, SMC evaluates a weight for each particle and we need to update this weight every time after we have sampled a new document. Our result is illustrated in Equation 4.28. Here we see that the weight update is proportional to the likelihood of the newly sampled data.

$$\begin{aligned} \omega^f_{1:(t,d)} \propto &\omega^f_{1:(t,d)-1} \prod_{n=1}^{N_{t,d}} P(w_{t,d,n} = v|s_{t,d} = k, \phi_{t,k,v}) \\ &P(l_{t,d}|z_{t,d} = z, \mu_z, \Sigma_z) \end{aligned} \tag{4.28}$$

4.4.6 Algorithm

The general procedure of our approach is illustrated in Algorithm 1. We organize our data into epochs and for each epoch t we process documents one by one. For each document (t, d), only two variables z and s are sampled and we iterate through the MCMC step $MaxIter$ times. Particle weights are then updated and we evaluate whether it is necessary to resample particles by comparing the L2 norm of the particle weight to a threshold.

Algorithm 1 Particle Filtering Algorithm Framework

Initialize ω_1^f to $\frac{1}{F}$ for all $f \in \{1, ..F\}$
for epoch t from 1 to T **do**
    for document d from 1 to D_t **do**
        for particle $f \in \{1, ..F\}$ **do**
            for iter $\in MaxIter$ **do**
                Sample s,z using Eq. 4.26 and 4.27
            end for
            Update ω^f using Eq. 4.28
        end for
        Normalize particle weight ω^f
        if $||\omega_t||_2^{-2} <$ threshold **then**
            resample particles
        end if
    end for
end for

Table 4.2: Summary of dataset used in the experiment

Item	Statistics
Spatial Coverage	United States
Temporal Coverage	Aug, 2010 to Sep, 2012
Vocabulary Size	40,173
Num. Documents	5,298,978

4.5 Data

We collected Twitter data from August, 2010 to September, 2012 using Twitter's Decahose API. Only tweets with geo-coordinates within United States are kept; everything else is discarded. We conducted basic natural language processing on the data and deleted stop words and punctuations. Words in the documents are converted to lower cases and are tokenized. Low frequency words that appear below a threshold are deleted. At the end, we are left with a dictionary size of 40,173 unique tokens and a document size of 5,298,978. We keep 90% of them for training and 10% of them for testing.

4.6 Experimental Results

4.6.1 Qualitative Results

We evaluate our model qualitatively by examining the cluster contents discovered in the experiment. Figure 4.2 illustrated the contents of the superbowl cluster. Those two results are taken from the same cluster on February 2011 and February 2012 respectively. We also listed a fact sheet that about the superbowl evetns on those two years, which can be found in Table 4.3. Here

Table 4.3: Fact sheet about the Superbowl Event

	Superbowl XLV	Superbowl XLVI
Time	February 6, 2011	February 5, 2012
Location	Arlington, TX	Indianapolis, IN
Teams	Pittsburgh Steelers Green Bay Packers	New York Giants New England Patriots

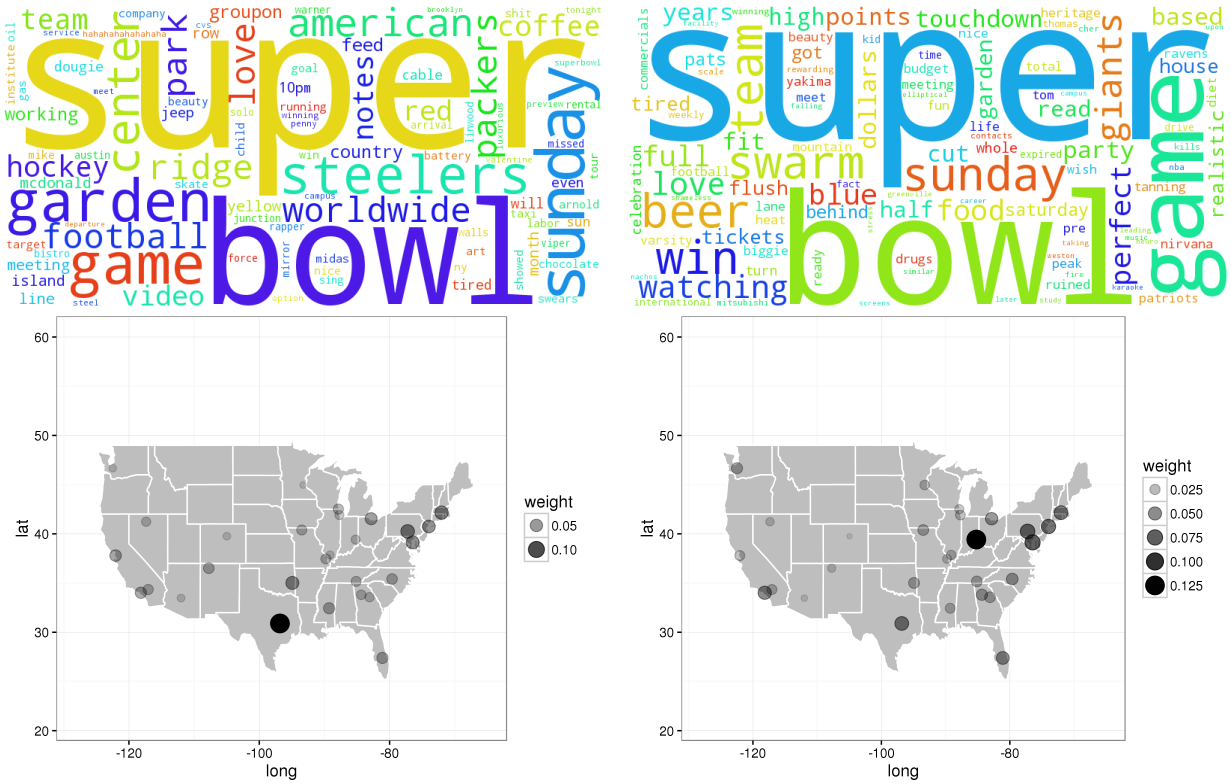
we see that those two events share certain common topics with leading words such as "super" and "bowl" dominating the the topical distributions. However, they also exhibited their characteristics. In the word cloud visualization in the first time step, we see keywords such as "stellers", "packers" which are the names of the two competing teams in the superbowl event. The spatial distribution of the event clearly highlight the state of Taxes, which is the actual location center where the superbowl event is held at that year. Since this is a nationwide event, minor centers to represent the point of interests exist across the country. On the right hand side of the results, we see the visualizations of the superbowl 2012 event. Here we see different topical patterns illustrated by word cloud, which contains keywords such as "giants", "patriots" which represent the names of the competing teams in the superbowl. The spatial visualization clearly illustrated the spatial center of the event, which lies in the state of Indiana. Again, since it's a major nationwide event, location of interests exist across the country. New york city and Boston areas, which are the hometowns of the participating teams, constitute to a significant portion of the location distribution.

4.6.2 Numerical Results

We conducted numerical results by first training our model using 90% of the data and then testing it on the rest of the data set by measuring its testing perplexity and the Mean Square Error(MSE) of prediction on held out document locations. We compared three different solutions for Laplace approximation in Eq. 4.20, Eq. 4.21 and Eq. 4.25. Although our theoretical results favor **Solution 3**, comparison is still important since prior work reported using **Solution 1** and **Solution 2** in similar models.

In the perplexity result in Figure 4.3a, we tested for results using various parameterizations of the model and the three different solutions for the Laplace approximation. Here perplexity is calculated according to Equation 4.29. The first thing we see here is that **Solution 2** generally performs better than the other two baseline approaches. However, it is clear that this result is impacted by the setting of model parameters.

For example, perplexity on testing data changes with the parameter α, which is the decay factor of the RCRP prior defined in Equation 4.1- the perplexity when α is large is generally better than those with smaller α. Our proposed approach beats the other two baseline methods when α is larger than 0.1. This indicates that our model prefers a less radical weight distribution on previous epochs. Instead, taking more epochs into considerations generates a much better result.



(a) Superbowl event on February 2011

(b) Superbowl event on February 2012

Figure 4.2: Superbowl Event Detected by Our Algorithm

We observe a similar pattern for variable γ, which is a dispersion parameter that controls the way that new clusters are created in the model. We see that a high γ yields a better performance for our proposed approach, which is not surprising since generally the higher model complexity the better expressiveness the model will be able to generalize our data.

The pattern of perplexity for variable τ_0 is also very interesting. Recall that τ_0 is the Markov transiting prior that controls the variance of the same topic from one time to the other. Since both the baseline solutions ignore this prior information, one of their performance will change with τ_0. Our approach, however, does change with τ_0 and we can clearly observe a region where our approach outperform the others. When τ_0 is close to infinity, our proposed approach approximates **Solution 1**. When τ_0 is close to zero, no counting on the current time step (i.e. $n_{t,k,i}$) is being used and nothing is being learned in the model. The Markov transiting prior for location, ρ_0 has less impacts to the perplexity results than the prediction error for location, which we will discuss in the next paragraph.

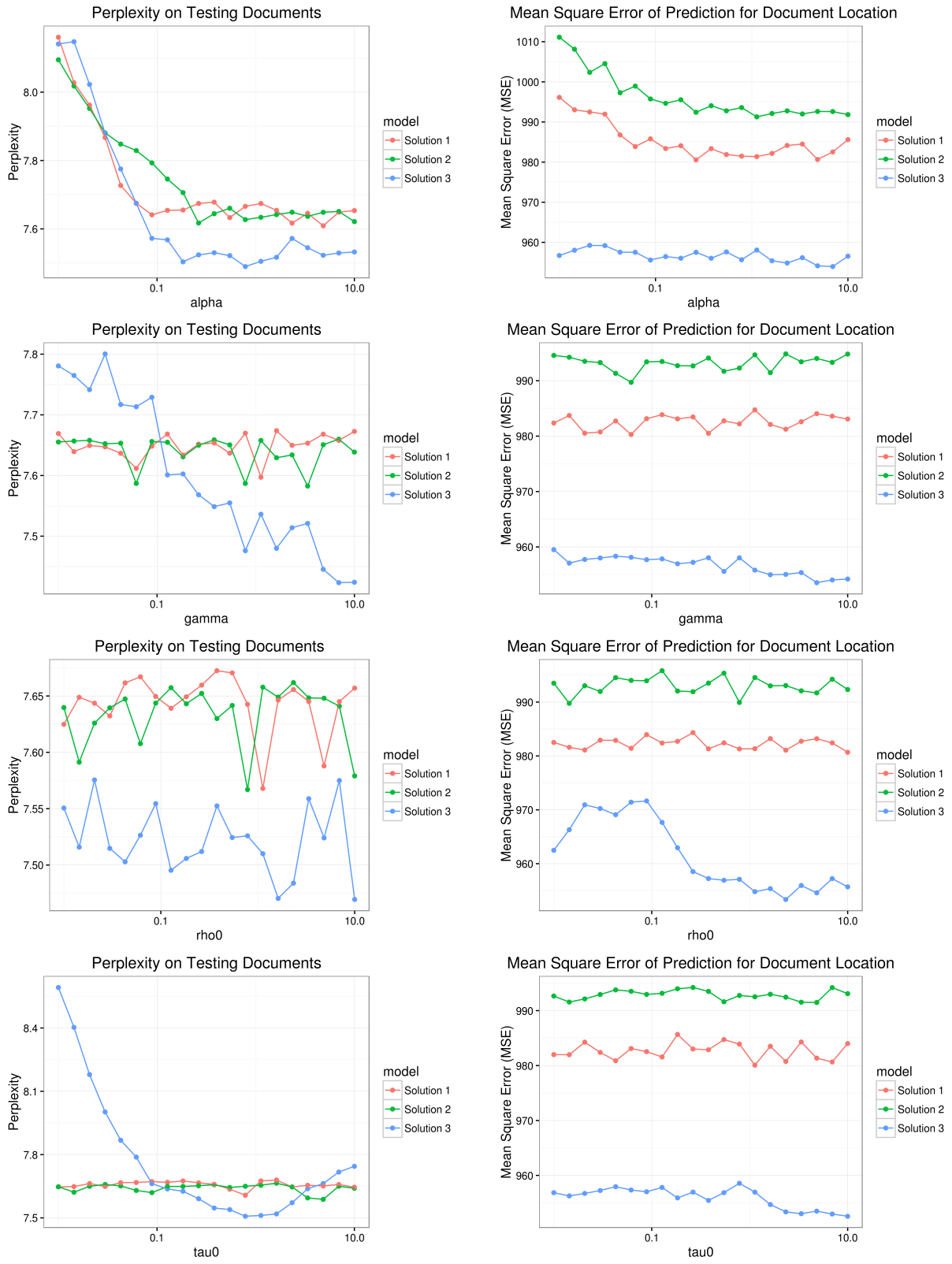
$$perp(\mathcal{D}_{test}) = -\frac{\Sigma_t \Sigma_d \Sigma_n \log p(w_{t,d,n}|s_{t,d} = k, \phi_{t,k})}{\Sigma_t \Sigma_d \Sigma_n 1} \tag{4.29}$$

Figure 4.3b shows the Mean Square Error (MSE) for the task of prediction the location specified of the left out posts. The results are illustrated in Figure 4.3b. Here we see that our proposed methods outperform the baselines significantly throughout the range of the tested parameters. Most parameters do not impact location prediction results. The only exception is ρ_0, which is the Markov transitioning prior for location. Here we see that the MSE climbed slightly with an increase of ρ_0 but declined sharply after ρ_0 reaches 0.1. This indicates that higher prior values that put more weight on recent counting information are beneficial for the model to effectively learn to predict locations.

4.7 Discussion

In this paper, we proposed a Bayesian non-parametric model to discover evolutionary social events. We experimented with the model on Twitter data and are able to identified evolutionary latent social events. Equally as importantly, the algorithm we develop is highly efficient, parallelizable and can be applied to a variety of problems that are related to document clustering with evolutionary clusters. By comparing this approach with inference algorithm to similar problems in other literature, we found that our approach significantly outperform other baseline methods in terms of perplexity on testing data as well as document location predictions.

There are several limitations to this paper that are open to potential future work. First, the assumption that each document has to belong to a specific event is a bit simplistic considering many tweets are not event-centric. Second, the fact that the spatial priors are predetermined makes the model difficult to deal with streaming data outside the predetermined spatial regions. And finally, spatial components do not penalize towards their distances to the event centers and this assumption can sometime generate universal events that doesn't contain any specific event information. Future word should address these issues.



(a) Perplexity on testing data set when model parameters are changed

(b) Mean Square Error (MSE) of document location when model parameters are changed

Figure 4.3: Perplexity and prediction for document location when model parameters are changed

Chapter 5

Modeling Events from Multiple Data Sources

5.1 Introduction

The web has become a primary source of information for many individuals, both because of the large quantity of information it provides and because of the effortless and timeliness with which this information is provided to us. More specifically, online newspaper agencies and social media websites provide an important (if somewhat biased [66]) way to access breaking news and major events. With so much information being generated each day, it is increasingly clear that the problem with individuals receiving information online today is not that the information is unavailable but rather that we are overloaded [43] with all kinds of information on things we may care little about. It is therefore difficult to find a way to explore the major events that have happened in a timely and concisely manner.

One way to solve this problem is to utilize topic modeling techniques such as Latent Dirichlet Allocation [18] to organize documents into topics. From here, those accessing information on the web could quickly skim over content related to a variety of themes. However, people are generally interested in what is happening *here and now*, and topics are just one of the dimension of ongoing events. Thus, vanilla topic models overlook and many other important elements of newsworthy content. For example, although they might share similar collections of topics, a news article talking about a social revolution in Egypt is likely to be interesting to a different group of people than an article discussing revolution in Syria. Thus, incorporating the location of focus for a document, in addition to its content, is very important. Another example is the time of the events; for example, we would not want to group together the presidential debate in 2008 with the one in 2016.

One problem that is often overlooked is that multiple media cover the same events, a fact that is missing from some of the recent work on social event discovery which considers only a single media [27, 38, 69, 70, 83]. However, combining multiple media, particularly those coming from social as opposed to traditional news media, can be valuable. On the one hand, more traditional sources of news, such as online newspapers, usually contain more formal language than social media such as Twitter and Facebook. This seems to make topics more interpretable

in such media. On the other hand, information circulated in social media usually tracks what is happening "on the ground" more quickly than traditional news media. By utilizing data from multiple data sources, one might therefore be able to compliment the pros and cons of each and enable the construction of better and more timely information on events. Moreover, while we have qualitative expectations of the formalness and timeliness of different media, such a model would help us to better quantify and measure these characteristics of different media.

In this chapter, we propose a Bayesian non-parametric model to discover latent events by leveraging data from multiple media sources. The non-parametric [36] natural of our model allows us to infer an infinite number of such events without having to restrict the number before running the model. We utilize Sequential Monte Carlo methods [33] and develop a parallel online algorithm that works on streaming data. The following research questions guide our efforts:

1. Can we efficiently generalize latent events in a timely fashion and link the relevant documents across multiple media?
2. Can we generate latent representations of these events that have strong interpretability?
3. Can we learn the nature of a specific online media in terms of reporting social event?

We will answer these questions by first introducing our statistical model and then providing experiments using the model on a large scale data that contains both Twitter and newspaper data.

5.2 Background

5.2.1 Topic Modeling

Forms of topic modeling have long existed in the information extraction literature as Latent Semantic Analysis [35]. However, it was not until the development of Latent Dirichlet Allocation (LDA) [18] and its subsequent Collapsed Gibbs sampling solution [41] that make the methodology to become popular. By assuming a Multinomial distributed topic for each word in the document and its Dirichlet distributed word topic distribution, LDA can be used to effectively discover latent topics in large corpora of text. Since the original model, many work have been proposed to either improve its sampling complexity [51, 91] to model words that are correlated [16], to include temporal dynamics on topics [15] or to extend its application beyond just text data [44, 83].

5.2.2 Non-parametric Bayesian Modeling

One of the problems in parametric topic models is that the number of topics needs to be determined from the beginning and can not be changed throughout the experiments. This limits the applications of topic modeling to streaming data, in which the number of the topics can change in the future and thus cannot be determined beforehand. To solve this issue, non-parametric versions of topic models have been developed by utilizing Dirichlet Process [36]. In Dirichlet Process-based topic models, the document membership index s can be generated in Equation 5.1, which is proportional to the number of documents that belongs to an existing cluster k, n_k or the

dispersion parameter γ if the model "decides" to create a new cluster for this document.

$$P(s_d = k|s_{1:d-1}, \gamma) \propto \begin{cases} n_k, & k \text{ is an existing cluster} \\ \gamma, & k \text{ is a new cluster} \end{cases} \tag{5.1}$$

Extensions of the Dirichlet Process, such as the Hierarchical Dirichlet Process [78], Recurrent Chinese Restaurant Process [1] and Dirichlet Hawkes Process [34], have been developed to enable models to determine the number of topics and hence the complexity of the model by itself based on the data set. One benefit of non-parametric models is that they can dynamically generate new topics based on data streams if online inference algorithms are developed for them [2]. Another benefit of using non-parametric models is the ability to model recursive topics. Methods such as the nested Chinese Restaurant Process[19] and the Nested Chinese Restaurant Franchise Process [4] are proposed to provide infinite hierarchical structures amount topics. In this chapter, we build our model based on the standard Dirichlet Process to ensure that online inference is scalable. However, more sophisticated models such as the hierarchical ones can also be applied, which opens the possibility for future work on, e.g., hierarchical clustering of events.

5.2.3 Sequential Monte Carlo

One of the challenges in using non-parametric models in real-world applications is the computational complexity raised during inference. Sequential Monte Carlo (SMC) methods, otherwise known as particle filtering [24, 33], are widely used to solve non-parametric models [2, 34] and even some parametric ones [23]. SMC-based algorithms function by tracking multiple "particles", or samples, over time and sequentially updating them. SMC algorithms scan documents one by one; at any particular document d, with data $\mathcal{X}$, membership index s, and other hidden variables $\mathcal{Z}$, each particle keeps track of the posterior distribution up to the current document d, $P(\mathcal{Z}_{1:d}, s_{1:d}|\mathcal{X}_{1:d})$. When a new document $d+1$ arrives, each particle updates its estimation of the posterior to $P(\mathcal{Z}_{1:d+1}, s_{1:d+1}|\mathcal{X}_{1:d+1})$, where we use the notation $s_{1:d}$ to denote the set of variable from document 1 to document d.

At this point, each particle then "uses" a proposal distribution $Q(\mathcal{Z}_d, s_d|\mathcal{Z}_{1:d-1}, s_{1:d-1}, \mathcal{X}_{1:d})$ to sample the latent variables of the current document. Finally, the algorithm updates the particle weight w_f for each particle f. These weights determine the extent to which each particle is representative of the true underlying posterior. Over time, some particles might gain weight that are much higher than others and begin to dominate the posterior distribution. A common way to address this fact that we leverage here is to simply resample the particles to keep only the important instances alive.

There are many benefits of using SMC. First we can develop online algorithms since SMC scans one document at a time. Second, the sampling of each particle can be done independently and they only needs to be synchronized when resampling happens, which enables us to develop parallel versions of the sampling algorithm.

5.2.4 Event Detection

Topic modeling has been leveraged in the literature on event detection. Most of these methods, however, focus either solely on extracting events from the text of documents [70, 86] or from

Table 5.1: Notations

Symbol	Description
M	size of the vocabulary
G	num. of spatial location centers
K	num. of event clusters in the current system
B_d	num. of location coordinates for document d.
D	num. of documents
G	Dirichlet Process
s_d	event index of document d
μ_g,Σ_g	location priors for Gaussian center g
$l_{d,b}$	b^{th} location of document d
$z_{d,b}$	region index for b^{th} location of document d
$w_{d,i}$	i^{th} word of document d
$q_{d,i}$	topic category of i^{th} word in document d. 0 - event topic, 1 - background topic
c_d	media indicator of the document. 0 - social media, 1 - newspaper
δ_d	topic category distribution of document d.
ϕ_k	topical distribution for event k.
ϕ_c	background word distribution for media c.
π_k	location center distribution for event k.
θ_k	universal temporal center for event k
δ_k^2	universal temporal variance for event k.
$\theta_{k,c}$	media specific temporal center for event k and media c.
γ	dispersion parameter for Dirichlet Process
α, β	Inverse Gamma priors for δ_k^2
λ	parameter to generate media specific temporal center $\theta_{k,c}$.
η_c,η_k,η_π	Dirichlet priors for ϕ_c,ϕ_k and π_k.
C_g^k	number of locations that belong to center k and event k. i.e. $\Sigma_d \Sigma_b I(s_d = k, z_{d,b} = g)$
D_q^d	number of words that belong to topical category q in document d. i.e. $\Sigma_n I(q_{d,n} = q)$

meta-data involved in the documents [73]. While approaches have been developed that leverage both meta-data and content [27, 76, 83], these methods have largely focused on a single media and have tended to be largely parametric in nature. Our work thus extends existing efforts on event detection in that it combines a non-parametric approach, the use of meta-data and document content and considers multiple media within a single model.

5.3 Statistical Model

Our statistical model assumes a particular way in which events and documents are being created. The graphical model is presented in Figure 5.1 and Table 5.1 provides a list of the notation used in this chapter. In a nutshell, events are created in a way that is independent of documents and

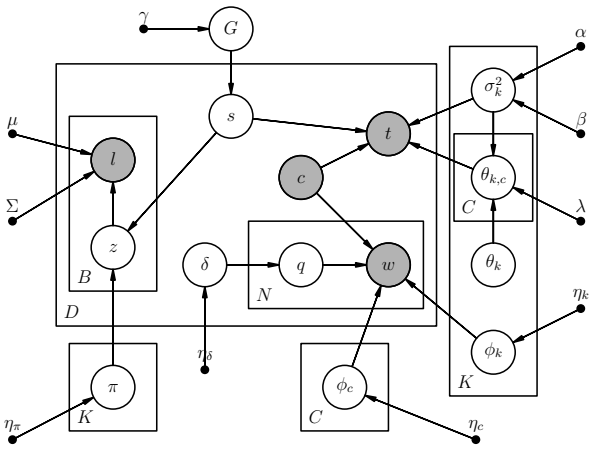


Figure 5.1: Graphical Model

each is represented by global parameters that are irrelevant to media and local parameters that are specific to media types. Documents for each media are generated by first choosing an unique event index using Dirichlet Process defined in Equation 5.1 followed by its observed information such as location l, text w and time t by drawing from its corresponding event distributions.

5.3.1 Event Component

For each event cluster k there is a location region distribution π_k and a topical distribution ϕ_k, both of which are Dirichlet distributed with hyper-parameter η_π and η_k respectively. To divide geo-spaces into regions, we first train a Gaussian mixture model with G clusters on a large set of geo-spatial coordinates in our data set and provide the spatial centers μ and spatial co-variance matrix Σ to serve as priors of the Gaussian distributed spatial coordinates. The location regions distribution π_k can therefore be interpreted as a distribution of an event cluster over these G dimensions. The topical distribution ϕ_k, on the other hand, represents the frequency of choosing one of the M vocabularies in the documents that belong to this event cluster.

Events in our model also have a temporal distribution with parameters σ_k^2 and θ_k. Those two parameters are global to different media and events don't directly generate document time through these two parameters. σ_k^2 is generated using an Inverse-Gamma distribution with hyper-parameters α and β while we let θ_k be unrestricted.

Since we use a non-parametric treatment in our model, event clusters are created dynamically. We will illustrate the mechanism behind this in Section 5.3.3.

5.3.2 Media Component

Although most of the parameters in the event cluster are global, there are certain local variants of parameters that are different for each media. First of all, each media has a media-specific "perception" of the temporal centers of the event temporal distributions, $\theta_{k,c}$. These centers are generated using a Gaussian distribution centered on the event's "true" temporal center θ_k with an event variance σ_k^2/λ. The factor λ here ensures that different media can have their own

time at which they focus on a particular event that is centered around the true temporal center. The parameter λ controls how strongly we allow media specific centers $\theta_{k,c}$ to deviate from the universal temporal center of events, θ_k. The larger the λ, the less variation we allow for media specific centers, and vice versa. This parameterization, as we will show, is useful in the situation that we want to study which media "broke the news" of a given event first. Finally, each media has a media specific background word ϕ_k which is Dirichlet distributed with hyper-parameter η_c. In the document model which we will introduce shortly, words in the document can be chosen from either the event specific distribution ϕ_k or the background distribution ϕ_c baed on the value of topical indicator q. The background topic can be used to trap meaningless words that have very little value to distinguish social events, which will help the model to focus on the event specific words.

For simplicity, the present work focuses on two types of media: Twitter and Newspaper. However, we note that without loss of generality, one can easily extend the model to apply them into multiple media sources.

5.3.3 Document Component

As we mentioned in the beginning of the section, each document in the model is associated with an event cluster membership index s, which is generated using Dirichlet Process in Equation 5.1. When necessary, a new event cluster will be generated along with all its parameters.

After the event index is determined, data in the documents are generated according to the parameters from the corresponding event cluster. First, each document needs to determine each its media type, c. In this chapter, c can be either Twitter or newspaper. Second, the document time t is generated using a Gaussian distribution centered on the media specific temporal center $\theta_{k,c}$ and event global variance σ_k^2. Third, a Dirichlet distributed variable δ is generated for each document to determine the distribution of event related words and background words. For each word in the document, we first generate the word category variable q, which indicates whether this words fall into event specific ones or the background topic. The actual text w is then generated using a Multinomial distribution based on both the value of q and the corresponding event topical distribution ϕ_k and background word distribution π_c.

$$w_{d,i} \sim Multi(\phi_k)^{I(q_{d,i}=0)} \cdot Multi(\phi_c)^{I(q_{d,i}=1)} \tag{5.2}$$

Finally, for each location in the document we generate the region index z, from the cluster region distribution π_k using a Multinomial distribution and the Gaussian distributed location l from the corresponding Gaussian distributions parametrized by μ and Σ. Here, we note that for many media types such as Twitter, there is a unique location attached to it. However, for our newspaper data, we have acquired many location labels and thus multiple locations are present for each document.

5.3.4 Generative Model

The model can be summarized with a description of its generative process, which is as follows:

1. for each media c draw $\phi_c \sim Dir(\eta_c)$

2. For each document d
   (a) Draw event index $s_d \sim DP(\gamma, s^{1:-d})$
   (b) Draw $\delta_d \sim Dir(\eta_\delta)$
   (c) if $s_d = k$ is a new event, draw ϕ_k, π_k, θ_k, $\theta_{k,c}$, δ_k^2
   (d) for each location b
      i. draw $z_{d,b} \sim Multi(\pi_{z_d})$
      ii. draw $l_{d,b} \sim \mathcal{N}(\mu_{z_{d,b}}, \Sigma_{z_{d,b}})$
   (e) draw $\delta_d \sim Dir(\eta_\delta)$
   (f) for each word i
      i. draw $q_{d,i} \sim Multi(\delta_d)$
      ii. draw $w_{d,i}$ according to Eq.5.2
   (g) draw $t_d \sim \mathcal{N}(\theta_{s_d,c}, \delta_{s_d})$

5.4 Scalable Inference

We start our inference procedure by integrating out most of the Dirichlet distributed variables including δ_d, ϕ_k, ϕ_c, π_k, Dirichlet Process G_0 and the media specific temporal mean $\theta_{k,c}$. Since these are conjugate priors and this procedure is widely used in the literature [41, 61], we omit the exact derivations here. After performing these integrations, we are left with three document level variables s_d, $z_{d,b}$, $q_{d,n}$ and two cluster level variables δ_k^2 and θ_k to be solved in our inference. We let $\mathcal{Z} = \{z, q, \delta_k^2, \theta_k\}$ denote the latent variables other than s and let $\mathcal{X} = \{l, w, t, c\}$ denote the set of observed variables.

We adopt a Sequential Monte Carlo framework to solve our model. In particular, we keep track of the posterior distribution up to document d, $P(s_{1:d}, \mathcal{Z}_{1:d}|\mathcal{X}_{1:d})$. We use the posterior distribution $p(s_d, \mathcal{Z}_d|s_{1:d-1}, \mathcal{Z}_{1:d-1}, \mathcal{X}_{1:d})$ to serve as the proposal distribution, $q(s_d, \mathcal{Z}_d|s_{1:d-1}, \mathcal{Z}_{1:d-1}, \mathcal{X}_{1:d})$. This is different from the usual choice of using prior as the proposal distribution as reported in [33] because posterior is usually hard to sample. However, recent research have suggested that the posterior can minimize the variance across particles [2].

For each new document d, we run MCMC over the latent variables that belong to the document and a MAP estimation on the cluster time parameters. When evaluating particle weights, we use use the samples from the last several MCMC samples. However, only the last sample is recorded to be the final value of the latent variables belong to each document. The pseudo-code for the algorithm can be found in Algorithm 2

5.4.1 MCMC Step - Sample z

To sample $z_{d,b}$, the region index for the b^{th} location of document d, we conduct a discrete sampling over the conditional probability of $z_{d,b}$, which is illustrated in Equation 5.3. Here we use $z^{-d,b}$ to denote the z variables from document 1 to document d excluding the b^{th} location and beyond and we use $C_{\tilde{g},d}^{k,-d,b}$ to indicate the number of locations up to but not including the b^{th}

location in the d^{th} document. We see here that the location region index $z_{d,b}$ can be sampled from a distribution that is proportional to the occurrence of this particular location g and the how close this particular location to its actual location $l_{d,b}$ measured by a Gaussian density function, $P(l_{d,b}|\mu, \Sigma, z_{d,b})$.

$$\begin{aligned} &P(z_{d,b} = g|s_d = k, z^{-(d,b)}, l_{d,b}) \\ &\quad \propto \frac{(C_g^{k,-d,b} + \eta_{\pi,g})}{\Sigma_g C_g^{k,-d,b} + \eta_{\pi_g}} P(l_{d,b}|\mu_g, \Sigma_g) \end{aligned} \tag{5.3}$$

5.4.2 MCMC Step - Sample q

The conditional probability of $q_{d,n}$, the topic category of the n^{th} word in the d^{th} document, is given in Equation 5.4. Here, we use similar notations as the previous section, with $n_{v,0}^{*,k-d,n}$ denoting the number of occurrences of a word v that belongs to event cluster k up to (but excluding) the n^{th} word in the d^th document. Similarly, we define $n_{v,1}^{c,*-d,n}$ to be the number of occurrences of word v that belong to media c up to (but excluding) the n^{th} word in the d^{th} document.

$$\begin{aligned} &P(q_{d,n} = q|s_d = k, c_d, q^{-d,n}, w_{d,n} = v) \\ &\quad \propto \frac{D_q^{-d,n} + \eta_\delta}{\Sigma_g D_q^{-d,n} + \eta_\delta} \begin{cases} \frac{n_{v,0}^{*,k-d,n} + \eta_{k,v}}{\sum_v n_{v,0}^{*,k-d,n} + \eta_{k,v}}, & \text{if q=0} \\ \frac{n_{v,1}^{c,*-d,n} + \eta_{c,v}}{\sum_v n_{v,1}^{c,*-d,n} + \eta_{c,v}}, & \text{if q=1} \end{cases} \end{aligned} \tag{5.4}$$

5.4.3 MCMC Step - Sample s

We sample the event indexes s after we acquire samples of z and q in the previous subsections. Equation 5.5 shows the probability of s, which involves terms $T(\cdot)$ and $H(\cdot)$ defined in Equation 5.6 and Equation 5.7 below. Notice that $H(\cdot)$ contains the likelihood on documents where $q = 0$ along with the the probability of the newly sampled z_d. On the other hand, $T(\cdot)$ basically contains information on whether the timestamp of the document fits the given cluster k, which turns out to be a Gaussian with a reweighted mean and an amplified variance. Here we use $\widehat{\theta}_k$ and $\widehat{\delta}_k^2$ to denote the MAP estimation of the event time center and event time variance. The parameter $n_{c,k}^{-d}$ is the current number of the documents that belongs to media c and cluster k which does not count document d. Similarly, $\overline{x_{c,k}}^{(-d)}$ also doesn't count the current document d.

$$\begin{aligned} &P(s_d = k|, s_{1:d-1}, q_{1:d}, z_{1:d}, \widehat{\theta}_k, \widehat{\sigma_k^2}) \\ &\quad \propto P(s_d = k|s_{1:d-1}, \gamma) T(t_d) H(z_d, q_d|s_d) \end{aligned} \tag{5.5}$$

$$H(z_d, q_d|s_d) = \prod_{w;q_{d,w}=0} \frac{n_{w,0}^{*,k-d,w} + \eta_{k,w}}{\sum_v n_{v,0}^{*,k-d,v} + \eta_{k,v}} \prod_b \frac{C_{z_{d,b}}^{k-d,b} + \eta_{\pi,z_{d,b}}}{\sum_g C_g^{k-d,b} + \eta_{\pi,g}} \tag{5.6}$$

$$T(t_d) = \mathcal{N}(\frac{\lambda\widehat{\theta}_k + n_{c,k}^{-d}\overline{x_{c,k}}^{(-d)}}{n_{c,k}^{-d} + \lambda}, \frac{(n_{c,k}^{-d} + \lambda + 1)}{n_{c,k}^{-d} + \lambda}\widehat{\delta}_k^2) \tag{5.7}$$

One thing the above equations show is that $H(\cdot)$ is linear in the number of locations and words in the document and this term needs to be evaluated K different times in order to sample s. This is very slow considering that in our model the number of clusters can increase. To overcome this issues we apply a hybrid Metropolis algorithm by setting a proposal distribution to be $q(s^*) = P(s_d = k|s_{1:d-1}, \gamma)T(t_d)H(z_d, q_d|s_d)$. When we sample form the proposal distribution, the newly sampled s^* represented the information from the Dirichlet Prior and the temporal part of the model, which is usually a strong feature to distinguish between clusters. The newly acquired sample is then accepted with probability $A = min\{1, r\}$ with r value defined in Equation 5.8. By using this technique, we can only need to evaluate $H(\cdot)$ term once each time we sample s.

$$r = \frac{H(z_d, q_d|k^*)}{H(z_d, q_d|k)} \tag{5.8}$$

5.4.4 Estimating Temporal Parameters

We use MAP to estimate the mean and variance of temporal distribution for each cluster, namely θ_k and δ_k. Here, $n_{c,k}$ does count the current document d, which is different from what we saw when sample s in the previous subsection. To simply our notation, we introduce two auxiliary variables $\omega_{0,k} = n_{0,k}(n_{1,k} + \lambda)$ and $\omega_{1,k} = n_{1,k}(n_{0,k} + \lambda)$. As it turned out in Equation 5.9, the MAP solution is a weighted average of the two mean values of document timestamps that belong to cluster k of different media. The weight $\omega_{0,k}$ and $\omega_{1,k}$ here are served as normalized weights. As is clear, the media with more documents attached to it will have a larger weight in the estimation of cluster time center.

$$\widehat{\theta_k} = \frac{\omega_{0,k}\overline{x_{0,k}} + \omega_{1,k}\overline{x_{1,k}}}{\omega_{0,k} + \omega_{1,k}} \tag{5.9}$$

The MAP solution of cluster time variance is illustrated in Equation 5.10. Here we see that variance is a weighted combination of the hyper-parameter β, the two media specific variances $\sigma^2_{c,k}$ and finally a coupling term that involves the mean of the two cluster specific centers $\overline{x_{c,k}}$. Basically, the more those media disagree with each other, the higher this estimation of the variance will be in addition to the variance from each specific media. Here $n_k = n_{0,k} + n_{1,k}$ is the total number of documents belong to cluster k. The parameters α and β here serve as smoothing parameters to initialize this estimation when one or two media does not have any data.

$$\widehat{\sigma^2_k} = \frac{2\beta + n_{0,k}\sigma^2_{0,k} + n_{1,k}\sigma^2_{1,k} + \frac{\lambda n_{0,k} n_{1,k}(\overline{x_{0,k}} - \overline{x_{1,k}})^2}{2n_{0,k}n_{1,k} + (n_{0,k} + n_{1,k})\lambda}}{n_k + 2\alpha + 2} \tag{5.10}$$

Both the mean and variance of cluster time can be estimated in a very efficient manner since they can be decomposed into the media specific cluster variance $\sigma^2_{c,k}$ the media specific cluster centers $\overline{x_{c,k}}$ and the number of documents count $n_{c,k}$. These statistics can be updated in constant time.

5.4.5 Initializations

A naive approach to initialize the latent variables is to use uniformly generated random variables to serve as the initial values of z,q and s. However, if those initial values are bad, it is likely that the algorithm would take a long time to reach equilibrium. Instead, we can sample the initial values z^* and q^* using parts of Equation 5.3 and Equation 5.4 that do not require our knowledge of cluster index s. For location index z, its initial value is sampled purely based on its location proximity to the Gaussian centers. For word category variable q, its initial values are determined by both the values of previous words in the current document d and prior η_δ.

$$P(z^*_{d,b} = g) = P(l_{d,b}|\mu_g, \Sigma_g) \tag{5.11}$$

$$P(q^*_{d,n} = q) = \frac{D_q^{-d,n} + \eta_\delta}{\Sigma_g D_q^{-d,n} + \eta_\delta} \tag{5.12}$$

5.4.6 Particle Weight

After several rounds of MCMC steps and the consequent estimation for time parameters, we update the particle weights using Equation 5.13. Here we see that the weight update is proportional to the likelihood of the data on temporal, spatial and textual dimensions. Please note that for the purpose of simplifying notations we use substitutions $s_d = k$, $c_d = c$, and omitted all the subscriptions that are related to d. We then normalize the weight and resample the particles when $||w_t||_2^{-2}$ is smaller than a threshold we set.

$$\begin{aligned} w^f_{1:d} \propto w^f_{1:d-1} T(t_d) \prod_b \frac{C^{k-d,b}_{z_b} + \eta_{\pi,z_b}}{\sum_g C^{k-d,b}_g + \eta_{\pi,g}} \\ \prod_{w;q_w=0} \frac{n^{*,k-d,w}_{w,0} + \eta_{k,w}}{\sum_v n^{*,k-d,v}_{v,0} + \eta_{k,v}} \prod_{w;q_w=1} \frac{n^{c,*-d,w}_{w,1} + \eta_{c,w}}{\sum_v n^{c,*-d,v}_{v,1} + \eta_{c,v}} \end{aligned} \tag{5.13}$$

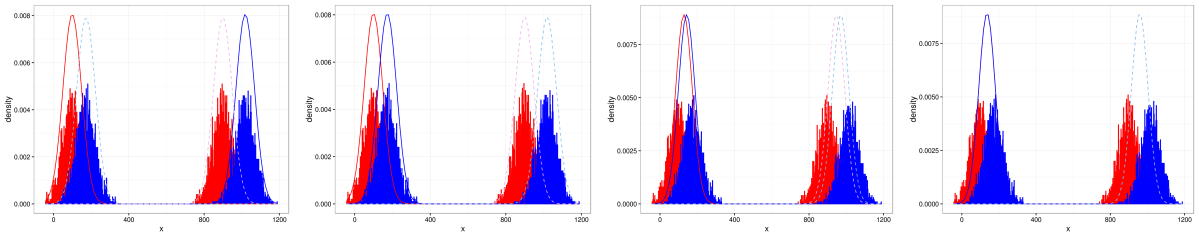
5.4.7 Recovering Cluster Parameters

We are able to recover those variables that are integrated, namely π, δ, ϕ_c, phi_k and $\theta_{k,c}$. We will omit illustrating the recovering of Dirichlet distributed variables since those procedures can be find in many LDA articles. Here we focus on the recovering the media specific temporal cluster centers $\theta_{k,c}$. We take posterior mean of the variable and the results are shown in Equation 5.14. Basically, the media specific center is an weighted average between the cluster's actual center, $\overline{x_{c,k}}$ and the joint center $\widehat{\theta_k}$ estimated in Equation 5.9. One of the implications of this equation is that the parameter λ serves as the link between those two clusters. As we will see in the experimental results on synthetic data, small λ tends to break the dependencies between the two clusters while large λ will make the media specific centers to be more alike.

$$\theta_{k,c} = \frac{\lambda\widehat{\theta_k} + n_{c,k}\overline{x_{c,k}}}{\lambda + n_{c,k}} \tag{5.14}$$

Algorithm 2 Particle Filtering Algorithm Framework

Initialize ω_1^f to $\frac{1}{F}$ for all $f \in \{1, ..F\}$
for document d from 1 to D **do**
    for particle $f \in \{1, ..F\}$ **do**
        for iteration $\in \{1..maxIter\}$ **do**
            Sample z,q and s using Eq. 5.3,5.4 and 5.5
        end for
        Estimate σ_k^2 and θ_k using Eq.5.9 and Eq.5.10
        Update particle weight ω^f using Eq.5.13
    end for
    Normalize particle weights ω^f
    if $||\omega^f||_2^{-2} <$ threshold **then**
        Resample particles
        Set particle weight to uniform, $\omega^f = 1/F$
    end if
end for



(a) $\lambda = 0$ (b) $\lambda = 1$ (c) $\lambda = 1000$ (d) $\lambda = \infty$

Figure 5.2: Experimental results on a synthetic data set with different values of γ ranging from 0 to ∞. Dark red and blue solid curves represent the media specific centers of cluster 1 while the lighter red and blue dashed curves represent those centers that belong to cluster 2

Table 5.2: Statistics of actual data being used in this chapter

	U.S.	U.K.	Japan	Arab
Num. Tweets	390,159	705,769	534,759	200,868
Num. News	41,726	3,583	1,486	8,518
Num. Total	431,885	709,935	536,245	209,386
Time	Jan,1,2011 - Dec,31,2012			

5.5 Data Sets

We use two data sets collected from Twitter's Decahose API and New York Time's article search API. Both of which covered the time frame from January 1, 2011 to December 31, 2012. For twitter data we toss out tweets that are non-geo tagged and keep those tweets with geo-coordinates fall into a particular subset of spatial areas of interest to the present work. More specifically, we consider four spatial areas of interest to the present work: the United States, the United Kingdom, Japan and a set of four countries in the Middle East/North Africa (MENA) region - Libya, Syria, Egypt and Tunisia- that we will refer to as the Arab spatial area. These spatial areas are chosen due to the high volume of data we can obtain relevant to them and the existence of interesting events during the timeframe of study that occurred within these areas.

For newspaper data, we collected all the newspaper and blogs returned from the New York Time API in the same time range as the Twitter data. The API can only return a snippet or the leading paragraph of the newspaper articles and we combined it with the title of the article to form the text of the newspaper data. Geo-spatial coordinates are acquired for these newspaper articles by reverse geo-coding the geo-region keywords annotated by the New York Times for each article using the Google Map API. Since there are less then 1000 unique geo-region keywords, we are able to examine them one by one and did minor modifications to the keyword sets to help Google Map's API. Articles without geo-region keywords are discarded. Please note that newspaper can have multiple geo-region keywords and hence it is possible to have multiple geo-coordinates being attributed to each article.

We apply standard text processing practices to the text, eliminating punctuation and stop words. One important consequence of this decision is that the # sign is removed from Twitter hashtags. We also eliminated non-ascii words (e.g. emojis) in the documents and made all the tokens lower case. Finally, we also discarded tokens that appeared infrequently in the text and eliminated documents that had no words left after these processing steps. We are left with roughly 2.1 million documents after these pre-processing steps - the breakdown of this data is illustrated in Table 5.2.

5.6 Results

5.6.1 Synthetic Data

We first look at results on a synthetic data set that only contains timestamps but no text or spatial information in order to illustrate the role of hyper-parameter λ in our model. As we discussed in

the previous section, λ determines the level of deviation of media specific temporal centers from the universal temporal centers of events. Having such a control on the deviations to the universal center helps the algorithm to detect a set of integrated newspaper and Twitter temporal centers that are close to each other as opposed to two temporal centers that are widely apart. This, in turn, allows the model to more readily connect documents across various media that refer to the same event. What is unclear, however, and thus worth experimenting, is exactly how strongly λ impacts results at varying parameterizations. In this synthetic data set, we prepared two sets of Gaussian distributions to mimic the temporal distributions of two event clusters, each with its own Twitter (red) and newspaper (blue) sub-distributions with their centers approximating each other. For the purpose of training this data, we ignored the textual and spatial components in our model and only considered the temporal parts with different values of hyper-parameter λ.

We present experimental results in Figure 5.2 by varying λ from 0 to ∞. For each sub-plot in Figure 5.2, the histograms (filled in bars) represent the underlying synthetic data given to the algorithm. Note that there are two clear events, one on the left and one on the right side of each sub plot, and that the media differences are clearly distinguishable. The density curves shown represent the media-specific estimated temporal patterns for the two event clusters picked up by the algorithm when run on the synthetic data. The algorithm picks up two event clusters with dark red and blue solid curves representing the media specific centers of cluster 1 while the lighter red and blue dashed curves represent those centers that belong to cluster 2. When λ is 0, the media-specific cluster center $\theta_{k,c}$ are generated from distributions with the same mean but infinite variance (i.e. $\delta_k^2/\lambda = \infty$). This essentially makes $\theta_{k,c}$ independent with each other as we can see from Equation 5.14. The learned events in Figure 5.2a are therefore interleaved with Twitter and newspaper clusters that belong to cluster 1 on the two sides and cluster 2 in between.

This result is not ideal, as we know reports from different media relevant to the same event should be close to each other. Having completely uncorrelated centers for different media will result in dramatically wide gap between the temporal distributions of the centers of different media. If we slightly increase λ, centers of that belong to the same event will be closer to each other but not completely overlapping, which is seen in Figure 5.2b. This is probably the most ideal situation since the restrictions put on the centers will guide algorithm to pick up events that are temporally cohesive in contents reported in two different media. When we further increase λ to 1000, the gap will be further tightened and it will be completely closed when λ reaches ∞, in which case the media specific centers are exactly the same as the universal cluster temporal center. In those cases, our algorithm lost its ability to allow flexible individual temporal centers in the events. In conclusion, we want to set the parameter λ to an appropriate value so that we allow some restrictions on the their deviations from the universal temporal center but not too much. Having such restriction is critical to the learning of our algorithm.

5.6.2 Real World Events

In this section, we demonstrate that our algorithm is able to generate latent representations of events that are naturally interpretable. In order to do so, we executed the algorithm on each of the four spatial areas individually and looked at the resulting clusters. Here, we demonstrate four events in Figure 5.3, one from each spatial area, that are significant with respect to each during that time period. We illustrate the topical distribution of each event as a world cloud and

their spatial and temporal distributions as maps and histograms with overlaying density plots, respectively. In the temporal visualizations, we distinguish between tweets (red) and newspapers (blue) and use Equation 5.14 to recover their media specific centers. As a form of validation, we collected information on the relevant event from Wikipedia and provide this information in Table 5.3.

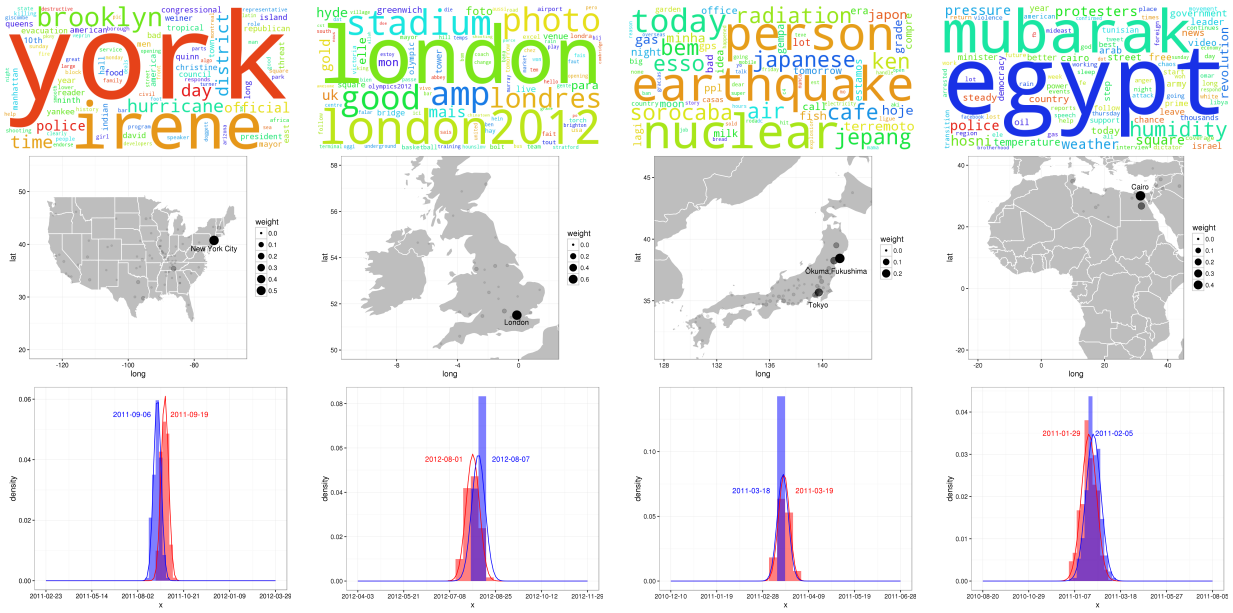
The first cluster, presented in Figure 5.3a, is centered on a latent event representative of Hurricane Irene, a large hurricane that hit much of the northeastern coast of the United States. This is clear from the world cloud, where we see highly ranked words such as ”york”, ”irene”, ”hurricane”, ”brooklyn” and ”manhattan”, which are strong indicators of the event. The spatial visualization shows that the event is centered mostly near New Jersey and New York City, where the majority of the damage was inflicted. Temporal visualizations shows the reports from newspaper came first in this event, followed by Twitter. Interestingly, news reports started to appear in mid August, when Irene was hitting the city, but the peak of the newspaper report comes early in September. Further, it is not until several days after that the tweets begin to catch up. This delay in both news and Twitter fits with our understanding of this disaster in that a lack of preparedness led to weeks and months of coverage on the effects of the hurricane, rather than the period before it. It also shows that for this particular event, traditional media led social media, which was focused more heavily on reactions once the hurricane had passed through and relief efforts were underway.

In the second cluster, in Figure 5.3b, we see a event located in London that roughly centered in August of 2012. Different from the first event, this time we see tweets lead the reports of the event and followed by newspaper. The actual Olympic games happened during July 27 to Aug 12, which nicely bounded our the majority of the mass in our histogram. The word cloud contains high frequency keywords such as ”london”, ”london2012”, ”stadium” and ”olympic” which were clearly relevant to the London olympic Games in 2012.

In the third cluster we see a event that talks about the Japan earthquake in Tohoku. In this event, the geo-spatial visualization actually contain two major spatial centers, one in the coastal city where the earthquake impacted most and the other is in Tokyo. This happens when the event location is different from the location of interest on this event. In this case, although the damage was made to the city of Tohoku, the majority of the discussions and reports came from Tokyo. The actual date of the event is March 11, 2012 and the temporal distributions of both Twitter and newspaper centered around March 18. Although this time newspaper was ahead of Twitter in terms of event time, the difference is quite minor. High frequency words in this event include ”earthquake”, ”nuclear”, ”radiation”, ”japanese” since the earthquake resulted in a massive nuclear disaster in a nuclear plant near Okuma, Fukushima.

Finally, the last cluster we will discuss covers the event of Egypt revolution. The event took place in Cairo, the capital city of Egypt from Jan 25 to Feb 11 on 2012. Our temporal visualizations show that tweets lead the discussion followed by newspapers. Since the Egypt revolution is well known for its usage of Twitter as a tool to assemble revolution gatherings, our model thus connects nicely to existing literature [75]. The event eventually resulted in the overthrown of the president Mubarak of Egypt. In the word cloud we see high frequently appeared words such as ”egypt”, ”mubarak”, ”police”, ”protesters” and ”revolutions”.

In summary, our study of real-world events shows two important ways in which our model can be used relative to other event-detection models. First, we can rapidly use temporal, spatial



(a) Hurricane Irene (b) London Olympic (c) Japan earthquake (d) Egypt revolution

Figure 5.3: Topical, spatial and temporal characteristics of four real life events discovered in our data sets. In the temporal visualizations, red histograms and curves represent Twitter data and clusters while the blue histograms and curves represent newspaper data and clusters

Table 5.3: Fact Sheet of Discovered Events

Name	Region	Dates
Hurricane Irene	New York City	Aug 21 - Aug 30, 2011
London Olympics	London	Jul 27 - Aug 12, 2012
Japan Earthquake	near Tohoku	Mar 11, 2011
Egypt Revolution	Cairo	Jan 25 - Feb 11, 2011

and topical information in tandem to better understand certain events. This may allow content providers to generate or provide hyper-relevant content to consumers rapidly by focusing only on content fitting all three of these dimensions. Second, and perhaps more importantly, is that our model is the first we are aware of to be able to consider differences in reporting particular events between news media and Twitter. While these two media systems are closely intertwined, a pattern that emerges from our results is that although Twitter is more reactive than news media, this does not necessarily mean that tweets "beat" news media to coverage. Rather, Twitter may react to events that occur after a major event (e.g. Irene) and thus may fall behind more traditional news cycles which are interested in "hyping up" events in anticipation of coverage. In cases where an event does not necessarily have such a "build-up" phase (i.e. earthquakes), both news and Twitter respond almost immediately, and thus the reactionary nature of Twitter is also not likely to generate a leading indicator.

Rather, our results suggest that only in cases where on-the-ground actors may actually have a hand in *generating* the content that is later utilized by news media, as was the case for those watching the Olympic games and those on the ground tweeting about the early days of the Arab Spring, where Twitter acts as a leading indicator to news media content. While this finding is initial, it may have important consequences for future, more directed work on the nature of the relationship between the news media and Twitter across different types of events.

5.6.3 Numerical Results

Finally, in order to validate that the model is able to capture important generalizations about coverage of events across media, time, space and text documents, we run predictions on missing meta information (e.g. locations and timestamps) of the documents as well as perplexity results on held out testing data sets in order to quantify the performance of our model under different parameter settings. For each of the four data sets, we reserve 90% for training and the other 10% for testing purposes. Three prediction tasks are evaluated for each of the data set: prediction of missing location coordinates of documents, prediction of missing timestamps of documents and the perplexity of testing data set, as defined in Equation 5.15. We vary parameters that are specific to the prediction task and compared our full Bayesian model with baseline models that are structural ablations of our full model. Table 5.4 lists our experiment settings for each prediction task and the baseline models used for comparison. Here "full" refers the full Bayesian model proposed in Figure 5.1 without any modification while "no time", "no location" and "no background word" refer to the baseline models (structural ablations of our model) with their respective time, location and background word components eliminated.

$$perp(\mathcal{D}_{test}) = -\frac{\Sigma_d \Sigma_n \log p(w_{d,n}|s_d = k, c, \phi_k, \phi_c)}{\Sigma_d \Sigma_n 1} \tag{5.15}$$

We present the experimental results in Figure 5.4. In the figure, sub-figures in a given column represent those for which the same experiment task is performed. Sub-figures in the same row were trained and tested on the same data set. For the prediction results on locations (column 1), we use Vincenty's formula [80] to evaluate the distance (in miles) between the actual locations of documents in the held out data set and the predict locations of those documents. We then vary the Dirichlet prior η_π on location distributions and compare the results between our full model

Table 5.4: Experiment Settings for Quantitative Results

Task	Variable	Models Considered
Predict Location	η_π	full, no time
Predict Time	λ	full, no location
Perplexity	η_k	full, no background word

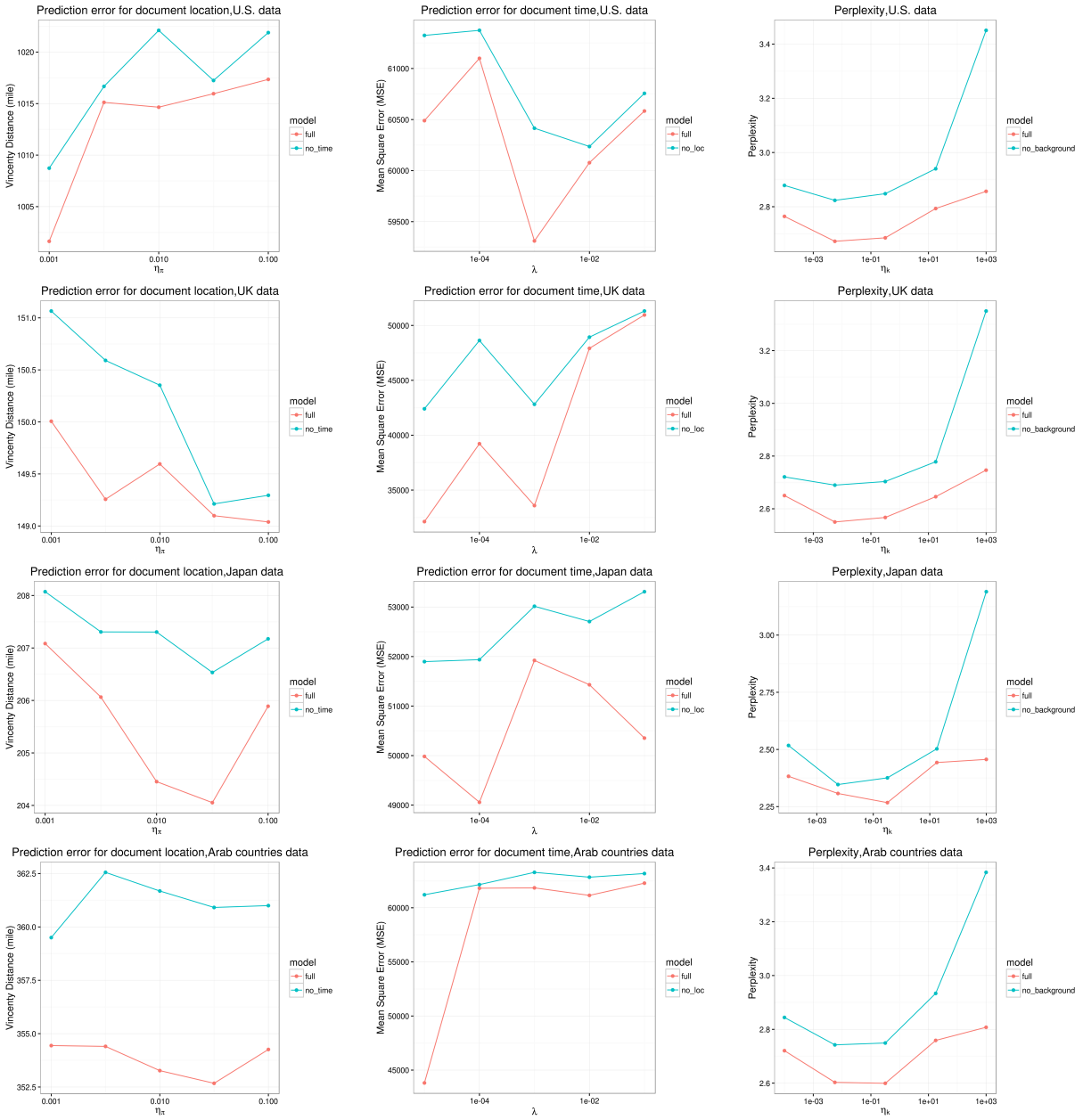
and the "no_time" model. We see that across all four datasets, the full model consistently has a lower error in terms of miles from the post's actual position compared to the baseline model, which indicates a better performance. The trends of the plots also suggest that there exists a specific value of the hyper-parameter η_π that minimizes the location prediction errors. However, this value is sensitive to the data and might needed to be turned in a case by case basis.

In the second column of Figure 5.4, we illustrate the prediction error of document time when we vary the hyper-parameter λ, which controls how much correlation we want to pose between the temporal distributions of newspaper and tweets. We report the results by evaluating the Mean Square Error (MSE) of prediction when compared to the true document time. When calculating MSE, we turned both the predicted and the true timestamps into days. We see here that the performance of the model is quite sensitive to the value of λ and for some data sets there exist multiple optimal values. Finally, we looked at how well our full model is able to predict the words in the left out documents as compared to a baseline model that does not model background distributions of terms, as is common in the literature. In column three of Figure 5.4, we see the full model is a clear winner over this baseline model in terms of text perplexity compared to the one that doesn't have the background word components. This suggested the importance of having such a component in order to filter out non-event words and to boost performance. We also experimented how changing η_k, which is the cluster word distribution can affect perplexity results on our model. The perplexity results suggest an optimal value of around 0.1 for best performance.

5.7 Discussion

In this chapter, we proposed a Bayesian non-parametric model to discover latent social events from multiple media types based on documents with spatial, temporal meta information. The model improves over prior models in the literature in that it is a) non-parametric, allowing us to model an infinite number of such events, b) can be trained online and thus can expand to streaming data, and c) models documents from multiple media simultaneously.

The model we develop is able to detect major events from four data sets that consist of Twitter and newspaper data such as the events of Olympic Games and 2011 Japan earthquake. By tuning the temporal penalty parameter λ, we showed that we could restrict the models to discover events with tight gap between their temporal distributions of different media, which greatly helped the algorithm to generate interpretable results. Leveraging this, we showed some interesting relationships between the type of event occurring and whether news media or social media respond first to the event, or whether they respond at approximately the same time. Results suggest that novel events with "social build-up" (e.g. the Arab Spring) are likely to be captured



(a) Vincenty Distance of document location prediction

(b) Mean Square Error (MSE) of document time prediction

(c) Perplexity of testing documents

Figure 5.4: Document location and time predictions as well as perplexity for the held out testing data set

first on social media, recurring events with build-up (e.g. hurricanes) are more likely to be captured by news media, and truly sudden events (e.g. earthquakes) are likely to be captured at approximately the same time by both social and news media. There are many potential ways we can expand the model. First, hierarchical structures of events can be studied by using techniques such as recursive Chinese Restaurant Process. Such a model would allows us to explore general events as well as their sub-events. Second, our current model does not allow the variance of the temporal distribution to differ for each media, which might not represent the real data. Future work should be done to address this issue. Finally, our model doesn't not penalize location centers that are far apart when learning the model. This allows events to freely take spatial centers but might also make the model too sensitive to outliers. Future work should address these and other issues to better understand how events are covered and arise across multiple media.

Chapter 6

Conclusions and Future Work

6.1 Summary of Contributions

In this thesis, I proposed three statistical models to discover latent social events. Those models are extensions of Bayesian topic modeling by expanding cluster distributions to spatial and temporal domains in addition to topic distributions. We start the thesis by arguing that applying models with these expanded domains on data sets such as geo-tagged tweets and newspaper articles help us to define and explore a completely new problem, namely event discovery. In the event discovery problem, our goal is to learn latent spatial, temporal and topical distributions of social events by scanning text data with spatial and temporal meta-information. A general insight into how this might work is that newspaper articles and geo-tagged tweets sent by mobile devices contain critical information on major events. Although there exist non-event centered tweets and news, we are able to detect major events by utilizing structures to trap data that is not event related. This work distinguish itself from prior work on event discovery by its rigorously defined methodologies that are rooted in Bayesian statistics and machine learning techniques.

I start the thesis by studying a simple proof of concept parametric model to extract event clusters from geo-tagged Twitter data. In this model, each document has an event identifier with its spatial coordinates, timestamp and text to be drawn from a multi-dimensional cluster distribution that is unique to each event. To ensure that only event related words are included in the event clusters, we created universal topic structures to trap spatial, temporal and background words. This structural design also helps the model to learn regional specific linguistic characteristics as well as temporal related languages, which turned out to be useful in tasks such as missing location and timestamp information predictions. Scalable inference is developed for the model and I am able to discover major events such as the Egypt revolution and the foot and mouth outbreaks.

As we have discussed in the introduction section of this thesis, the event discovery problem is needed for mainly two reasons. First, to generate meaningful summaries of social events from text data with meta information. And second, to provide timely and accurate descriptions of the events as early as possible. To better address those two needs and improve the independent event model, I first studied a model to deal with temporal evolutionary events, which provides

more meaningful and also more general forms of events. I then studied a model to combine information from both Twitter and Newspaper to form accurate event clusters in a timely fashion.

In the event evolutionary model I explored a non-parametric model that is able to detect evolutionary event clusters. The motivations here is that many social events are in constant states of changing and static models are not sufficient to capture its dynamics. Another characteristic of this model is that the number of clusters increase as the pattern of the data changes. Because of the choice of Logistic-normal distributions on both the cluster location center and the cluster topic center and its non-conjugacy to Multinomial distribution, techniques are developed to approximate the integral of the posterior. I implemented a parallel algorithm that can efficiently handle millions of tweets our approximation beat many of the existing solutions for similar models reported in prior work.

I also studied a non-parametric event discovery model by considering the differences between different data sources. The main idea behind this model is that each media has its own perception of when the event is happening and therefore a more complicated model needs to be developed in order to learn the event information from multiple data sources together. In this model I developed a mechanism to control the dependencies between the temporal distributions of different media types. Those two distributions are restricted to be close to each other to represent the same event but not completely overlapped. By applying the algorithm on real data sets, I found that newspaper comes first on events such as natural disasters while twitter leads public events such as social revolutions and sports events.

Finally, I note that although the models studied in this thesis are motivated by the event discovery problem, they can serve as general document clustering techniques on text data with spatial and temporal meta information. Many techniques used in the thesis, such as the solution to the Laplace Approximation in the Recurrent Chinese Restaurant Process can be applied to a variety range of problems that model evolutionary clusters.

6.2 Scalability

In this section I provide a summary of the complexity analysis for each algorithm in this thesis. Table 6.1 illustrated the notations used in this section and Table 6.2 listed the actual complexity of the algorithms. We see here that independent event model can only run in batch model and it cannot run on streaming data. Its run time complexity is high because of the EM swaps and the iteration based gradient descent in its M step. Both the latter two models support streaming data and their complexity are proportional to the number of particles, F. Since the algorithms can run in parallel, F is heavily factorized. The media structure model has much lower complexity thanks to the Metropolis algorithm used when infer the cluster identity k for each document.

Table 6.1: Notations

Symbol	Comments
F	number of particles
G	number of MCMC steps
M	number of gradient descent iterations
EM	number of EM swaps
W	number of words in the data set/streaming batch
K	number of clusters

Table 6.2: Complexity of the Algorithms

Name	Complexity	runtime on 1m data	Streaming?
Independent Event Model	O((W*K*G+M)*EM)	1 hr	No
Evolutionary Model	O((W+B)*K*G*F)	2 hrs	Yes
Media Structure Model	O((W+B+K)*G*F)	30 mins	Yes

6.3 Frequency and Aggregation

One of the issues in the thesis is how can we prevent wide spread events to dominate the cluster which forced the less known events to blend in into the results. In reality, I find it to be less of a problem of frequency than a problem of distinguishable pattern. No matter how frequent an event is, its patterns are defined by the documents that were identified to be in the same event. In other word, it will not interrupt with other events unless its pattern becomes so ambiguous os that all the documents can be attributed to this event. Experimentally, we can wisely choose the hyper-parameter in order to turn up the sensitivity of the model and avoid super clusters. For example, we can turn down the Dirichlet hyper-parameter so that clusters will only tolerate documents that are almost identical to the ones already in the cluster. This will inevitably create more clusters as the algorithm find it less likely to find existing clusters to match new documents. However, the overall integrity of the algorithm is preserved and frequent events should be able to stay in their own clusters without interrupting less frequent events.

In the section of evolutionary events, data needs to be aggregated into epochs first before the algorithms can be executed. The particular choice of this time interval is an issue and it can affect not only the choice of hyper-parameters in the model but also the results and its interpretability. In this thesis, I use 1 month as the interval to divide data into epochs. However, few work have been done to discuss the optimal time window. Instead of finding the optimal window, one alternative approach is to use models such as the Hawkes process to deal with continuous time steps which avoided the needs to divide data into epochs in the first place.

6.4 Limitations

There are many limitations to the work in this thesis and here we list several major ones.

Firstly, the assumption that datasets have to contain sufficient amount of event centered information is required for algorithms in this thesis to work. Geo-tagged Tweets and Newspaper data are rich in event related information in general. However, it might be hard to generalize some of the conclusions in this thesis by using other datasets that contains little event centered information. Although various measures have been taken place to prevent non-event centered documents to disturb the patterns of the true event clusters, it is still likely that certain non-event centered patterns are strong enough to form actual event clusters. Twitter, for example, contains a high volume of spamming information and many of them are geo-tagged (probably an effort to avoid themselves from being caught by Twitter's spammer detection system). Many of these documents form meaningless event clusters and it's fairly difficult to filter them out during the training or testing stage. Instead, reliable preprocessing practices have to be implemented in order to prevent these documents from generating meaningless clusters.

Secondly, the spatial components in models proposed in Section 4 and Section 5 do not penalize distances between location centers within an event cluster. In those models, event location distribution is modeled as a mixture of Gaussian with global centers across all events as opposed to a single Gaussian distribution seen in Section 3. The event location distribution is differed from each event by their weights on those Gaussian distributions and this structural design is developed to allow events with multiple spatial centers. Although in those models I have developed a smoothing parameter to control (roughly) the number of location centers one event can have, it is possible that those centers are far away from each other and there is currently no mechanism built to discourage this from happening. In a radical case my model might generate event clusters that contain locations from the opposite sites of the earth, which is possible in principle but not usual in practice.

Thirdly, the assumption that location and time distributions are independent to each other might not be the best one to facilitate the learning of the event discovery algorithm. Take the downtown area of a city for example, its activities are only likely to occur during normal business hours while businesses around a night bar street will see activities during the midnight. Allowing the spatial and temporal distributions to correlate to some extent helps algorithm to generate spatial-temporal patterns like this, which will eventually benefit the learning of subsequent events that occurs later on.

Fourthly, the algorithms in this thesis did not provide a mechanism to phase out old events. Take the non-parametric models for example, the fact that these algorithms can work on streaming data in theory allows them to run indefinitely as long as new data is continuously being provided. However, since new event clusters will be generated over time, our algorithms will eventually consume all of the memory on the machine as the number of events become too large. Even though this rarely happens since the cluster number usually increases in the log scale of the number of samples, large number of clusters will significantly slow down the sampling

algorithm. Since time is used as one of the dimensions of the event distributions, there needs to have a mechanism to clear away old events that won't have any chances to attract upcoming data. This will significantly speed up the systems that have been running for a long time and will also avoid letting those old events to occupy all the memory in the machine. If this mechanism is built, algorithms can run forever and there will be no reason to restart them.

Finally, the way time is handled put strong limitations on the types of events that can be detected using the models proposed in this thesis. Throughout the thesis I have developed several ways to model time of an event ranging from a single Gaussian distribution in Section 3, discrete time intervals in Section 4 to two dependent Gaussian distributions around a common mean in Section 5. Those models server their purposes well in teams of detecting event clusters of with particular forms of temporal distributions. However, they are not able to detect events with multiple temporal peaks or event periodicity. A better model on the temporal distributions of events needed to be developed to handle events with different temporal characteristics.

6.5 Future Work

Future work can be done in many different areas to expand and improve the event detection models. Here we list a few of them.

6.5.1 Hierarchical Models

Many events exhibit hierarchical structures in nature. For example, take the Olympic Games as a general event, there might be sub-events about a specific item within the general one. Another example would be the Japanese earthquake and its subsequent aftershock, which are different events with clear dependencies. If we apply the models in this thesis, those events would have appeared as independent ones and their structural dependency information will be lost. Another benefit of using a hierarchical model is that sub-events with little training data can be discovered since parent events can provide much of the background information. In a model that assumes independent event, however, pattern related to the sub-events have to be strong enough in order to be considered as an independent event.

There are several existing work explore such a hierarchical structure for graphical models. For example, the Hierarchical Dirichlet Process [78] assumes a two layer Dirichlet hierarchy for topic models. In our case, a two layer hierarchy is probably not enough in which case we should explore nested hierarchical models such as the ones found in [4, 40, 63]. In a nested hierarchical Dirichlet model, clusters are organized in a tree with descendants of the tree being the sub-clusters of their parents. Both the number of nodes in each layer and the depth of the tree can be infinite. By applying such models into the event discovery problem, we will be able to detect the entire structure of the events and present them in a tree structure. In addition to modeling events as a tree, we are also able to model them as a graph [94], which is a more general approach to discover the cluster dependencies that are not restricted to tree structures.

One of the caveats here is that nested hierarchical Dirichlet models are computational intensive and efficient inference techniques need to be developed in order to handle the scale of the data we use in this thesis.

6.5.2 Integrating Mutually Excited Point Process

As we already mentioned in the limitation section, one of the issues in the models proposed in this thesis is that their temporal components do not allow for events that occur intermittently or periodically. Using the models in this thesis, events exhibiting such characteristics will be recognized as separate ones and we might ended up having many events having similar topical information. This is because our models only assume a single peak or a single interval in the temporal distributions and one an event "dies out", it will have very little chance to accept new documents and thus new clusters have to be created.

One solution to solve this issue is to use mutually excited point process or otherwise referred to as Hawkes process [42] or survival models [53, 54] to model the temporal distributions of event clusters. In a Hawkes Process, the probability of generating a specific observation is dependent on all the historical data points. In other words, each prior sample will have some mutual impacts on future data points. As a result, we are able to model event clusters with multiple peaks on temporal distributions and event periodicity. By using techniques such as Dirichlet-Hawkes process [34], the Hawkes process is naturally integrated into the non-parametric models and amount of excitement around a particular time plays a role in determining the event clusters.

Another benefit of using models such as Dirichlet-Hawkes process is that we are able to predict certain characteristics of new-born events when we correlate the coefficients of the kernel functions with frequently occurred words. For example, if the word "disaster" is highly correlated with the kernel function that triggers a life cycle of 2 weeks, seeing a new cluster that contains keyword "disaster" will probably mean that the event will last for about 2 weeks. The same technique can be used to predict spatial life cycles by expanding the point process to include geo-coordinates.

6.5.3 Extensions via Deep Learning

Statistical techniques that utilize layered network structures called deep learning [50] can be powerful learning devices for supervised learning. Many deep neural network models have been proposed, such as the feed forward neural networks[9], convolutional neural networks [48] and recurrent neural networks [59]. These models have been shown to improve the state of the art computer vision [87, 88] and computational linguistic problems[89, 90**?**] when GPU-accelerated devices are used. As a by-product of the supervised task, neural networks can often be used as a latent representations of the knowledge. For example, word2vec [60] is essentially one of the hidden layers of the neural network model trained for a supervised task and can be considered as a latent representation of the words.

By utilizing neural networks in text data with spatial and temporal information, we are able to train statistical models that aims to predict some of the missing information in the documents. This process, on the other hand, produces a new representation of the data, which is similar to our event clusters that are studied in graphical models. One of the most significant advantages of using this approach is scalability. Neural network methods today are highly scalable thanks for the varies inference techniques that are both efficient and parallelizable as well as the advancement of computational devices. However, by using neural networks, we lose much of the interpretations compared to the results generated by Bayesian approaches. Much of the information in the hidden layers are open to speculations and their exact statistical meanings are usually obscure.

6.5.4 Improving Efficiencies

The models proposed in the thesis are fairly computationally intensive and scalability becomes the leading factors to limit the applications of the algorithms into large scale datasets. Much of the computational complexity comes from the fact that the topical component needs to scan the entire corpus each time when a new round of sampling is taken place. In the thesis, I didn't explore much of the options to speed up this process since my focus in on the modeling and the subsequently interpretations. I note that there are several known techniques that can potentially be used to further reduce the running time. One example would to utilize the sparsity of the sampling [91]. Another option is to explore the possibility of Alias Sampling [51] when doing topic modeling.

Bibliography

[1] Amr Ahmed and Eric P Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *SDM*, pages 219–230. SIAM, 2008. 1, 2.3, 4.1, 4.2.1, 4.2.2, 4.2.3, 4.2.3, 4.3, 4.3, 4.4.2, 5.2.2

[2] Amr Ahmed, Qirong Ho, Choon H Teo, Jacob Eisenstein, Eric P Xing, and Alex J Smola. Online inference for the infinite topic-cluster model: Storylines from streaming text. In *International Conference on Artificial Intelligence and Statistics*, pages 101–109, 2011. 4.2.4, 5.2.2, 5.2.3, 5.4

[3] Amr Ahmed, Mohamed Aly, Joseph Gonzalez, Shravan Narayanamurthy, and Alexander Smola. Scalable inference in latent variable models. In *International conference on Web search and data mining (WSDM)*, pages 123–132, 2012. 4.2.1

[4] Amr Ahmed, Liangjie Hong, and Alexander J Smola. Nested chinese restaurant franchise process: Applications to user tracking and document modeling. In *ICML (3)*, pages 1426–1434, 2013. 2.3, 4.2.2, 5.2.2, 6.5.1

[5] David J Aldous. *Exchangeability and related topics*. Springer, 1985. 2.3

[6] Lisa Anderson. Demystifying the arab spring. *Foreign Affairs*, 90(3):2–7, 2011. 1, 3.5, 3.5.2

[7] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003. 3.4

[8] Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584, 2001. 2.3

[9] George Bebis and Michael Georgiopoulos. Feed-forward neural networks. *IEEE Potentials*, 13(4):27–31, 1994. 6.5.3

[10] Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 389–398. Association for Computational Linguistics, 2011. 2.1, 3.1, 3.2.1

[11] Alex Beutel, Amr Ahmed, and Alexander J Smola. Accams: Additive co-clustering to approximate matrices succinctly. In *Proceedings of the 24th International Conference on World Wide Web*, pages 119–129. ACM, 2015. 2.3

[12] David Blackwell and James B MacQueen. Ferguson distributions via pólya urn schemes. *The annals of statistics*, pages 353–355, 1973. 2.3

[13] D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007. URL http://www.jstor.org/stable/10.2307/4537420. 4.3

[14] David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 18:147, 2006. 4.2.1

[15] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006. 2.2, 4.1, 4.2.1, 4.2.3, 5.2.1

[16] David M Blei and John D Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007. 5.2.1

[17] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 2.2, 3.2.3, 3.3.3

[18] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944937. 4.1, 4.2.1, 5.1, 5.2.1

[19] David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010. 4.2.2, 5.2.2

[20] Jordan L Boyd-Graber and David M Blei. Syntactic topic models. In *Advances in neural information processing systems*, pages 185–192, 2009. 2.2

[21] Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 211–218. ACM, 2002. 2.1

[22] Thorsten Brants, Francine Chen, and Ayman Farahat. A system for new event detection. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 330–337. ACM, 2003. 2.1, 3.2.1

[23] Kevin Robert Canini, Lei Shi, and Thomas L Griffiths. Online inference of topics with latent dirichlet allocation. In *AISTATS*, volume 9, pages 65–72, 2009. 4.2.4, 5.2.3

[24] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007. 5.2.3

[25] K. M. Carley, Wei Wei, and Kenneth Joseph. High dimensional network analysis. In *Big Data Over Networks, Robert Cui (Eds)*. Cambridge University Press. 2

[26] George Casella and Edward I George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992. 3.2.3

[27] Allison JB Chaney, Hanna Wallach, Matthew Connelly, and David M. Blei. Detecting and Characterizing Events. 2015. URL http://ajbc.io/projects/papers/ChaneyWallachConnellyBlei2016.pdf. 5.1, 5.2.4

[28] Jonathan Chang and David M Blei. Hierarchical relational models for document networks. *The Annals of Applied Statistics*, pages 124–150, 2010. 2.2

[29] Jianfei Chen, Jun Zhu, Zi Wang, Xun Zheng, and Bo Zhang. Scalable inference for logistic-normal topic models. In *Advances in Neural Information Processing Systems*, pages 2445–2453, 2013. 4.2.3

[30] Francesca Comunello and Giuseppe Anzera. Will the revolution be tweeted? a conceptual framework for understanding the social media and the arab spring. 23(4):453–470. ISSN 0959-6410. doi: 10.1080/09596410.2012.712435. 3.5

[31] Robert M Corless, Gaston H Gonnet, Dave EG Hare, David J Jeffrey, and DE Knuth. Lambert?s w function in maple. *Maple Technical Newsletter*, 9:12–22, 1993. 4.4.2

[32] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM, 2014. 2.2

[33] Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001. 4.2.4, 5.1, 5.2.3, 5.4

[34] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 219–228. ACM, 2015. 2.3, 4.2.4, 5.2.2, 5.2.3, 6.5.2

[35] Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004. 2.2, 5.2.1

[36] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973. 1, 2.3, 4.2.2, 4.3, 5.1, 5.2.2

[37] Seth Flaxman, Andrew Gordon Wilson, Daniel B Neill, Hannes Nickisch, and Alexander J Smola. Fast kronecker inference in gaussian processes with non-gaussian likelihoods. In *International Conference on Machine Learning*, volume 2015, 2015. 2.3

[38] Matthias Gall, Jean-Michel Renders, and Eric Karstens. Who broke the news?: an analysis on first reports of news events. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 855–862. International World Wide Web Conferences Steering Committee, 2013. URL `http://dl.acm.org/citation.cfm?id=2488066`. 5.1

[39] Zoubin Ghahramani and Geoffrey E Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996. 4.1

[40] DMBTL Griffiths and MIJJB Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:17, 2004. 2.3, 6.5.1

[41] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the Na-*

tional academy of Sciences, 101(suppl 1):5228–5235, 2004. 2.2, 3.4.1, 3.4.1, 4.2.1, 5.2.1, 5.4

[42] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971. 6.5.2

[43] Starr R. Hiltz and Murray Turoff. Structuring computer-mediated communication systems to avoid information overload. *Communications of the ACM*, 28(7):680–689, 1985. URL `http://dl.acm.org/citation.cfm?id=3895`. 5.1

[44] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, pages 769–778. ACM, 2012. 2.2, 3.2.2, 3.6, 4.1, 5.2.1

[45] Michael Irwin Jordan. *Learning in Graphical Models:[proceedings of the NATO Advanced Study Institute...: Ettore Mairona Center, Erice, Italy, September 27-October 7, 1996]*, volume 89. Springer, 1998. 2.2, 3.2.3

[46] Kenneth Joseph, Chun How Tan, and Kathleen M Carley. Beyond local, categories and friends: clustering foursquare users with latent topics. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 919–926. ACM, 2012. 2.2

[47] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 4.1

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 6.5.3

[49] Giridhar Kumaran and James Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304. ACM, 2004. 2.1, 3.1, 3.2.1

[50] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015. 6.5.3

[51] Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 891–900. ACM, 2014. 2.2, 4.2.1, 5.2.1, 6.5.4

[52] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009. 2.2

[53] Ziqi Liu, Alexander J Smola, Kyle Soska, Yu-Xiang Wang, and Qinghua Zheng. Attributing hacks. *arXiv preprint arXiv:1611.03021*, 2016. 6.5.2

[54] Ziqi Liu, Alexander J Smola, Kyle Soska, Yu-Xiang Wang, and Qinghua Zheng. Joint hacking and latent hazard rate estimation. *arXiv preprint arXiv:1611.06843*, 2016. 6.5.2

[55] Gilad Lotan, Erhardt Graeff, Mike Ananny, Devin Gaffney, Ian Pearce, and Danah Boyd.

The revolutions were tweeted: Information flows during the 2011 tunisian and egyptian revolutions. *International Journal of Communication*, 5:1375–1405, 2011. 3.5

[56] Tamas Matuszka, Zoltan Vinceller, and Sandor Laki. On a keyword-lifecycle model for real-time event detection in social network data. In *Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on*, pages 453–458. IEEE, 2013. 2.1, 3.2.1

[57] R Daniel Mauldin, William D Sudderth, and SC Williams. Polya trees and random distributions. *The Annals of Statistics*, pages 1203–1221, 1992. 2.3

[58] Jon D Mcauliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008. 4.2.1

[59] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010. 6.5.3

[60] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 6.5.3

[61] Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, 1(2σ2):16, 2007. 5.4

[62] David Newman, Chaitanya Chemudugunta, and Padhraic Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686. ACM, 2006. 2.2

[63] John Paisley, Chong Wang, David M Blei, and Michael I Jordan. Nested hierarchical dirichlet processes. *arXiv preprint arXiv:1210.6738*, 2012. 2.3, 6.5.1

[64] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010. 3.1

[65] André Panisson, Laetitia Gauvin, Marco Quaggiotto, and Ciro Cattuto. Mining concurrent topical activity in microblog streams. *arXiv preprint arXiv:1403.1403*, 2014. 2.1, 3.2.2

[66] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011. 5.1

[67] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006. 2.3

[68] Alan Ritter, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM, 2012. URL `http://dl.acm.org/citation.cfm?id=2339704`. 2.1, 3.1, 3.2.2

[69] Alan Ritter, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM, 2012. URL `http://dl.acm.org/citation.cfm?id=2339704`. 5.1

[70] Alan Ritter, Evan Wright, William Casey, and Tom Mitchell. Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web*, pages 896–905. ACM, 2015. URL `http:`

//dl.acm.org/citation.cfm?id=2741083. 5.1, 5.2.4

[71] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004. 2.2

[72] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010. 2.1, 3.1, 3.2.2, 3.6

[73] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772777. URL http://doi.acm.org/10.1145/1772690.1772777. 5.2.4

[74] Tadej Štajner and Marko Grobelnik. Story link detection with entity resolution. In *WWW 2009 Workshop on Semantic Search*, 2009. 2.1, 3.2.1

[75] Kate Starbird and Leysia Palen. (How) will the revolution be retweeted?: information diffusion and the 2011 Egyptian uprising. In *Proceedings of the acm 2012 conference on computer supported cooperative work*, pages 7–16. ACM, 2012. URL http://dl.acm.org/citation.cfm?id=2145212. 5.6.2

[76] Yusuke Takahashi, Takehito Utsuro, Masaharu Yoshioka, Noriko Kando, Tomohiro Fukuhara, Hiroshi Nakagawa, and Yoji Kiyota. Applying a Burst Model to Detect Bursty Topics in a Topic Model. In Hitoshi Isahara and Kyoko Kanzaki, editors, *Advances in Natural Language Processing*, volume 7614 of *Lecture Notes in Computer Science*, pages 239–249. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-33982-0. URL http://www.springerlink.com/content/8713w7x712k38630/abstract/. 5.2.4

[77] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006. 2.3

[78] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 2012. 4.2.2, 5.2.2, 6.5.1

[79] Dennis Thom, Harald Bosch, Steffen Koch, Michael Worner, and Thomas Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE*, pages 41–48. IEEE, 2012. 3.1

[80] CM Thomas and WE Featherstone. Validation of vincentys formulas for the geodesic using a new fourth-order extension of kiviojas formula. *Journal of Surveying engineering*, 131 (1):20–26, 2005. 5.6.3

[81] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006. 3.4

[82] Xiaogang Wang and Eric Grimson. Spatial latent dirichlet allocation. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1577–1584. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3278-spatial-latent-dirichlet-allocation.pdf.

2.2

[83] Wei Wei, Kenneth Joseph, Wei Lo, and Kathleen M Carley. A bayesian graphical model to discover latent events from twitter. In *Ninth International AAAI Conference on Web and Social Media*, 2015. 2.2, 4.1, 4.2.1, 4.3, 5.1, 5.2.1, 5.2.4

[84] Chao-Yuan Wu, Alex Beutel, Amr Ahmed, and Alexander J Smola. Additive co-clustering of gaussians and poissons for joint modeling of ratings and reviews. In *NIPS workshop on Nonparametric Methods for Large Scale Representation Learning*, volume 4, 2015. 2.2

[85] Chao-Yuan Wu, Alex Beutel, Amr Ahmed, and Alexander J Smola. Explaining reviews and ratings with paco: Poisson additive co-clustering. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 127–128. International World Wide Web Conferences Steering Committee, 2016. 2.2

[86] Bishan Yang and Tom Mitchell. Joint Extraction of Events and Entities within a Document Context. 2016. URL `http://www.cs.cmu.edu/~bishan/papers/joint_event_naacl16.pdf`. 5.2.4

[87] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. *arXiv preprint arXiv:1511.02274*, 2015. 6.5.3

[88] Zichao Yang, Marcin Moczulski, Misha Denil, Nando de Freitas, Alex Smola, Le Song, and Ziyu Wang. Deep fried convnets. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1476–1483, 2015. 6.5.3

[89] Zichao Yang, Zhiting Hu, Yuntian Deng, Chris Dyer, and Alex Smola. Neural machine translation with recurrent attention modeling. *arXiv preprint arXiv:1607.05108*, 2016. 6.5.3

[90] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016. 6.5.3

[91] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 937–946. ACM, 2009. 2.2, 4.2.1, 5.2.1, 6.5.4

[92] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 186–194. ACM, 2012. 2.2

[93] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. Lightlda: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1351–1361. International World Wide Web Conferences Steering Committee, 2015. 2.2

[94] Aonan Zhang and John Paisley. Markov mixed membership models. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 475–483, 2015. 6.5.1

[95] Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. Towards real-time

summarization of scheduled events from twitter streams. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 319–320. ACM, 2012. 2.1, 3.2.1