Learning Latent Event Representations:
Structured Probabilistic Inference on Spatial, Temporal and Textual Data

Wei Wei

Fall 2015

Societal Computing Program
Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Kathleen M. Carley, Chair, CMU
Huan Liu, ASU
Tom Mitchell, CMU
Alexander J. Smola, CMU

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

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 textual data. The event detection problem can be treated as a generalization of the topic modeling problem where events can be considered as topics that are augmented by location and time. Preliminary models can effectively learn the representations of major events covered in a corpus of Twitter data and can also be used for various prediction tasks such as predicting the spatial coordinates, time stamps of the documents as well as estimating life cycles of new born events.

The approaches proposed in this thesis are largely based on Bayesian non-parametric methods to deal with steaming data and unpredictable number of data clusters. The research proposed 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.

1 Introduction

With the prevalence of mobile and Internet services, datasets of text today are massive. Understanding such datasets requires models that are both scalable and effective at conveying subsets of information found within the data. Structured probabilistic inference techniques have proved to be effective in modeling text data with complex latent structures. For example, latent Dirichlet allocation [9] is a structured inference technique that has successfully used to study hierarchical latent structures of text data. Unlike the unstructured techniques, structured probabilistic models can provide rich latent representations of data. These latent representations can be very useful in interpreting model results, which often leads to a better understanding of the stochastic dependencies among the data.

Text data today, however, is not only only large in size. It also comes with a significant amount of meta information. For example, consider a tweet sent through a mobile phone device. The message will contain not only the text body but also meta data such as time stamp and geo-location coordinates. Unfortunately, time and spatial information often requires different treatments than text, and existing topic modeling techniques cannot be directly applied. For example, time needs special treatment because of its periodical nature. Information tagged with time stamps on Mondays might share similar patterns. On the other hand, documents tagged with year 1990 might have different patterns from documents tagged with year 2010, which means the absolute temporal magnitudes also matters. One other example is geo-location data, which often makes the documents to exhibit unique patterns over a specific spatial region. Such facts require the development of algorithm that can be applied to spatio-temporal text data.

I propose the event detection problem, which studies how spatial, temporal and text data can be used to form meaningful latent representations of an *event*. In this thesis, an event is defined to be a stochastic distribution in space, time and text. For example, consider New Year's Eve fireworks as an event. It will have a temporal distribution with probability mass arounds Dec 31 9:00PM to Jan 1 12:00AM, a spatial distribution around various downtown areas across the world, and a topical distribution with high probability on words like "fierworks", "New Year" and "wish". Documents with meta data, such as Tweets and newspaper articles, are assumed to be drawn from one of the event distributions. Documents that talk about the same event should concentrate on a particular position in the event space with certain variance that represents the observer's perceptions and and differences of media.

To tackle this problem, I first studied a parametric model to detect events on Twitter data by assuming events remain static over time [41]. Experiments were conducted on a set of Twitter data collected over the country of Egypt during the famous Arab Spring revolutions[4]. I showed that events discovered 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 location and time of the tweets.

To improve the original event detection problem, I propose three new research avenues. First, I will relax the assumption that events can only be static by allowing them to change over time. In particular, I assume events evolve in a Markovian fashion and that both their topical distributions and their spatial distributions are dependent on those in the previous time step. By doing this, I will be able to examine how the topical focus, options as well as the spatial spreads of a particular event of interest change over time. This study of the evolution of events which will be discussed in Section 5. Second, I will study whether certain aspects of events can be predicted. I will concentrate on the temporal aspect of events and propose methods to predict event *life cycles*, which is the period of time that an event will keep being mentioned in newspaper or social media. The rationale of this research is that certain topical focus or the location of events often determine the popularity of certain events. By learning patterns of past events we will able to discover this correlation and eventually lead to a prediction on event life cycles. This will be discussed in Section 6. Finally, I will present a model to extract events from multiple media. Although social media data, in particular Twitter data, contains all the aspects of data we need to extract events, the fact that tweets are limited to 140 characters and have a unique grammar make it difficult to learn strong representations of events. On the other hand, newspaper data usually has more detailed and much higher quality text but lacks the explicit spatial meta-data. By learning events using different data sources, we will be able to learn events that are of better quality by utilizing the strength of both data sets. Additionally, we can also study the differences between different media. For example, we can study which media source come up with the information first and their differences in term of the use of language. This will be discussed in Section 7.

2 Related Work

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, [25] use cosine similarity. Other methods include Hellinger distance [12], Kullback-Leibler divergence [11] and TF-IDF similarity [37].

The second class of methods for detecting events in text are based on abnormality detection of frequent words. For example, [28] 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 [45] 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. [6], 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 [36] 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 [34] and [32] 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) [16] which is based on Singular Vector Decomposition (SVD) and Latent Dirichlet Allocation (LDA) [9] which is based on probabilistic graphical models [23]. 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 [20]. However, just like many statistical learning approaches, its application-agnostic nature allowed it to extend to other areas such as clustering region functions [43] and clustering check-in patterns [24]. The LDA model can be extended with additional meta-data, such as author-topic model [35], relational topic model [13], named entity topic model [30] Syntactic topic model [10], dynamic topic model [8], sentiment topic model [27] and Spatial LDA [39]. The computational intensive nature of LDA leads to many work that improves its efficiency by introducing different sampling techniques such as Gibbs Sampling [20], Sparse-LDA [42], Alias-LDA[26] and light-LDA [44]. Finally, probabilistic models that contains an LDA component but serves other purposes are also proposed. Examples include spatial topic pattern model [22], review aspect modeling and recommendation system [14] and event detection [41].

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 [18], Gaussian Process [33], Infinite Hidden Markov model [5] and Polya Trees [29]. 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 1 [7].

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

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 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 [3] 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 1.

Many non-parametric models related to LDA have been proposed. For example, the Hierarchical Dirichlet Process [38] 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 [19], Nested Chinese Restaurant Franchise Process [2] and Nested Hierarchical Dirichlet Process [31]. 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 [15].

3 Data Set

In order to validate our method, data with both spatial, temporal and textual information are required. GPS-enabled social media data are ideal to serve as the validation data set because they have all of the three data features. In this thesis, the experiments are conducted mainly on a Twitter data set collected from Nov 2009 to Dec 2013. The data set contains roughly 1.1 billion geo-tagged tweets from around the world collected using Twitter's gardenhose API. The garden hose API will return approximately 10% random sample of all the available geo-tagged tweets at any moment [17]. However, as discussed in the introduction, Twitter data suffers from the problem of low text quality because of its 140 character text limit and the frequent use of slang. I remedy this issue by using an auxiliary newspaper dataset collected using LexisNexis API [40]. The newspaper data does not contain explicit geo-location information as the Twitter data does. However, it will contain a much richer text which will eventually benefit the research in Section 7.

4 Completed Work: Basic Event Model

In this section, I describe a parametric version of the event model that is capable of capturing latent event representations on spatial, temporal and textual data [41]. To start with, I assume each document is associated with one event. The geo-location, time stamp and the text of that specific document are generated from the corresponding event distribution that this document belongs to. Depending on the individual's perception, the document time, location and the text can vary. However, documents belong to the same event should probabilistically centered on certain points to reflect the identify of the event.

The graphical model of the basic event model is illustrated in Figure 1. There are three components in the model. The *Event component* contains the information about a particular event, which will be explained in Section 4.1. It has E replications and contains the mean and variance parameter of event time distribution $\theta^{(T)}$ and $\sigma^{(T)}$, the mean and variance of event spatial distribution $\theta^{(T)}$ and $\sigma^{(T)}$ as well as a word distribution $\Phi^{(E)}$. The *Document component* contains the observed information of an document such as its text w, location l and time t. It also contain several latent variables such as the event index e that this document belongs to, the category distribution π and the exact category of each word z. We will see the explanations of this component in Section 4.2. And finally, we will see how the *Language component* work in Section 4.3 by introducing additional word distributions such as $\Phi^{(0)}$, $\Phi^{(L)}$ and $\Phi^{(T)}$ that will help to learn the event distributions better.

4.1 Event Component

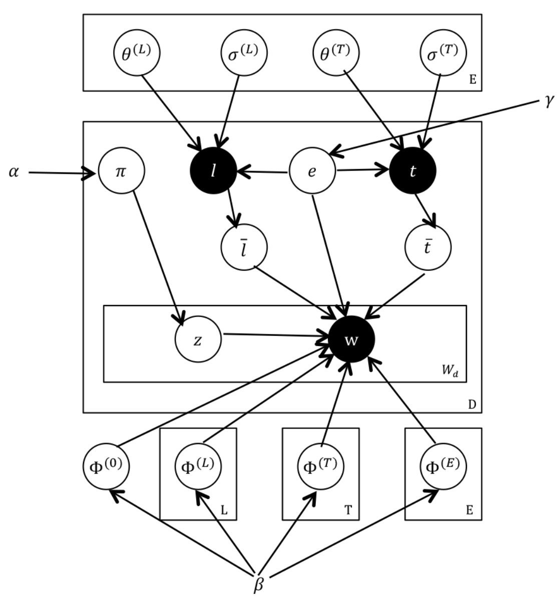


Figure 1: Illustrations of the event model in plate notations

Events are defined by three distributions. First, each event has a spatial center $\theta_e^{(L)}$ as well as a spatial variance controlled by a diagonal covariance matrix with each value defined by $\sigma_e^{(L)}$. The location of a report that belongs to event e is assumed to be drawn from a two dimensional Gaussian distribution governed by these parameters.

$$l \sim \mathrm{N}(\theta_e^{(L)}, I \cdot \sigma_e^{(L)}) \tag{2}$$

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

$$t \sim \mathrm{N}(\theta_e^{(T)}, I \cdot \sigma_e^{(T)}) \tag{3}$$

Finally, events have a topic distribution(or distribution over words). I defer the introduction of this topic distribution to the language model along with all other topic additional distributions that do not belong to a event.

4.2 Document Component

An observed document contains only three elements: observed event time t, observed event location l and a set of narrative words describing the event w. Here the observed event location must be in the format of lat/lon pair. In order to construct dependency structure between events and documents, additional latent variables must also present in the document component. First, each document contains a latent event identity e that identify 1 out of the E events that this specific document is describing. I assume a multinomial prior γ for each event identity e.

$$e \sim \mathrm{Mult}(\gamma) \tag{4}$$

Second, each word w_i in the document text 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:

$$z{\sim}\mathrm{Mult}(\pi) \tag{5}$$

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

$$\pi{\sim}\mathrm{Dir}(\alpha) \tag{6}$$

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 finding their positions on a two dimensional spatial grid and one dimensional temporal grid. These indices are used for the language model to retrieve the corresponding topics from these categories in a manner that will be introduced later.

4.3 Language Component

The language model defines how words within a document are drawn from topics (within specific categories). Topic distributions for each category are generated using a Dirichlet prior β:

$$\Phi_*^{(*)}{\sim}\mathrm{Dir}(\beta) \tag{7}$$

Each topic contains the probability of words 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 [9], since now topics are also hierarchically organized by the four different categories. For a model with one global topic (i.e. topic "0"), 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 spatial, temporal and event indices $\bar{l}$, $\bar{t}$ and e, respectively, depending on which category is being used. This is represented mathematically in Equation 8 below:

$$\begin{aligned} &P(w_i|\bar{l}, \bar{t}, e, z_i, \Phi^{(0)}, \Phi^{(L)}, \Phi^{(T)}, \Phi^{(E)}) \\ &= P(w_i|\Phi^{(0)})^{I(z_i=0)} \cdot P(w_i|\Phi^{(L)}, \bar{l})^{I(z_i=L)} \cdot P(w_i|\Phi^{(T)}, \bar{t})^{I(z_i=T)} \cdot P(w_i|\Phi^{(E)}, e)^{I(z_i=E)} \end{aligned} \tag{8}$$

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

Table 1: Spatial and temporal parameters of 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

Table 2: Top words for each event

4.4 Generative Model

The graphical model I defined above can be used as a generative model that produces new documents based on learned events. The generative process is as follows:

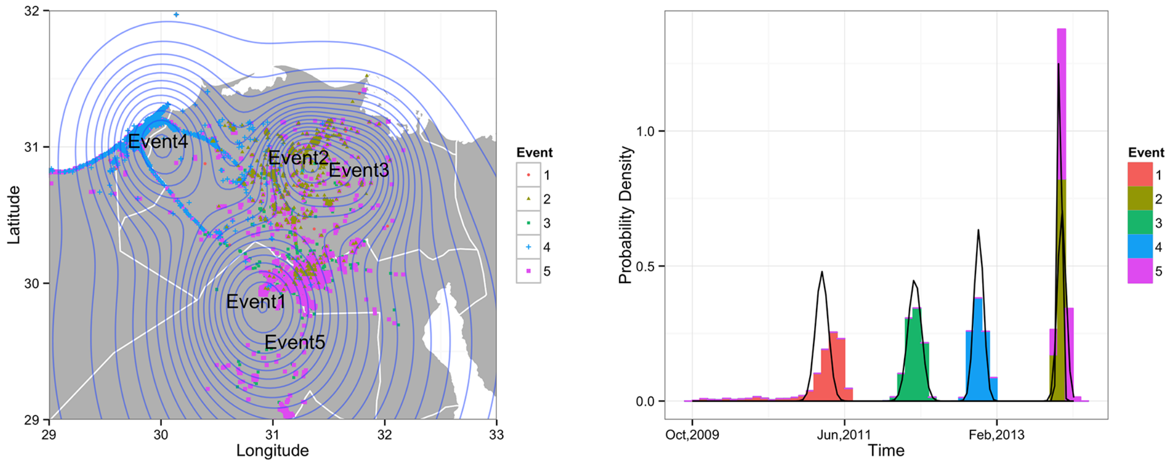
- Pick an event $e \sim \mathrm{Mult}(\gamma)$.
- Generate observed location $l \sim \mathrm{N}(\theta_e^{(L)}, \sigma_e^{(L)})$
- Generate observed time $t \sim \mathrm{N}(\theta_e^{(T)}, \sigma_e^{(T)})$
- Pick a category distribution $\pi \sim \mathrm{Dir}(\alpha)$
- For each word w_i, first pick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

I implemented the event detection algorithm and experimented it on a subset of our Twitter data set that covers the geo region of Egypt with roughly 1.4 million tweets. In this section I will show that the events detected using our algorithm match the information on Wikipedia and official government documents.

4.5 Visualizations of Representative Events

To begin with, I set the number of events in our model to be 100 and selected 5 representative events that spanned different spatial regions and time periods. Those events are summarized in Table 1. The start date and end date of the events are determined by $\theta_e^{(T)} - \sigma_e^{(T)}$ and $\theta_e^{(T)} + \sigma_e^{(T)}$.

The spatial and temporal distributions of those five events are illustrated in Figure 2. In the spatial visualization in Figure 2 (a), each point represents a tweet and a particular event being ascribed to by the color and shape. The figure overlays a contour graph of the spatial distributions of the events described by our graphical model. The contour plot shows three clear geographical clusters that corresponds 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 topical information 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. In the temporal visualization in Figure 2 (b), I see 4 clear clusters with Gaussian peak and centers for each of the events spread out during the time frame of the data set. Two of the event overlap with each other on the right most spike.



(a) Spatial visualizations of events

(b) Temporal visualizations of events

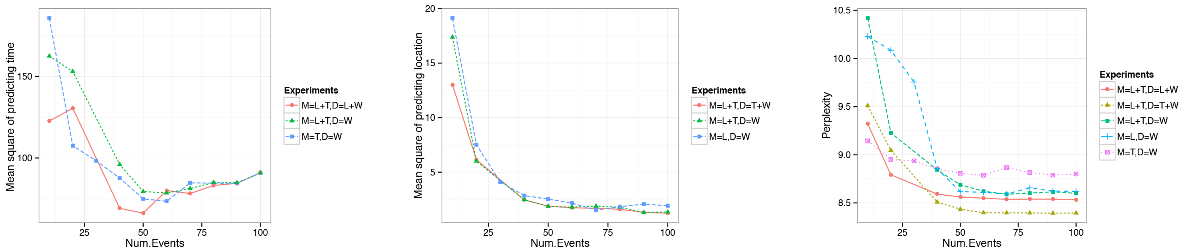
Figure 2: Spatial and temporal visualizations of events

The semantic interpretations of the events will be most clear when I combine the spatial, temporal and topical distributions together. The topical distributions are illustrated in Table 2. In that table I listed some of the top words that have high probability to appear in a event (i.e. words w that have largest $\Phi_E(w)$). Here I first focus on event 1, which has top associated words such as "jan25", "arrested", "Egypt" and "tortured". The spatial distribution of that event suggest it is largely concentrated in Cairo, which is the capital of Egypt. And the temporal distribution of the event are centered on early 2011. The start date and end date of the event is recorded in Table 1 to be Jan,30 and Mar,21. Searching through the web, I found that this event corresponds to the beginning of the Arab Spring demonstration that happened in the Tahrir Square of Cario, Egypt. Wikipedia [1] confirmed the date of the actual event lasts from January 25 to 11 February, which largely overlaps with the detected time range of our model. While I focus here on Event 1, I noticed 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. This event is reported in the official document of Food and Culture Organization of the United Nations [2].

4.6 Numerical Results on Predictions

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, I compare three variants of the model and use each for three different prediction tasks given varying amounts of information about the test data. I 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 testing.

[1] http://en.wikipedia.org/wiki/Egyptian_Revolution_of_2011
[2] http://www.fao.org/news/story/en/item/129919/icode/



(a) Prediction error of time (b) Prediction error of location (c) perplexity

Figure 3: Numerical evaluation of the event model

The first model variant I consider is the full model proposed in Figure 1, marked as **M=L+T**. Second, I use a model with only the location component, ignoring information on time and thus ignoring $\bar{t}$, and $\Phi^{(T)}$. I denote this as **M=L**. Finally, I 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**.In the first task, I use each model and the information given to us in the test data to predict the words in each tweet. I evaluate this by using perplexity. Second, I use each model to predict the time of each tweet in the test data. Finally, I use each model to predict the location of each tweet in the test data.

Experimental results for perplexity are illustrated in Figure 3(C), 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 I 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(c) 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.

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 I will achieve. This is illustrated in Figure 3(a) and Figure 3(b). 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.

5 Proposed Work: Temporal Evolution of Events

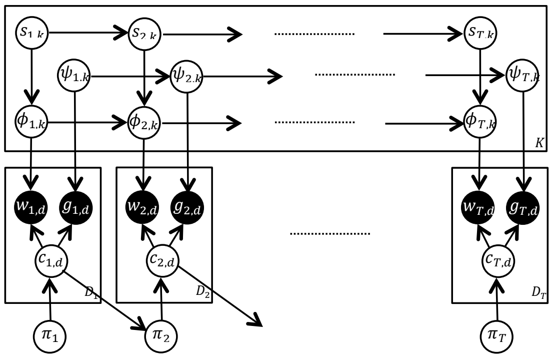


Figure 4: Graphical model of temporal evolution of events

The model presented in Section 4 overlooks the fact that events can evolve with time. That is, although the model includes time, it assumes spatial and textual distributions stay unchanged over time. However, real life scenarios do not support this assumption. For example, the recent Arab Spring revolutions that fundamentally changed government structures in the Arab world [4] can be treated as a series of revolutionary events that are highly related but evolved over time. The event began in Tunisia in late 2010 with government being overthrown. It then quickly spread into other neighboring countries over time such as Egypt, Yemen and Syria. This example motivated us to consider temporally related events as a single series of events.

In general, there are several benefits to treat temporally related events to be a single and unified event with an evolving nature. First, I will be able to see how an event's geographical centers and topical concentrations change over time. This is especially useful for recurring events with as social revolutions and demonstrations. Second, I are able to connect the dots and use data across a much longer time frame. This enables to model to learn topical distribution better than using only a slice of the data. And finally, by adding a sentiment component, I will be able to see the fluctuations of opinions over time toward a single event.

Another drawback of the parametric model proposed in Section 4 is its inability to adjust the dimension of parameter space based on the data. Because of the nature of parametric model, the number of events K has to be pre-fixed and no known methods are effective to determine this value before the actual learning begins. In this research, I utilize a non-parametric technique known as the Recurrent Chinese Restaurant Process (RCRP)[1]. RCRP is a generalization of Dirichlet Process [18] that is capable of accommodating the temporal dynamics of the Dirichlet Process over time. Using the same Chinese Restaurant metaphor as I used before, events are tables and documents are customers. At a specific time t, a customer i with parameter $\theta_{t,i}$ can either choose an existing table k with parameter $\beta_{t,k}$ or create a new table with parameter $\beta_{t,k+1}$ drawn from base distribution G_0 according to Equation 9. Different from Dirichlet Process defined in Equation 1, table parameters (i.e. β_t, k) evolve over time in a Markovian way using Equation 10. Another thing that is different from the DP is that the probability of choosing a specific table k is now proportional to not only the current number of customers at time step t but also the number of customers on the previous time step $t-1$. Note that a table (i.e. event) can die if no documents are attached to it on a specific time period (i.e. $n_t, k = 0$). This is because on the next time step, the probability of choosing this table is precisely 0 and will continue to be 0 after that.

$$\theta_{t,i}|\{\theta_{t-1,\cdot}\}, \theta_{t,1:i-1}, G_0, \alpha \sim \frac{1}{N_{t-1}+i+\alpha-1} \times \left[\sum_{k \in I_{t-1} \cup I_t^{(i)}} (n_{t-1,k} + n_{t,k}^{(i)} \delta(\beta_{t,k}) + \alpha G_0 \right] \quad (9)$$

$$\beta_{t,k} \sim P(\cdot|\beta_{t-1,k}) \quad (10)$$

The graphical model is illustrated in Figure 4. Here, events form a Markov chain with K repetitions. Similar to the basic event model in Section 4, at each time t, an event k has a topical distribution $\phi_{t,k}$, a spatial distribution $\psi_{t,k}$ and a sentiment label $S_{t,k}$. The topical distribution is generated by a sentiment label $s_{t,k}$. The reason to add this sentiment label is because I want to see how opinions about a certain event change over time. For the purpose of clarity, I ignored all the hyper-parameters in the Figure. The event parameter will now include both spatial, textual distributions and the sentiment label, i.e. $\beta_{k,t} = \{\phi_{t,k}, \psi_{t,k}, S_{t,k}\}$. Both $s_{t,k}$, $\phi_{t,k}$ and $\psi_{t,k}$ will change over time according to according to Equation 10 by applying the notation $\beta_{t,k} = \{s_{t,k}, \phi_{t,k}, \psi_{t,k}\}$. Here, the proposal distribution $P(\cdot)$ can be for example a Gaussian distribution with certain variance and a mean centered at $\beta_{t-1,k}$. In the document plate, each time step t will has a collection of D_t documents. For a document d at time t, the observed document text $w_{t,d}$ will be generated using a distribution parametrized by $\phi_{t,k}$ while the observed spatial coordinates $g_{t,d}$ will be generated using a distribution parametrized by $\psi_{t,k}$. A variable $c_{t,d}$ determines the event index of the document. Finally, π_t is the prior probability of the event index that is determined by RCRP in Equation 9.

6 Proposed Work: Predicting the Life Cycles of New Born Events

The event representations from the models proposed in Section 4 and Section 5 can only reflect the knowledge of the events that I currently observed. Although this is a reasonable assumption for an event detection model, it is not useful in some situations where foresight about events are necessary. Consider, for example, a marketing team that wants to start a new product sales campaign. Decision makers on the team might need to know what kinds of campaign or events will allow them to trigger longer impact. Both the topical concentration of the event and the spatial characteristics will affect the outcome of event life cycles and the research question here is how I can learn this pattern. After I have learned this pattern, I can predict the event life cycles based on some initial social media posts that are attached to this event.

In order to achieve the prediction of event cycles, I use a continuous treatment for time, which is different from the discrete assumption in Section sec:temporal. The benefit of using a continuous temporal model is clear: the event life start and end time does not need to align with the discretized temporal boundaries determined by epochs and more realistic and flexible temporal cycles can be learned. Different from the Gaussian assumption in the basic event model, here I model the arrival of events in using Hawkes Process[21]. A Hawkes process is essentially an inhomogeneous Poisson process $Poisson(\lambda(t))$, with $\lambda(t)$ defined in Equation 11. The intensity function $\lambda(t)$ depends on the sum of γ_0, which is the intensity of a homogeneous background Poisson Process and the accumulation of triggering kernel $\gamma(t, t_i)$ over each previously seen document i.

$$\lambda(t) = \gamma_0 + \sum_{t_i \in \tau} \gamma(t, t_i) \tag{11}$$

A treatment of non-parametric clustering that is similar to the one found in Dirichlet Process is achieved by applying Dirichlet-Hawkes Process (DHP) [15]. Similar to Dirichlet Process, each cluster k in Dirichlet-Hawkes Process has parameter β_k drawn from the base distribution G_0. Each cluster k has its own triggering kernel $\gamma_{\beta_k}(t, t_j)$ along with its separate Hawkes Process $\sum_{t_j \in \mathcal{T}} \gamma_{\beta_k}(t, t_j)$. A document j that belongs to one of the clusters and will have a event parameter θ_j which value is taken from one of the event parameter β_k that it belongs to. Using that notation, I can alternatively represent the intensity function of Hawkes process of cluster k (I let $\gamma_0 = 0$) up to time i, $\lambda_k^i(t)$ to be:

$$\lambda_k^i(t) = \sum_{j=1}^{i} \gamma_{\theta_j}(t, t_j)\delta(\theta_j = \beta_k) \tag{12}$$

The Dirichlet Hawkes Process of the event model is defined in Equation 13. Note that the form of Equation 13 is very similar to that of Dirichlet process defined in Equation 1 in that here γ_0 serves the same purpose of dispersion parameter α in Equation 1 and the count n_k^i is now replaced by $\lambda_k^i(t)$. To understand the relationship between standard Hawkes Process and Dirichlet-Hawkes Process, remember the splitting property of Poisson process where a unified Poisson Process with intensity function $\gamma_0 + \sum_k \lambda_k^i(t)$ can be forked to $k+1$ independent Poisson Process with intensity function $\frac{\gamma_0}{\gamma_0 + \sum_k \lambda_k^i(t)}$ and $\frac{\lambda_k^i(t)}{\gamma_0 + \sum_k \lambda_k^i(t)}$. The former one will correspond to the case where a new cluster is generated from G_0 while the later one will choose an existing cluster parameter β_k.

$$\theta_i | \{\theta_{1:i-1}\}, G_0 \sim \frac{1}{\gamma_0 + \lambda_k^i(t)} \times \left[\sum_k \lambda_k^i(t)\delta(\beta_k) + \gamma_0 G_0 \right] \tag{13}$$

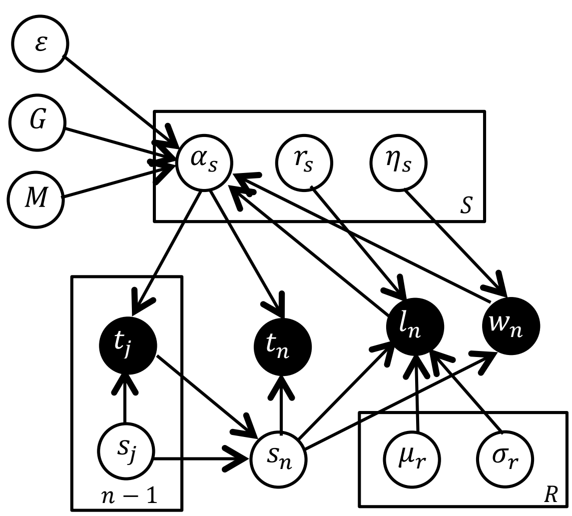


Figure 5: Graphical notation of the proposed model

The graphical representation of our proposed model is illustrated in Figure 5. Here the cluster index s_n for the n^{th} document is generated by following the Dirichlet-Hawkes process. Based on DHP, the value of s_n is dependent on the previous values of cluster indices $s_i, s_{i-1}, ..., s_2, s_1$ as well as the time of the previous documents $t_i, t_{i-1}, ..., t_2, t_1$. After that, time t_n is generated according to a Poisson Poisson with intensity function in Equation 12. Different from the original DHP model, here I expand spatial data into the model by introducing a region cluster index r_s in each cluster s. Each region cluster r_s corresponds to a two-dimensional Gaussian distribution with mean and variance of (μ_r, σ_r). Different cluster can share the same cluster region index and there are a total of R regions. Location l_n is then generated based on this Gaussian distribution determined by r_s. Word vector w_n for each document n is generated by Dirichlet distribution parametrized by η_s.

As in the original DHP model, I define the triggering kernel of each event cluster to be a weighted combination of L basic radial based functions (RBF) $\kappa(\tau_l, \Delta)$ with parameter τ_l. Those set of L kernel functions are weighted by vector α_s defined in each cluster. One example of kernel function is Gaussian RBF and $\kappa(\tau_l, \Delta) = exp(-(\Delta - \tau_l)^2/2\pi_l^2)\sqrt{2\pi\sigma_l^2}$. In this case, I can pre-select some common values of τ_l such as 1,12,24,48,96 hours and let the model to learn a $\alpha(W_{\theta_k}, L_{\theta_k})$ in the sense that each event's life cycle is a combination of basic cycles weighted by a function that is related to the text and location of the documents.

$$\gamma_{\theta_j}(t, t_j) = \alpha \sum_{l}^{L} \kappa(\tau_l, t - t_j) \tag{14}$$

In order to predict the event life cycles based on text and spatial data, I define kernel weight α_s in Equation 15 to be a function of text w_n that belong to cluster s as well as the index of the spatial Gaussian center r_s. Here, N_s is the total number of documents that belong to cluster s. I average the word vector of each such document and multiply it by a matrix M. This will generate a vector that represents the strength of the L kernel functions on this particular cluster. On the other hand, G is a matrix that contains the strength vector of kernel by each spatial region. The weight vector from both text and spatial data is reweighted by a hyper-parameter ϵ.

$$\alpha_s = \epsilon M \sum_{n=1} w_n \delta(\theta_n = \beta_s)/N_s + (1 - \epsilon) G_{r_s} \tag{15}$$

The primary goal of the algorithm is to learn matrix M and G, which are temporal and spatial patterns that affect the event life cycles. After this pattern is learned, we can predict the life cycle of this emerging events by looking at the text and location of the several early documents and try to match that with historical patterns in matrix M and G. For example, certain keywords might trigger long event life cycles and will generate a weight vector that in favor of those kernels with longer life cycles. Similar patterns should be able to be learned from locations where the geo-regions determined the popularities and hence the life cycles of new born events.

7 Proposed Work: Learning Events Using Social Media and Newspaper Data

Although events can be learned through social media posts with spatial and temporal meta data, certain characteristic of such kinds of data can affect quality of the learning. First, social media posts are usually limited to a certain number of words. Take Twitter for example, the text of each tweet is limited to only 140 characters, which makes it difficult for the learning algorithm to differentiate between different events in some situation. Second, the fact the social media posts are usually composed by non-professional writers makes it prone to contain typos, slangs and abbreviations. Additional efforts are needed in order to deal with those language characteristics. And finally, contents on social media posts might not be completely focused on events. Although the geo-coded tweets should contain a significant proportion of data that are focused on events, it is likely that some of them are talking about topics that are deviated from the events characterize by their temporal and spatial meta data.

Data Source	Social Media	Newspaper
Geo-coded	Yes	No
Quality of text	Mixed	Good
Realtime	Yes	No
Focus on events	Usually	Yes

Figure 6: Comparison between social media and newspaper in terms of event learning

In this section, I propose a joint event learning model by utilizing the information on both social media and newspaper data. A comparison between the two data sources in terms of event learning can be found in Table 7. Here we can see that the two types of data have benefits that can complement each other. Newspaper data has the drawbacks of lacking explicit spatial coordinates and a possible delay. However, the benefits of using newspaper data is also obvious: it delivers a much higher quality of text and it is usually highly focused on a specific event that it is reporting, both of which can be used to make up the poor text quality and the sometime irrelevance nature of tweets.

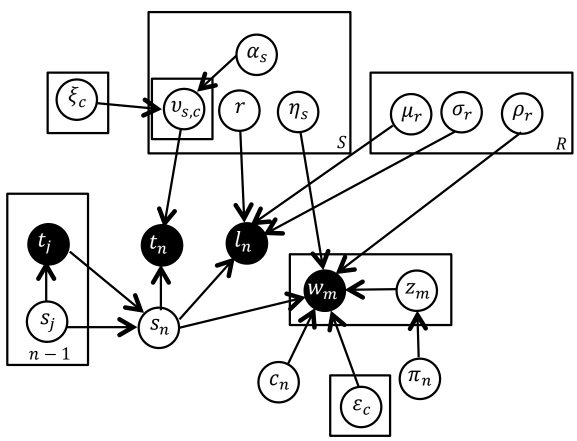


Figure 7: Graphical notation of the proposed model

The graphical model I propose is depicted in Figure 7. The basic idea here is to maintain a joint event representation that is shared across two different data sets. Again I utilize a modified Dirichlet-Hawkes process to model events. Each event here is represented by 4 parameter: the weights on the temporal kernel functions, α_s, a localized version of it based on data set $\upsilon_{s,c}$, an event region indicator r and a list of event related words η_s. After the event index s is generated according to the Dirichlet-Hawkes Process, the document location l_n will be generated by drawing a Gaussian distributed variable from the event spatial region specified by region index r.

$$l_n \sim \mathcal{N}(\mu_r, \sigma_r^2) \tag{16}$$

However, not all the data will have an location in the generated document. In tweets, I assume locations are known. In newspaper, I still assume a location variable and the dependency structures but treat it as location latent variable since it's not present in the actual data.

The dependency structures of time and text will reflect not only the characteristics of a joint event model but also the characteristics of the corresponding data. For each word in the document w_m, a category variable z_m will first be drawn from a multinomial distribution determined by parameter π_n. If $z_m = 0$, the words will be drawn from a location specific dictionary ρ_r. Otherwise, it will be drawn from the dictionary determined by the event η_s. A data type variable c_n will also affect the text of the document w_n. When $c_n = 0$, the document is assumed to be a tweet while $c_n = 1$ making the document to be a newspaper. The word distribution will be localized by a data source specific language parameter ϵ_c in Equation 17.

$$w_m \sim P(w_m|\gamma\eta_s + (1-\gamma)\epsilon_c)\delta(z_m = 0) + P(w_m|\gamma\rho_r + (1-\gamma)\epsilon_c)\delta(z_m = 1) \tag{17}$$

The use of data specific language variance ϵ_c makes the model to be able to differentiate the differences in use of language when switching from one data source to the other. This helps to learn a better unified event distribution as well as a data specific language difference. For example, I might expect that Twitter has a higher probability on words that are less formal while newspaper should have probabilities on words that are formal and professional.

Finally, the time stamps t_n will be generated by Dirichlet-Hawkes process but with a slightly different weight vector. Here each event will have a base weight vector α_s. This weight vector is localized based on the data set and generated a localized weight vector $\upsilon_{s,c}$ for each data type c.

$$\upsilon_{s,c} \sim \mathcal{N}(\alpha_s, \epsilon_c) \tag{18}$$

The localized weight vector $\upsilon_{s,c}$ will serve the same purpose as the α_s found in Equation 15 and serve as the kernel weight of the Dirichlet-Hawkes process model. The benefits of instancing the weight vector for each data type is that I am now able to observe the different temporal patterns of the same event and see how they differ from one type of data to the other.

8 Time Line

Dec, 2015 - Mar, 2016	Section 6: Predicting the Life Cycles of New Born Events
Mar ,2016 - Jun, 2016	Section 5: Temporal Evolutions of Events
Jun, 2016 - Sep, 2016	Section 7: Learning Events Using Social Media and Newspaper Data
Sep, 2016 - Oct, 2016	Writing the thesis

9 Conclusions

In this thesis, I proposed the event detection problem, which is a latent clustering problem on spatial, temporal and textual data. The event detection problem can be treated as a generalization of the topic modeling problem where events can be considered as topics that are augmented by location and time. Several different approaches are proposed to learn different aspects of events. The approaches proposed in this thesis are largely based on Bayesian non-parametric methods to deal with steaming data and unpredictable number of data clusters. I believe the research proposed 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.

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. 2.3, 5

[2] Amr Ahmed, Liangjie Hong, and Alexander Smola. Nested chinese restaurant franchise process: Applications to user tracking and document modeling. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1426–1434, 2013. 2.3

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

[4] Lisa Anderson. Demystifying the arab spring. *Foreign Affairs*, 90(3):2–7, 2011. 1, 5

[5] 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

[6] 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

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

[8] 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

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 1, 2.2, 4.3

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

[11] 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

[12] 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

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

[14] 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

[15] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. 2015. 2.3, 6

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

[17] Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. Diffusion of lexical change in social media. 2014. 3

[18] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973. 2.3, 5

[19] 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

[20] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004. 2.2

[21] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971. 6

[22] 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

[23] 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

[24] 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

[25] 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

[26] 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

[27] 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

[28] 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

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

[30] 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

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

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

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

[34] 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

[35] 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

[36] 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

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

[38] 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

[39] 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

[40] David A Weaver and Bruce Bimber. Finding news stories: a comparison of searches using lexisnexis and google news. *Journalism & Mass Communication Quarterly*, 85(3):515–530, 2008. 3

[41] 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. 1, 2.2, 4

[42] 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

[43] 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

[44] 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

[45] 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