Detecting and Characterizing Events

Anonymous EMNLP submission

Abstract

Significant events are characterized by interactions between entities (e.g., countries, organizations, individuals) that deviate from typical interaction patterns. Investigators, such as historians, commonly read large quantities of text to construct an accurate picture of who, what, when, and where and event happened. In this work, we present the *Capsule* model for analyzing documents to identify and characterize events of potential significance. Specifically, we develop a model based on topic modeling to distinguish between topics that describe "business-as-usual" and topics that deviate from these patterns. To demonstrate this model, we analyze real-world datasets, including a corpus of over 2 million US State Department cables from the 1970s; we provide open-source implementations of an inference algorithm for the Capsule model and a visualization of its results.

1 Introduction

Historical events are difficult to define; historians and political scientists read large quantities of text to construct an accurate picture of a single event. Events are interesting by definition: they are the hidden causes of anomalous observations. But they are also inherently abstract—we can observe that changes occur, but we cannot directly observe whether or not an event occurs.

Consider embassies sending diplomatic messages, such as shown in Figure **??**. The Bangkok and Hong Kong embassies have *typical concerns* about which they usually send messages. At date d, however, the message content changes for both embassies—again, we only observe the changes in message content, and do not observe the event directly. Our first goal is to determine *when* events happen, or identify these rare but pervasive deviations from the typical concerns.

Our second goal is to characterize *what* occurs. We rely on topic models (**?**) to summarize documents and use that same latent space to characterize events.

We develop a Bayesian model that discovers the typical concerns of authors, identifies when events occur, and characterizes these events; we call this the *Capsule* model, as it encapsulates events.

Our final goal is to visualize the results of the Capsule model to make them accessible. We provide source code for both Capsule and its associated visualization.

We first review previous research related to event detect, summarization, and visualization. In Section 2, we describe the Capsule model and how to infer the latent parameters (the appendix provides further inference details). Section 3 provides an exploration of results on simulated and three real-world datasets, and we conclude with a discussion in Section 4.

Related work. We first review previous work on automatic event detection and other related concepts.

While Capsule uses text documents and associated metadata as input, event detection is often performed with univariate input data. In this context, bursts that deviate from typical behavior (e.g., noisy constant or a repeating pattern) can define an event (**?**; **?**); Poisson Processes (**?**) are often used to model events under this definition. Alternatively, events can be construed as "change points" to mark when typical

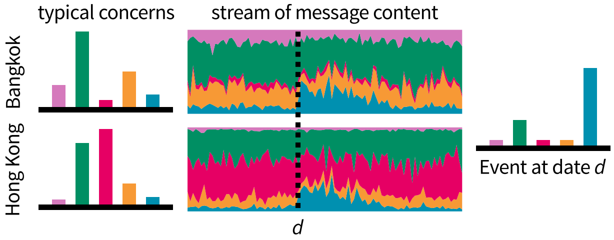

Figure 1: Cartoon intuition of Capsule. Both the Bangkok and Honk Kong embassies have typical concerns about which they usually send messages (represented in topic space). When an events occurs at date d, the stream of message content alters to include the event, then fades back to "business as usual." Capsule discovers both entities' typical concerns and the event locations and content.

observations shift semi-permanently from one value to another (?). In both univariate and multivariate settings, the goal is often the same: analysts want to predict whether or not a rare events will occur (?; ?). Capsule, in contrast, is designed to help analysts explore and understand the original data: our goal is interpretability, not prediction.

Text is often used in event detection, as it is an abundant source of data. In some applications, documents themselves are considered to be observed events (?; ?), or events are predetermined and tracked through the documents (?; ?). We are interested in detecting *unobserved* events which can be characterized by patterns in the data.

A common goal is to identify clusters of documents; these approaches are used on news articles (?; ?; ?; ?; ?; ?) and social media posts (?; ?; ?; ?; ?; ?; ?). In the case of news articles, the task is to create new clusters as novel news stories appear—this does not help disentangle typical content from rare events of interest. Social media approaches identify rare events, but the methods are designed for short, noisy documents; they are not appropriate for larger documents that contain information about a variety of subjects.

Many existing methods use document terms as features, frequently weighted by tf-idf value (?; ?; ?; ?; ?; ?); here, events are bursts in groups of terms. Because language is high dimensional, using terms as features limits scalability.

Topic models (?) reduce the dimensionality of text data; they have been used to help detect events mentioned in social media posts (?; ?) and posts relevant to monitored events (?). We rely on topic models to characterize both typical content and events, but grouped observations can also be summarized directly (?; ?; ?).

In addition to text data over time, author (?), news outlet (?), and spatial information (?; ?; ?) can be used to augment event detection. Capsule uses author information in order to characterize typical concerns of authors.

Detecting and characterizing relationships (?; ?; ?) is related to event detection. When a message recipient is known, Capsule's author input can be replaced with a sender-receiver pair, but the model could be further tailored for interactions within networks.

Once events have been identified and characterized, visualization translates a model's output into sometime intepretable for non experts. LeadLine (?) is an excellent example of a visualization of event detection. We build on topic model visualization concepts (?) to provide tailored visualization code for Capsule.

2 The Capsule Model

In this section we develop the Capsule model. Capsule captures patterns in entity behavior and identifies events that cause deviations from these patterns among many entities. The model relies on rich entity behavior data over time, such as messages being sent between entities; text data can summarized (making the model more tractable) with a topic model (?). We first review topic models at a high level and give the intuition on Capsule. Then, we formally specify our model and discuss how we learn the hidden variables.

Background: Topic Models. Capsule relies on topic models to summarize text data, making the model tractable. Topic models are algorithms for discovering the main themes in a large collection of documents; each document can then be summarized in terms of the global themes. More formally, a topic k is a probability distribution over the set of vocabulary words. Each document d is represented as a distribution over topics θ_d. Thus we can imagine that when we generate a document, we first pick which topics are relevant (and in what proportions); then, for each word, we select a single topic from this distribution over topics, and finally select a vocabulary term from the corresponding topic's distribution over the vocabulary. We use the LDA topic model (**?**; **?**) to summarize text data, and assume that these summaries are held fixed. Our model could be extended to include topic modeling as component, but in practice the results would be similar to the stage-wise approach.

The Capsule Model. Topic models are often applied to provide a structure for an otherwise unstructured collection of documents. Documents, however, are often accompanied by metadata, such as the date written or author attribution; this information is not exploited by traditional topic models. The Capsule model uses both author and date information to identify and characterize events that influence the content of the collection.

Consider an entity like the Bangkok American embassy, shown in Figure **??**. We can imagine that there is a stream of messages (or *diplomatic cables*) being sent by this embassy—some might be sent to the US State Department, others to another American embassy like Hong Kong. An entity will usually talk about certain topics; the Bangkok embassy, for instance, is concerned with topics regarding southeast Asia more generally.

Now imagine that at a particular time, an event occurs, such as the capture of Saigon during the Vietnam war. We do not directly observe that events occur, but each event can again be described in the same topic space used to describe individual messages. Further, when an event occurs, the message content changes for multiple entities. The day following the capture of Saigon, the majority of the diplomatic cables sent by the Bangkok embassy were about Vietnam war refugees. Thus we imagine that an entity's stream of messages is controlled by what it usually talks about as well as the higher level stream of unobserved events.

Model Specification. We formally describe Capsule. The observed data are documents represented in topic space; each document has an author (or entity) and a time (or date) associated with it. The document content for each document d is represented as θ_d, a K-dimensional vector, where K is the number of topics. The author and time associated with document d are represented as n_d and m_d, respectively.

The hidden variables of this model are the authors' typical concerns, event occurrences, and event descriptions. We represent the concerns of author n as ϕ_n, also a K-dimensional topic vector. For each time t we represent whether or not an event occurs with ϵ_t; when an event does occur we represent its content as π_t, another K-dimensional topic vector.

Conditional on the hidden variables and the author and time metadata, Capsule is a model of how document topics θ_d came to be; we generate the topics for each document

$$\theta_{d,k} \sim \text{Gamma}\left(\phi_{n_d,k} + \sum_{t=1}^{T} f(t, m_d)\epsilon_t \pi_{t,k}\right), ^{1} \quad (1)$$

where we define the event decay function f to be a simple linear decrease:

$$f(t,m) = \begin{cases} 1 - \frac{m-t}{\delta}, & \text{if } t \le m < t + \delta \\ 0, & \text{otherwise,} \end{cases}$$

with δ being the time duration after the event at time t is no longer relevant. Figure **??** shows the dependencies between the hidden and observed variables as a graphical model.

To complete the specification of all the variables, we place priors on all of the hidden variables. Author concerns $\phi_{n,k}$ and event content π_t are specified with Gamma priors. Event occurrence ϵ_t has a Poisson prior.

Learning the hidden variables. In order to use the Capsule model to explore the observed documents, we must compute the posterior distribution. Conditional on the observed document topics θ, our goals to to compute the posterior values of the hidden parameters—event occurrences ϵ and descriptions π, as well as entity concerns ϕ.

[1] Throughout this work, we use a non-traditional parameterization of the gamma distribution. Recall the shape a and scale b parameterization of the gamma, or

$$\text{Gamma}^*(xa, b) = \frac{1}{\Gamma(a)b^a} x^{(a-1)} e^{-x/b}.$$

In our alternative parameterization, we use a single mean parameter μ and a fixed sparsity hyperparameter α, or

$$\text{Gamma}(x\mu, \alpha) = \text{Gamma}^*(x\alpha, \mu/\alpha);$$

when a gamma distribution is only specified by a single parameter, it is the mean μ and the sparsity hyperparamter α is hidden for simplicity.

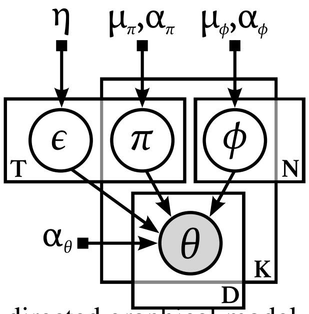

Figure 2: A directed graphical model of Capsule to show considered dependencies. Shaded document topics nodes θ are observed. unshaded nodes are hidden variables—these are event occurrences ϵ, event content descriptions π, and entity typical concerns ϕ. Plates denote replication: there are D documents, T time steps, N entities, and K topics. Hyperparameters η, μ, and α are fixed.

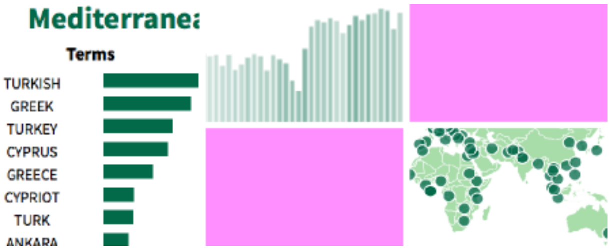

Figure 3: Screenshots of Capsule visualization of US State Department cables. Left: top words in a topic (manually labeled topic title). Center-top: events over time (height is volume of messages sent, color is probability of an event occurring). Center-bottom: topics for an event on <date TODO: cyprus coup?>. Right-top: cyprus entity topics? TODO. Right-bottom: entities shown on a map.

As is common for Bayesian models, the exact posterior for Capsule is not tractable to compute; approximating it is our central statistical and computational problem. We develop an approximate inference algorithm for Capsule based on variational methods (**?**).[2]

Variational inference approaches the problem of posterior inference by minimizing the KL divergence from an approximating distribution q to the true posterior p. This is equivalent to maximizing the ELBO:

$$(q) = _{q(\epsilon,\pi,\phi)}[\log p(\theta,\epsilon,\pi,\phi) - \log q(\epsilon,\pi,\phi)]. \tag{2}$$

We define the approximating distribution q using the mean field assumption:

$$q(\epsilon,\pi,\phi) = \prod_{t=1}^{T} q(\epsilon_t \lambda_t^{\epsilon}) \prod_{k=1}^{K} \left[\prod_{n=1}^{N} q(\phi_{n,k} \lambda_{n,k}^{\phi}) \prod_{t=1}^{T} q(\pi_{t,k} \lambda_{t,k}^{\pi}) \right]. \tag{3}$$

The variational distributions $q(\pi)$ and $q(\phi)$ are both gamma-distributed with free variational parameters λ^{π} and λ^{ϕ}, respectively. The variational distribution $q(\epsilon)$ is Poisson-distributed with variational parameter λ^{ϵ}.

The expectations under q, which are needed to maximize the ELBO, do not have a simple analytic form, so we use "black box" variational inference techniques (**?**). Black box techniques optimize the ELBO directly with stochastic optimization (**?**). Full details on our inference algorithm can be found in the appendix. This algorithm produces a fitted variational distribution which can then be used as a proxy for the true posterior, allowing us to explore a collection of documents with Capsule.

Visualization. Capsule is a high-level statistical tool. In order to understand and explore its results, a user must scrutinize numerical distributions. To make Capsule more accessible, we developed an open source tool for visualizing its results.[3] Our tool creates a navigator of the documents and latent parameters, allowing users to explore events, entities, topics, and the original documents. Figure **??** shows several screenshots of this browsing interface.

3 Evaluation

In this section we study the performance of Capsule. Using simulated data, we compare Capsule to deterministic methods of event detection and show that Capsule outperforms them at identifying when events occur. We conclude by exploring three real-worlds datasets with Capsule.

3.1 Performance

We generated ten simulated datasets using our generative process. Each dataset spans 100 days and contains content associated with ten entities. Approximately ten events also exist in each dataset, ran-

[2] Source code available at https://github.com/ajbc/capsule. TODO: release on github repo (this link is not active.) submission is not anonymous, so why not?

[3] Source code: https://github.com/ajbc/capsule-viz. TODO

domly distributed in time and with a three day decay of relevancy.

To evaluate performance, we rank each day by its probability of having an event occur, and plot the number of true events discovered against the number of false positive events, as shown in Figure 4; the area under the curve (AUC) can be computed for a single evaluation metric. Note that this approach is only valid when true events are known, and thus we only apply it to simulated data.

We compare Capsule to two baseline approaches: one considers the greatest document outlier on a given day–days with the furthest outliers are the most likely to have events. The other approach is similar: days are represented by an average of all documents associated with that day, and one considers how these averages deviate from the global average–the further away, the more likely an event.

Figure 4 shows that Capsule outperforms both of these approaches. It should be noted that inference on Capsule will produce different results, depending on the random seed; the results shown are the best of three random seeds.

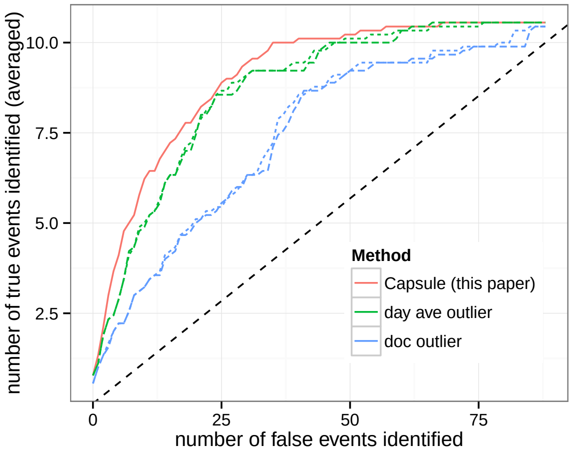

Figure 4: Average performance on ten simulated datasets; lines closer to the upper-left are better. Baselines consider outliers based on full corpus averages (dashed) and averages of all entity documents (dotted). Capsule performance is best of three random seeds.

3.2 Exploration

Cables ¶ where did we get it / size / preprocessing

¶ plot of events timeline with select real-world match evetns pointed out (verified by history lab)

¶ example interesting entities + figure

¶ explore pairwise entities? (quick with and single figure shared with enron); compare sender vs reiever for same pair (or does direction matter?? tyr both ways) look at sender in norma model vs sender in a few pairs under this construction

arXiv ¶ where did we get it / size / preprocessing

¶ plot of events timeline with select real-world match evetns pointed out (verified by history lab)

¶ example interesting entities + figure

enron ¶ where did we get it / size / preprocessing

¶ plot of events timeline with select real-world match evetns pointed out (verified by history lab)

¶ example interesting entities + figure

¶ explore pairwise entities?

4 Discussion

We have presented Capsule, a Bayesian model that identifies when events occur, characterizes these events, and discovers the typical concerns of author entities. We have shown that Capsule outperforms deterministic baseline methods and explored its results on three real-world datasets. We anticipate that Capsule and its visualization can be used by historians, political scientist, and others who wish to explore and investigate events in large text corpora. Future work includes expanding the model to incorporate messages recipients and allowing events to impact only a subset of entities.

A Inference

In this appendix, we describe the details of the variational inference algorithm for Capsule. This algorithm fits the parameters of the variational distribution q in Eq. 3 so that it is close in KL divergence to the posterior.

Recall that the variational distributions $q(\pi)$ and $q(\phi)$ are both gamma-distributed with free variational parameters λ^{π} and λ^{ϕ}, respectively. Each parameter λ has two components: sparsity α and mean μ, which parameterize a shape-rate gamma as $\text{Gamma}(\alpha, \mu/\alpha)$, as noted previously. Because these parameters are free, we use the softplus function $(x) = \log(1+\exp(x))$ to constrain them so that they do not violate the requirements of the gamma distribution. The variational distribution $q(\epsilon)$ is Poisson-distributed with variational parameter λ^{ϵ}, which is also constrained by the softplus function.

Minimizing the KL divergence between the true posterior p and the variational approximation q is equivalent to maximizing the ELBO (Eq. 2). This maximization is often achieved with closed form coordinate updates, but the Capsule model is not specified with the required conjugate relationships that make this approach possible (?). Instead, we rely on "black box" variational inference techniques (?) to perform this optimization.

Black box techniques optimize the ELBO directly with stochastic optimization, which maximizes a function using noisy estimates of its gradient (?). In this case, the function is the ELBO, and we take derivatives with respect to each of the variational parameters. To obtain the noisy estimates, we sample from the variational approximation q; these samples then give us the noisy, unbiased gradients used to update our parameters.

It is essential to employ variance reducing techniques; without them, the algorithm would converge too slowly to be of practical value. Details on each of these techniques may be found in the original black box variational inference paper (?).

One of these techniques is Rao-Blackwellization (?): for each variable, we can write the log probability of all terms containing that variable, giving us

$$\log p_t^{\epsilon} \quad \log p(\epsilon_t \eta_{\epsilon}) + \sum_{d \in D_t} \sum_{k} \log p(\theta_{d,k} \cdots),$$ [4]

$$\log p_{t,k}^{\pi} \quad \log p(\pi_{t,k} \mu_{\pi}, \alpha_{\pi}) + \mathbf{1}_{\epsilon_t} \sum_{d \in D_t} \log p(\theta_{d,k} \cdots),$$ [5]

and

$$\log p_{n,k}^{\phi} \quad \log p(\phi_{n,k} \mu_{\phi}, \alpha_{\phi}) + \sum_{d \in D} \log p(\theta_{d,k} \cdots).$$

Then we can write the gradients with respect to the variational parameters as:

$$\nabla_{\lambda_t^{\epsilon}} = {}_{q}\left[\nabla_{\lambda_t^{\epsilon}} \log q_t^{\epsilon} \left(\log p_t^{\epsilon} - \log q_t^{\epsilon}\right)\right],$$ [6]

$$\nabla_{\lambda_{t,k}^{\pi}} = {}_{q}\left[\nabla_{\lambda_{t,k}^{\pi}} \log q_{t,k}^{\pi} \left(\log p_{t,k}^{\pi} - \log q_{t,k}^{\pi}\right)\right],$$

and

$$\nabla_{\lambda_{n,k}^{\phi}} = {}_{q}\left[\nabla_{\lambda_{n,k}^{\phi}} \log q_{n,k}^{\phi} \left(\log p_{n,k}^{\phi} - \log q_{n,k}^{\phi}\right)\right].$$

Using these gradients, we construct our black box algorithm below in Algorithm 1. As shown, the algorithm does not subsample documents, but for large corpora, we subsample B documents at each iteration and scale the contribution of these samples by D/B.

While not shown explicitly in Algorithm 1, we also use control variates and RMSProp (?) to reduce variance.[7] Additionally, we truncate in two instances: sampled gamma variables are given a lower bound to avoid sampling too close to zero, and free parameters are given both lower and upper bounds—the latter is to avoid overflow.

For Reference The gamma distribution and derivatives:

$$\begin{aligned} \log \text{Gamma}(x\mu, \alpha) =& \alpha \log \alpha - \alpha \log \mu - \log \Gamma(\alpha) \\ & + (\alpha - 1) \log x - \frac{\alpha x}{\mu}, \end{aligned} \tag{4}$$

$$\nabla_{\mu} \log \text{Gamma}(x\mu, \alpha) = -\frac{\alpha}{\mu} + \frac{\alpha x}{\mu^2}, \tag{5}$$

$$\begin{aligned} \nabla_{\alpha} \log \text{Gamma}(x\mu, \alpha) =& \log \alpha + 1 - \log \mu - \Psi(\alpha) \\ & + \log x - \frac{x}{\mu}. \end{aligned} \tag{6}$$

[4] Note that we abbreviate

$$p_t^{\epsilon} = p_t^{\epsilon}(\theta, \epsilon, \pi, \phi)$$

and

$$p(\theta_{d,k} \cdots) = p(\theta_{d,k} \epsilon_t, \pi_{t,k}, \phi_{n_d,k}, \alpha_{\theta}),$$

and define

$$D_t \quad \forall d \in D : f(t, m_d) \neq 0.$$

[5] We use the indicator shorthand:

$$\mathbf{1}_{\epsilon_t} = \begin{cases} 0, & \text{if } \epsilon_t = 0 \\ 1, & \text{otherwise.} \end{cases}$$

[6] We employ yet another abbreviation:

$$q_t^{\epsilon} = q(\epsilon_t \lambda_t^{\epsilon}).$$

[7] In a conversation with Ranganath, he suggested replacing AdaGrad with RMSprop in setting the learning rate.

The Poisson distribution and derivative:

$$\log \text{Poisson}(x \mid \lambda) = x \log \lambda - \log(x!) - \lambda, \tag{7}$$

$$\nabla_\lambda \log \text{Poisson}(x \mid \lambda) = \frac{x}{\lambda} - 1. \tag{8}$$

The softplus function and derivative:

$$(x) = \log(1 + e^x),$$

$$'(x) = \frac{e^x}{1 + e^x}. \tag{9}$$

Note that the derivatives in Equations 5, 6, and 8 will always be used in conjunction with Equation 9, as part of the chain rule:

$$\frac{d}{dx} f((x)) = '(x) f'((x)). \tag{10}$$

Acknowledgments

Do not number the acknowledgment section.

References

James Allan, Ron Papka, and Victor Lavrenko. 1998. Online new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45. ACM.

Hila Becker, Mor Naaman, and Luis Gravano. 2010. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 291–300. ACM.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *JMLR*, 3:993–1022, March.

David M Blei. 2012. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Thorsten Brants, Francine Chen, and Ayman Farahat. 2003. 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.

George Casella and Christian P Robert. 1996. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.

Deepayan Chakrabarti and Kunal Punera. 2011. Event summarization using tweets. *ICWSM*, 11:66–73.

Algorithm 1: Inference for Cables Model

```
Input: document topics $\theta$
Output: estimates of latent parameters event
        occurrences $\epsilon$, event topics $\pi$, and entity
        topics $\phi$
Initialize $\lambda^\epsilon$, $\lambda^\phi$, and $\lambda^\pi$ to respective priors
Initialize iteration count $i = 0$ and $\sigma^\pi = 0$
while change in validation likelihood $< \Delta$ do
    for each sample $s = 1, \ldots, S$ do
        for each entity n and component k do
            draw sample entity topics
            $\phi_{n,k}[s] \sim \text{Gamma}((\lambda^\phi_{n,k}))$
            set $p$, $q$, and $g$ using Equations 4–6, 9,
            and 10:
            $p^\phi_{n,k}[s] = \log p(\phi_{n,k}[s]\mu_\phi, \alpha_\phi)$
            $q^\phi_{n,k}[s] = \log q(\phi_{n,k}[s](\lambda^\phi_{n,k}))$
            $g^\phi_{n,k}[s] = \nabla_{\lambda^\phi_{n,k}} \log q(\phi_{n,k}[s](\lambda^\phi_{n,k}))$
        end
        for each time step t do
            draw sample event occurrence
            $\epsilon_t[s] \sim \text{Poisson}((\lambda^\epsilon_t))$
            set $p$, $q$, and $g$ using Equations 7–10:
            $p^\epsilon_i[s] = \log p(\epsilon_i[s]\eta)$
            $q^\epsilon_i[s] = \log q(\epsilon_i[s](\lambda^\epsilon_i))$
            $g^\pi_i[s] = \nabla_{\lambda^\epsilon_i} \log q(\epsilon_i[s](\lambda^\epsilon_i))$
            if $\epsilon_i[s] \neq 0$ then
                for each component k do
                    draw sample event topics
                    $\pi_{t,k}[s] \sim \text{Gamma}((\lambda^\pi_{t,k}))$
                    set $p$, $q$, and $g$ using
                    Equations 4–6, 9, and 10:
                    $p^\pi_{ik}[s] = \log p(\pi_{ik}[s]\alpha_0, \beta_0)$
                    $q^\pi_{ik}[s] = \log q(\pi_{ik}[s]\lambda^\pi_{ik})$
                    $g^\pi_{ik}[s] =$
                    $\nabla_{\lambda^\pi_{ik}} \log q(\pi_{ik}[s]\lambda^\pi_{ik})$
                end
            end
        end
    end
    for each document d, sample s and component k
    do
        set $\mu_{d,k}[s] =$
        $\phi_{n_d,k}[s] + \sum_t f(t, m_d)\epsilon_t[s]\pi_{t,k}[s]$
        set $p^\theta_{d,k}[s] = \log p(\theta_{d,k}\mu_{d,k}[s], \alpha_\theta)$
        (Eqn. 4)
        $p^\phi_{n,k}[s]p^\theta_{n,k}[s]$
        for each timestep t where $t \leq m_d < t + \delta$
        do
            $p^\epsilon_t[s] \sum_k p^\theta_{d,k}[s]$
            if $\epsilon_t[s] \neq 0$ then
                $p^\pi_{t,k}[s]p^\theta_{t,k}[s]$
                update $\sigma^\pi_t 1$
            end
        end
    end
```

Allison June-Barlow Chaney and David M Blei. 2012. Visualizing topic models. In *ICWSM*.

Kaustav Das, Jeff Schneider, and Daniel B Neill. 2008. Anomaly pattern detection in categorical datasets. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–176. ACM.

Anish Das Sarma, Alpa Jain, and Cong Yu. 2011. Dynamic relationship and event discovery. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 207–216. ACM.

Yann N Dauphin, Harm de Vries, Junyoung Chung, and Yoshua Bengio. 2015. Rmsprop and equilibrated adaptive learning rates for non-convex optimization. *arXiv preprint arXiv:1502.04390*.

Wenwen Dou, Xiaoyu Wang, Drew Skau, William Ribarsky, and Michelle X Zhou. 2012. Leadline: Interactive visual analysis of text data through event identification and exploration. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 93–102. IEEE.

Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S Yu, and Hongjun Lu. 2005. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment.

Wei Gao, Peng Li, and Kareem Darwish. 2012. Joint topic modeling for event summarization across news and social media streams. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1173–1182. ACM.

Zoubin Ghahramani and Matthew J Beal. 2001. Propagation algorithms for variational bayesian learning. *Advances in neural information processing systems*, pages 507–513.

Valery Guralnik and Jaideep Srivastava. 1999. Event detection from time series data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–42. ACM.

Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864.

Alexander Ihler, Jon Hutchins, and Padhraic Smyth. 2007. Learning to detect events with markov-modulated poisson processes. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(3):13.

Alan Jackoway, Hanan Samet, and Jagan Sankaranarayanan. 2011. Identification of live news events using twitter. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pages 25–32. ACM.

J. F. C. Kingman. 1993. *Poisson Processes*.

Jon Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397.

Giridhar Kumaran and James Allan. 2004. 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.

Jey Han Lau, Nigel Collier, and Timothy Baldwin. 2012. On-line trend analysis with topic models:\# twitter trends detection topic model online. In *COLING*, pages 1519–1534.

Zhiwei Li, Bin Wang, Mingjing Li, and Wei-Ying Ma. 2005. A probabilistic model for retrospective news event detection. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 106–113. ACM.

Scott W Linderman and Ryan P Adams. 2014. Discovering latent network structure in point process data. *arXiv preprint arXiv:1402.0914*.

Xueliang Liu, Raphaël Troncy, and Benoit Huet. 2011. Using social media to identify events. In *Proceedings of the 3rd ACM SIGMM international workshop on Social media*, pages 3–8. ACM.

Michael Mathioudakis, Nilesh Bansal, and Nick Koudas. 2010. Identifying, attributing and describing spatial bursts. *Proceedings of the VLDB Endowment*, 3(1-2):1091–1102.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Daniel B Neill, Andrew W Moore, Maheshkumar Sabhnani, and Kenny Daniel. 2005. Detection of emerging space-time clusters. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 218–227. ACM.

Wei Peng, Charles Perng, Tao Li, and Haixun Wang. 2007. Event summarization for system management. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1028–1032. ACM.

Rajesh Ranganath, Sean Gerrish, and David M. Blei. 2014. Black box variational inference. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, AISTATS '14, pages 814–822.

Timo Reuter and Philipp Cimiano. 2012. Event-based classification of social media streams. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, page 22. ACM.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. 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.

Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. 2009. Event detection and tracking in social streams. In *ICWSM*.

Aaron Schein, John Paisley, David M Blei, and Hanna Wallach. 2015. Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1045–1054. ACM.

Courtland VanDam. 2012. A probabilistic topic modeling approach for event detection in social media. Master's thesis, Michigan State University.

Martin J. Wainwright and Michael I. Jordan. 2008. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January.

Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. 2007. Mining correlated bursty topic patterns from coordinated text streams. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 784–793. ACM.

Gary M Weiss and Haym Hirsh. 1998. Learning to predict rare events in event sequences. In *KDD*, pages 359–363.

Yiming Yang, Tom Ault, Thomas Pierce, and Charles W Lattimer. 2000. Improving text categorization methods for event tracking. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 65–72. ACM.

Yi Zhang, Jamie Callan, and Thomas Minka. 2002. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88. ACM.

Qiankun Zhao, Prasenjit Mitra, and Bi Chen. 2007. Temporal and information flow based event detection from social text streams. In *AAAI*, volume 7, pages 1501–1506.

Wayne Xin Zhao, Rishan Chen, Kai Fan, Hongfei Yan, and Xiaoming Li. 2012. A novel burst-based text representation model for scalable event detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 43–47. Association for Computational Linguistics.