Detecting and Characterizing Events

Allison J.B. Chaney
Princeton University
achaney@cs.princeton.edu

Hanna Wallach
Microsoft Research
wallach@microsoft.com

David M. Blei
Columbia University
david.blei@columbia.edu

Matthew Connelly
Columbia University
mjc96@columbia.edu

ABSTRACT

¶ events are interesting (why?)

¶ what are events? How are they usually found?

¶ we find and characterize (how?) events manually

¶ idea of validation results

From Text as data:

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 methods 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, where deviations are also indicated by particular entities interacting during particular periods of time To demonstrate this model, we analyze a corpus of over 2 million State Department cables from 1973 to 1977; we also use this corpus to demonstrate how Capsule (model name) can be used to enhance the exploration of a collection of semi-structured documents.

CCS Concepts

•Information systems → Document collection models; •Human-centered computing → *Visualization toolkits;*

Keywords

ACM proceedings; LaTeX; text tagging

1. INTRODUCTION

¶ why are events interesting?

¶ what is an event? (how do we characterize it)

¶ how can this construction be used? (why do we use the name "Capsule?")

KDD '16 San Francisco, California USA

ACM ISBN ??. . . $15.00

DOI: ?

¶ contributions list (vis, model, code for both, exploration on historical corpus and arxiv/enron)

¶ outline of remainder of paper

Related work.

¶ automatic event detection approaches

¶ topic modeling + viz (incl dynamic topic models)

¶ network and related work there (e.g. hanna's work); say this isnt explicitly about netorks, but the data has this structure dn the model can be extended to use these concepts (do small experiment where "entites" are defined as to/from pairs)

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 [1]. 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 [2, 6] 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,

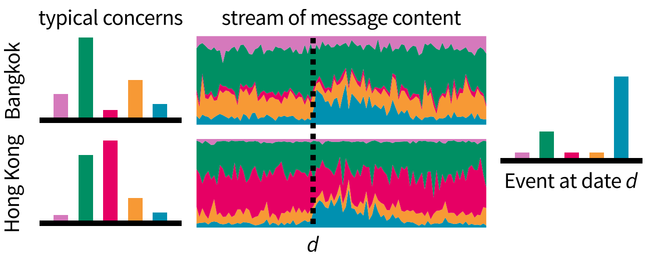

Figure 1: Cartoon intuition of Capsule. Both the Bangkok and Honk Kong embassies have typical concerns about which they usually send messages (represented in topic space). When an events occurs at date d, the stream of message content alters to include the event, then fades back to "business as usual."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

shown in Figure 2. 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

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

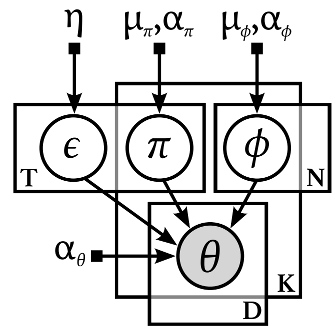

Figure 2: A directed graphical model of Capsule to show considered dependencies. Shaded document topics nodes θ are observed. unshaded nodes are hidden variables—these are event occurrences ϵ, event content descriptions π, and entity typical concerns ϕ. Plates denote replication: there are D documents,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

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

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

with δ being the time duration after the event at time t is no longer relevant. Figure 2 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.

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 [9].[2]

Variational inference approaches the problem of posterior inference by minimizing the KL divergence from an approx-

[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

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

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

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

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.

[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?

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.

imating distribution q to the true posterior p. This is equivalent to maximizing the ELBO:

$$\mathcal{L}(q) = \mathbb{E}_{q(\epsilon,\pi,\phi)}[\log p(\theta,\epsilon,\pi,\phi) - \log q(\epsilon,\pi,\phi)]. \qquad (2)$$

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

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

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 [7]. Black box techniques optimize the ELBO directly with stochastic optimization [8]. 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 2 shows several screenshots of this browsing interface.

3. EVALUATION

¶ outline two datasets (cables and enron) and the tasks to be done

3.1 Data

Cables
arXiv
Enron

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

¶ insert table and refetence for both (number of days, entities, total messages, or something); maybe a plot showing attributes of the data...somehow inform them that the state department is a bias for the cables data

¶ footnote on handling multiple recipients of message...

3.2 Metrics and competing methods

¶ how we evaluate based on real events

¶ how we evaluate based on perplexity (prediction of words)

¶ competing methods for perplexity: LDA, average user words?, dynamic topic model, network topic models

3.3 Performance and exploration

¶ sumry of comparison to gold-standard events for cables

¶ table of predictive likelihood results and summary pgh; and/or cite tea leaves paper

Exploration

¶ charachetrize events manually (based on cables) vs event detection characterization

¶ show descriptions for cables entities and select events; same for arxiv/enron

¶ any other exploration you can think of!

4. DISCUSSION

¶ brief summary of results and contributions

5. ACKNOWLEDGMENTS

TODO

6. REFERENCES

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

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, Mar. 2003.

[3] G. Casella and C. P. Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[4] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengio. Rmsprop and equilibrated adaptive learning rates for non-convex optimization. *arXiv preprint arXiv:1502.04390*, 2015.

[5] Z. Ghahramani and M. J. Beal. Propagation algorithms for variational bayesian learning. *Advances in neural information processing systems*, pages 507–513, 2001.

[6] M. Hoffman, F. R. Bach, and D. M. Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[7] R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, AISTATS '14, pages 814–822, 2014.

[8] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[9] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, Jan. 2008.

APPENDIX

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 $\mathcal{P}(x) = \log(1 + \exp(x))$ 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 [5]. Instead, we rely on "black box" variational inference techniques [7] to perform this optimization.

Black box techniques optimize the ELBO directly with stochastic optimization, which maximizes a function using noisy estimates of its gradient [8]. 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 [7].

One of these techniques is Rao-Blackwellization [3]: for each variable, we can write the log probability of all terms containing that variable, giving us

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

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

and

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

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

$$\nabla_{\lambda^\epsilon_t} \mathcal{L} = \mathbb{E}_q \left[\nabla_{\lambda^\epsilon_t} \log q^\epsilon_t \left(\log p^\epsilon_t - \log q^\epsilon_t \right) \right],\,[6]$$

$$\nabla_{\lambda^\pi_{t,k}} \mathcal{L} = \mathbb{E}_q \left[\nabla_{\lambda^\pi_{t,k}} \log q^\pi_{t,k} \left(\log p^\pi_{t,k} - \log q^\pi_{t,k} \right) \right],$$

and

$$\nabla_{\lambda^\phi_{n,k}} \mathcal{L} = \mathbb{E}_q \left[\nabla_{\lambda^\phi_{n,k}} \log q^\phi_{n,k} \left(\log p^\phi_{n,k} - \log q^\phi_{n,k} \right) \right].$$

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 [4] 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:

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

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

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

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:

$$\mathcal{P}(x) = \log(1 + e^x),$$

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

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:

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

---

[4] Note that we abbreviate

$$p^\epsilon_t = p^\epsilon_t(\theta, \epsilon, \pi, \phi)$$

and

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

and define

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

[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:

$$q^\epsilon_t = q(\epsilon_t \mid \lambda^\epsilon_t).$$

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

Algorithm 1: Inference for Cables Model

Input: document topics θ

Output: estimates of latent parameters event occurrences ϵ, event topics π, and entity topics ϕ

Initialize λ^{ϵ}, λ^{ϕ}, and λ^{π} 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
    - $\phi_{n,k}[s] \sim \text{Gamma}(\mathcal{P}(\lambda^{\phi}_{n,k}))$
    - set p, q, and g using Equations 4–6, 9, and 10:
    - $p^{\phi}_{n,k}[s] = \log p(\phi_{n,k}[s] \mid \mu_{\phi}, \alpha_{\phi})$
    - $q^{\phi}_{n,k}[s] = \log q(\phi_{n,k}[s] \mid \mathcal{P}(\lambda^{\phi}_{n,k}))$
    - $g^{\phi}_{n,k}[s] = \nabla_{\lambda^{\phi}_{n,k}} \log q(\phi_{n,k}[s] \mid \mathcal{P}(\lambda^{\phi}_{n,k}))$
  - **end**
  - **for** *each time step* t **do**
    - draw sample event occurrence
    - $\epsilon_t[s] \sim \text{Poisson}(\mathcal{P}(\lambda^{\epsilon}_t))$
    - set p, q, and g using Equations 7–10:
    - $p^{\epsilon}_i[s] = \log p(\epsilon_i[s] \mid \eta)$
    - $q^{\epsilon}_i[s] = \log q(\epsilon_i[s] \mid \mathcal{P}(\lambda^{\epsilon}_i))$
    - $g^{\pi}_i[s] = \nabla_{\lambda^{\epsilon}_i} \log q(\epsilon_i[s] \mid \mathcal{P}(\lambda^{\epsilon}_i))$
    - **if** $\epsilon_i[s] \neq 0$ **then**
      - **for** *each component* k **do**
        - draw sample event topics
        - $\pi_{t,k}[s] \sim \text{Gamma}(\mathcal{P}(\lambda^{\pi}_{t,k}))$
        - set p, q, and g using Equations 4–6, 9, and 10:
        - $p^{\pi}_{ik}[s] = \log p(\pi_{ik}[s] \mid \alpha_0, \beta_0)$
        - $q^{\pi}_{ik}[s] = \log q(\pi_{ik}[s] \mid \lambda^{\pi}_{ik})$
        - $g^{\pi}_{ik}[s] = \nabla_{\lambda^{\pi}_{ik}} \log q(\pi_{ik}[s] \mid \lambda^{\pi}_{ik})$
      - **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 $p^{\theta}_{d,k}[s] = \log p(\theta_{d,k} \mid \mu_{d,k}[s], \alpha_{\theta})$ (Eqn. 4)
  - $p^{\phi}_{n,k}[s] \mathrel{+}= p^{\theta}_{n,k}[s]$
  - **for** *each timestep* t *where* $t \leq m_d < t + \delta$ **do**
    - $p^{\epsilon}_t[s] \mathrel{+}= \sum_k p^{\theta}_{d,k}[s]$
    - **if** $\epsilon_t[s] \neq 0$ **then**
      - $p^{\pi}_{t,k}[s] \mathrel{+}= p^{\theta}_{t,k}[s]$
      - update $\sigma^{\pi}_t \mathrel{+}= 1$
    - **end**
  - **end**
- **end**
- set $\hat{\nabla}_{\lambda^{\phi}}\mathcal{L} \triangleq \frac{1}{S}\sum_s g^{\phi}[s](p^{\phi}[s] - q^{\phi}[s])$
- set $\hat{\nabla}_{\lambda^{\epsilon}}\mathcal{L} \triangleq \frac{1}{S}\sum_s g^{\epsilon}[s](p^{\epsilon}[s] - q^{\epsilon}[s])$
- set $\hat{\nabla}_{\lambda^{\pi}}\mathcal{L} \triangleq \frac{1}{\sigma_{\pi}}\sum_s g^{\pi}[s](p^{\pi}[s] - q^{\pi}[s])$
- set $\rho = (t + \tau)^{\kappa}$
- set $\lambda^{\pi} \mathrel{+}= \rho\hat{\nabla}_{\lambda^{\pi}}\mathcal{L}$
- set $\lambda^{\epsilon} \mathrel{+}= \rho\hat{\nabla}_{\lambda^{\epsilon}}\mathcal{L}$
- set $\lambda^{\phi} \mathrel{+}= \rho\hat{\nabla}_{\lambda^{\phi}}\mathcal{L}$

end

set $\mathbb{E}[\pi] = \lambda^{\pi,a}$

set $\mathbb{E}[\phi] = \lambda^{\phi,a}$

set $\mathbb{E}[\epsilon] = \lambda^{\epsilon}$

return $\mathbb{E}[\pi]$, $\mathbb{E}[\phi]$, $\mathbb{E}[\epsilon]$