Computational Methods for Exploring Human Behavior

Allison June Barlow Chaney

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: David M. Blei

September 2016

Abstract

Researchers and analysts from many diverse fields are interested in unstructured observations of human behavior; this variety of data is constantly increasing in quantity. In this dissertation, we describe a suite of computational methods to assist investigators in interpreting, organizing, and exploring this data.

We develop two Bayesian latent variable models for human-centered applications; specifically, we rely on additive Poisson models, which allow behavior to be associated with various sources of influence. Given observed data, we estimate the posterior distributions of these models with scalable variational inference algorithms. These models and inference algorithms are validated on real-world data.

Developing statistical models and corresponding inference algorithms only addresses part of the needs of investigators. Non-technical researchers faced with analyzing large quantities of human behavior data are not able to use the results of inference algorithms without tools to translate estimated posterior distributions into accessible visualizations, browsers, or navigators. We present visualization based on an underlying statistical model as a first-class research problem, and provide principles to guide the construction of these systems. We demonstrate these principles with exploratory tools for two latent variable models.

By considering the interplay between developing statistical models and tools for visualization, we are able to develop computational methods that provide for the full needs of investigators interested in exploring human behavior.

Acknowledgments

This dissertation would not have been possible without the help and support of many individuals and organizations.

The research presented in this dissertation was supported in part by Princeton University via a standard first-year fellowship and the Department of Computer Science department via two semesters of teaching assistant funding. The majority of my support was funded with grants secured by my advisor: NSF (IIS-0745520, IIS-1247664, IIS-1009542), ONR (N00014-11-1-0651), and DARPA (FA8750-14-2-0009, N66001-15-C-4032), as well as funding from Adobe, Facebook, Amazon, and the Sloan Foundation, John Templeton Foundation.

I received travel support from the Women in Machine Learning organization, as well as Princeton University's Dean's Fund for Scholarly Travel.

My adviser David Blei has been a more excellent mentor and collaborator than I could have imagined. I thank him for his incomparable personal and professional encouragement and guidance.

I thank the additional members of my PhD committee: Barbara Engelhardt, Brandon Stewart, and Elad Hazan. Barbara has been my academic "aunt," providing me with a second academic home in the face of logistical changes. Brandon has provided excellent feedback on Chapter 3 in an earlier form, acting as a discussant for the Text as Data 2015 Conference. Elad has challenged me to consider more theoretical aspects of the work presented here, for which

I am grateful. All of my committee members have pushed me to take on new research endeavors, which I hope they will continue to do.

I thank my collaborators. Tina Eliassi-Rad worked closely with me on the work presented in Chapter 4, including running the model on the "Social Reader" data. Matthew Connelly and his lab provided access to the cables data and a historian's perspective when developing and evaluating the work presented in Chapter 3. Hanna Wallach was deeply involved in Chapter 3 as well. Her talk at the 2016 ICML Workshop on Human Interpretability in ML talk helped develop Section 2.6.1. She also was a mentor more generally, and facilitated my involvement in the Women in Machine Learning organization.

It has been an honor to work with the members of the Women in Machine Learning (WiML) Board, the WiML community, and my 2014 workshop co-organizers Marzyeh Ghassemi, Sarah Brown, and Jessica Thompson. The connections and collaborations forged through WiML have been very valuable.

Thank you to Jake Hofman and Microsoft Research New York for hosting my 2013 summer internship; many of the skills I learned there have proved useful to the research presented in this dissertation. Thank you to eBay/hunch.com for hosting my 2012 summer internship.

Etsy, and Diane Hu in particular, have been generous in sharing data as well as in inviting me to present ideas, collaborate, and brainstorm research solutions to real-world problems.

Many professors who have taught at Princeton have been influential for me, especially Robert Schapire, whose teaching is an inspiration, and Adam Finkelstein whose enthusiasm for art, design, and media is contagious.

Thank you to Christiane Fellbaum for always being friendly and encouraging.

The PICSciE staff, especially Bill, have been very helpful in terms of computational support. The Computer Science Administrative and Technical staff have also been generous with their time and support, especially Pamela DelOrefice and Melissa Lawson.

I thank many individuals who have worked with me as fellow graduate students and post-docs; their insights, discussions, and friendships have heightened my graduate student experience. Some of these individuals are Rajesh Ranganath, Sean Gerrish, Chong Wang, Prem Gopalan, Maja Rudolph, Jaan Altosaar, Stephan Mandt, Laurent Charlin, Alp Kucukelbir, David Mimno, and members of the Blei Lab and Engelhardt "Beehive" Lab.

I also thank LibRec's creator, Guibing Guo.

I thank my local and non-local friends and family for their encouragement and companionship. Thank you to Lauren Anllo, Katie Wolf, and my local church community.

Thank you to Dad, for teaching me to "never settle for mediocrity when perfection is available." Thank you to Mom, for teaching me to love math, even when it gets the better of you. Thank you to Bill, who will always *really* be Billy, for always making me laugh, usually at myself.

Thank you, Nathaniel, for your enduring support and friendship. I look forward to our future, whatever it may hold.

To Nathaniel.

Contents

1 Introduction

Essentially, all models are wrong, but some are useful.

– George E. P. Box

Human behavior, either at an individual or collective level, is complex. This complexity warrants a myriad of disciplines dedicated to the study of human behavior, each with a unique perspective. Investigators from disparate fields find themselves interested in identical or overlapping data—for example, both economists and socialists analyze consumer purchases; both historians and linguists study written records. And now with the deluge of data emanating from the digital era, investigators find themselves relying on massive unstructured observational data—again, using the same data for different purposes.

Computational methods assist investigators with the analysis of such data. These methods are sufficiently generic to expose patterns in the data that are of interest across disciplines. Computer scientists typically separate these methods for analysis into two areas: statistical modeling for finding patterns in data and building tools for exploring data. While this distinction can prove useful, both areas are intrinsically connected: exploration relies on an underlying model to summarize the data, and modeling relies on exploratory tools to make the inferred results accessible to investigators.

Latent variable models are well-suited to exploratory data analysis because variables can map to intuitive concepts such as the "topic" of a document or the "influence" of one person

on another. These variables represent assumptions about some hidden structure that was involved in the creation of the data—we do not directly observe the "topics" of a document, only the resulting words. Given a joint probability model of latent and observed variables, the central computational task is to compute or estimate the posterior distribution of the latent variables, given the observed data.

The goal of exploratory tools is to translate a posterior distribution into a visualization, browser, or navigator that is accessible to an investigator. Ideally, such tools will display not only summaries of the entire data, but also relevant latent variables alongside the original observations; this allows the model to be a lens through which to view the data.

This dissertation is concerned with the interplay between statistical modeling of human behavior and the exploration of model results. In it, we develop additive Poisson models (a family of latent variable models which are discussed in detail in Section 2.4.1) for two human-centered applications; this model family is particularly convenient for attributing observed behavior to various sources of influence. We also present visualization based on an underlying statistical model as a first-class research problem, and provide principles to guide the construction of these systems. We demonstrate these principles with exploratory tools for two latent variable models.

The remainder of this dissertation is organized as follows.

We begin with preliminary material in Chapter 2. This chapter provides background on latent variables models and posterior inference for these models given observed data. It also contains descriptions for two specific latent variable models on which this work builds—latent Dirichlet allocation (LDA) (Blei et al., 2003) and Poisson Factorization (PF) (Canny, 2004), along with a discussion of their relationship.

In Chapter 3, we turn to modeling text when the authoring entities are being influenced by external events. We present *Capsule*, an additive Poisson model for capturing these events, and demonstrate that our model recovers real-world events and corresponding documents

relevant to these events.

In Chapter 4, we develop *social Poisson Factorization* (SPF), another additive Poisson model to identify social influence. Instead of using text documents as our observations, we now consider logs of user actions online, such as clicking on a product. And instead of external events prompting changes in behavior, we examine the online social network as a source of influence. We demonstrate that this model outperforms competing methods at predicting users' behavior, while also providing an interpretable scaffold with with to explore user preferences.

To make these and other latent variable models accessible to investigators, we present five principles for model development and visualization in Chapter 5, along with guidelines on their application. We first apply these principles by concretely demonstrating how results from the LDA model can be more accessible to investigators with a browsing tool. We show that LDA can be used to organize and navigate an unstructured collection of text documents, making it easier to find documents of interest. We again apply our modeling and visualization principles to demonstrate how the Capsule model for detecting and characterizing events can be used to organize and explore primary source documents.

Finally, we conclude with Chapter 6, in which we summarize the contributions of this dissertation and discuss directions for future work.

2 Preliminary Material

If I have seen a little further it is by standing
on the shoulders of giants.

– Isaac Newton

The ideas of this dissertation relies extensively on existing work, which will be outlined in this chapter. We first discuss latent variables models and common systems to represent them. Second, we describe variational inference, an approach to inferring the posterior of a given latent variable model. Then, we present two specific models—latent Dirichlet allocation (LDA) (Blei et al., 2003) and Poisson Factorization (PF) (Canny, 2004)—and describe their relationship. Finally, we briefly discuss how to develop models and use inferences for exploratory analysis.

2.1 Latent Variable Models

All statistical models are based on some underlying assumptions about how a collection of observed data was generated or is organized. A latent variable model encodes these assumptions with a set of *latent variables* that can have relationships among themselves and with the observations, or *observed variables* (Bishop, 1998). Variables that are "latent" are termed so because they are not observed directly; instead, their structure is assumed to exist

based on expert knowledge of the system, and estimates of their values are discovered via patterns in the observations.

Latent variable models are usually specified in two ways: graphical models and formal generative processes. We describe both varieties of specification.

2.1.1 Graphical Models

Graphical models depict potential dependencies between variables with a directed graph of edges and nodes. Each variable corresponds to a node and each possible dependency is indicated by a directed edge. Further, shaded nodes specify that a variable is observed, while unshaded nodes express that the variable is unobserved. Plates denote replication of variables.

As an example, imagine that we have N observations of some variable x. This variable could take discrete values, such as *heads* or *tails* from a coin flip, or it could take continuous values, like it would if it represented the height of various individuals. This variable is shown in Figure 2.1 as a shaded node within a plate labeled with the number of replicates N.

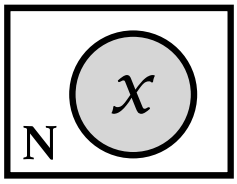


Figure 2.1: A simple graphical model showing N observations of some variable x.

Alone, this variable simply depicts the structure of the data. If we know that there are M coins that are each flipped N times, we could represent this with Figure 2.2. This could also represent the heights of individuals from M villages, each with population N.

To learn something about the data, we need to introduce unobserved variables. Figure 2.3 shows unshaded node z, which indicates one latent variable for each group m (e.g., coin or village). The latent variables $\mathbf{z}$ (indicated in bold when we refer to the entire collection

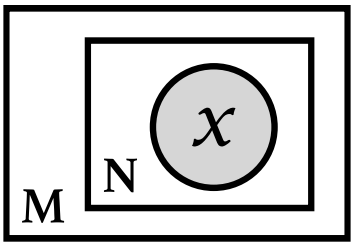


Figure 2.2: A graphical model showing M groups of N observations of variable x.

of these variables) could represent the bias of each coin m or an unobserved condition that impacts height for village m.

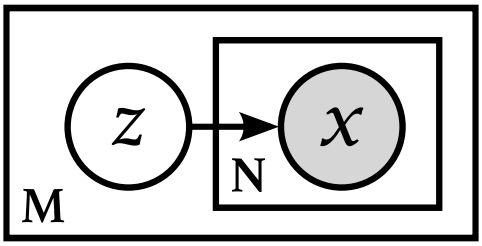


Figure 2.3: A graphical model showing observations of variable x, which may depend on its respective latent variable z.

Some graphical models omit fixed priors, or model hyper-parameters. In this dissertation, we include them as small solid squares; for example, Figure 2.4 shows fixed hyper-parameter μ, which could represent our prior belief about the distribution of coin biases or expert knowledge about the distribution of conditions that impact height.

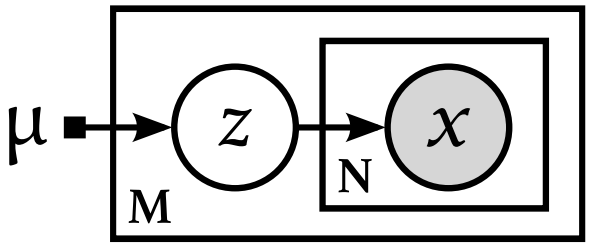


Figure 2.4: A graphical model showing observations of variable x, which may depend on its respective latent variable z, which in turn depends on prior μ.

Here we focus on directed graphical models, also known as "Bayesian networks," but undirected variants of graphical models exist and have their own principles and techniques (Koller and Friedman, 2009).

Graphical model representations are general and intuitive. They provide an easy way of visualizing dependencies between latent and observed variables; in the case of Figure 2.4, observed variables $\mathbf{x}$ can depend on latent variables $\mathbf{z}$. Graphical models specify a family of models with this dependency structure, but to infer the values of the latent variables, we need a formal generative process.

2.1.2 Formal Generative Processes

A formal generative process defines specific probability distributions from which the latent and observed variables are assumed to be generated. Each node, be it latent or observed, must be drawn from some distribution; this distribution can be conditional on other parameters or the fixed hyper-parameters.

Using the coin flip example from the previous section, we can construct a generative process to model this data. If we are to match the graphical model of Figure 2.4, then we need to specify a distribution for each coin bias z, conditional on prior μ; we also need a distribution for each observed coin flip result x, conditional on its corresponding coin bias z. Figure 2.5 defines a formal generative process that fulfills these requirements and matches the graphical model.

- for each coin $m = 1{:}M$,
  - draw coin bias $z_m \sim \text{Beta}(\mu_\alpha, \mu_\beta)$[1]
  - for each flip $n = 1{:}N$,
    - draw side of coin $x_{mn} \sim \text{Bernoulli}\,(z_m)$

Figure 2.5: A generative process for coin flips.

In this example, the coin biases $\mathbf{z}$ are each drawn from a beta distribution, but they could alternatively be drawn from other probability distributions such as the logit-normal distribu-

[1]The beta distribution is specified by two shape parameters, α and β. Thus the hyper-parameter μ is broken into two components: $\mu = (\mu_\alpha, \mu_\beta)$.

tion.

We can similarly specify a formal generative process for the village example, as shown in Figure 2.6. Note that both this generative process and the one shown in Figure 2.5 specify models consistent with the graphical model shown in Figure 2.4; both are members of the general family specified by the graphical model, but each generative process defines a unique model. Both generative processes and graphical models are useful in describing a model: graphical models provide quick intuitions and generative processes define precise models.

- for each village $m = 1{:}M$,
  - draw local condition $z_m \sim \mathcal{N}(\mu)$[2]
  - for each person $n = 1{:}N$,
    - draw height $x_{mn} \sim \mathcal{N}(z_m)$

Figure 2.6: A generative process for heights of villagers.

These specifications define a *joint probability distribution* of the hidden and observed parameters for a given model. Graphical models indicate the dependencies in the joint distribution, and generative processes prescribe their exact mathematical form.

2.1.3 Conditionally Specified Models

Directed probabilistic graphical models must be acyclic in order to guarantee that they define a true joint probability distribution. Similarly, generative processes must not create cycles in their dependencies.

Occasionally, models can only be understood in terms of their conditionally probabilities—the joint distribution may be difficult to investigate directly. In this case, one of two situations occur. The ideal situation is that a well-defined joint distribution exists and the probabilis-

[2]Here we use unit normal distributions for simplicity, but we could extend the model to account for variance.

tic interpretation of the model holds.[3] Alternatively, we have an improper conditionally specified model which may have utility despite its failure to qualify as a genuine joint distribution (Arnold et al., 1999).

In the particular case where the observed data $\mathbf{x}$ is specified by an improper conditional model, the specification defines a pseudo-likelihood (Besag, 1975) instead of a true likelihood. This pseudo-likelihood encapsulates some assumptions about the data and can be used to learn the model parameters for exponential family models (Billiot et al., 2008). Pseudo-likelihoods are typically used to approximate well-defined likelihoods, but if we are using an improper conditional model, we do not know if a genuine likelihood exists to match to our approximation. If the model successfully produces accurate predictions and organizes the original data in an interpretable way, we can posit that there exists and unknown but well-defined likelihood.

Even if the model is useful for prediction and exploring the data, one should avoid making formal causal claims based on inferences under an improper model; its exploratory value is primarily in discovering non-causal associations. As improper models may be easier to develop, they can be used as a precursor to formal causal models.

2.2 Variational Inference

Once an investigator has both data of interest and a formally specified model of how the data was generated, the task is to infer the hidden parameters in the model from the data. This is essentially reversing the generative process to determine the distribution of all the latent variables conditional on the observed data, or the *posterior distribution*.

Using Bayes' law, we can construct the posterior distribution for our running example from

[3] See Arnold et al. (1999) for the conditions under which this occurs.

Figure 2.4:

$$p(\mathbf{z} \mid \mathbf{x}, \mu) = \frac{p(\mathbf{z}, \mathbf{x} \mid \mu)}{\int p(\mathbf{z}, \mathbf{x} \mid \mu) d\mathbf{z}}. \tag{2.1}$$

On the left, we are mathematically representing the posterior distribution: the probability of the latent parameters $\mathbf{z}$ given data $\mathbf{x}$ and hyper-parameters μ. The right-side numerator is the joint distribution of latent parameters $\mathbf{z}$ and observed data $\mathbf{x}$; the joint is easy to evaluate for a single setting of latent parameters $\mathbf{z}$.

The challenge arises from the denominator of Equation (2.1): we want to obtain the joint probability with any given single setting of latent parameters $\mathbf{z}$, relative to the joint under all possible settings of $\mathbf{z}$, which is why we integrate over these values. In simple models, this can be computationally feasible, but for most models of interest to investigators, it is usually not tractable. This means that we cannot evaluate the posterior exactly and our central statistical and computational problem is to approximate the posterior.

Variational inference (Jordan et al., 1999; Wainwright and Jordan, 2008) approximates the posterior p with a family of distributions q. The distributions in q are defined over the latent variables and parameterized with a set of *variational parameters*. For our example model, we write this family as $q(\mathbf{z} \mid \boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ are the variational parameters.

The approximation q is commonly defined using the mean-field assumption, or that the distribution factorizes and each variable is independent:

$$q(\mathbf{z} \mid \boldsymbol{\lambda}) = \prod_{m=1}^{M} q(z_m \mid \lambda_m). \tag{2.2}$$

Each latent variable receives its own free variational parameter λ (or set of parameters). The q family can also be defined to maintain some dependency structure (Han et al., 2013; Hoffman and Blei, 2015).

Given this paradigm, the goal is to find the settings of the variational parameters $\boldsymbol{\lambda}$ that define a distribution in q which is as close as possible to the true posterior p. Closeness is

measured in terms of Kullback-Leibler (KL) divergence (Kullback, 1997; MacKay, 2003), which is an asymmetric measure of distance between distributions:

$$\mathrm{KL}(q||p) = \mathbb{E}_q\left[\log \frac{q(\mathbf{z})}{p(\mathbf{z}\mid\mathbf{x})}\right]. \tag{2.3}$$

Minimizing the KL divergence from an approximating distribution q to the true posterior p cannot be done directly, but it is equivalent to maximizing the evidence lower bound (ELBO) (Hoffman et al., 2013),

$$\mathcal{L}(q) = \mathbb{E}_q\big[\log p(\mathbf{x},\mathbf{z}\mid\mu) - \log q(\mathbf{z}\mid\boldsymbol{\lambda})\big]. \tag{2.4}$$

If we can write out the *complete conditional distribution* for each latent variable, or the probability of a variable given all other latent and observed variables, then we can use coordinate ascent to maximize the ELBO.[4] With coordinate ascent, we update each variable in turn, holding all the others fixed.

In order for a model to have an analytic complete conditional for each variable, the dependency relationships need to be conjugate (Carlin and Polson, 1991; Gelman et al., 2014). Without these relationships, further approximations are needed; Wang and Blei (2013) derived variational algorithms for a wide class of non-conjugate models and Ranganath et al. (2014) developed "black box" variational inference, which can be applied to any model specified with exponential family distributions.

These optimization algorithms return values for each of the variational parameters $\boldsymbol{\lambda}$; these parameters select the one approximate posterior distribution from the full q family. With this distribution, we can compute expected values of the latent parameters, e.g., $\mathbb{E}[z_m]$. These expectations can then be used to explore the model and the original data.

[4] Other optimization approaches, such as gradient ascent, can be used instead of coordinate ascent.

2.2.1 Stochastic Variational Inference

Traditional or *batch* variational inference considers all the data in every iteration; for massive or even streaming data, this is not practical. For these situations, Hoffman et al. (2013) developed stochastic variational inference, which uses samples from the data to estimate the updates for each variable. With stochastic variational inference, parameter updates that involves a function of the data are altered—the original data is replaced with a scaled sample of the data.

There is some finesse in selecting the scope of the sampling. For instance, if the data is naturally grouped, sampling may be most efficient at the group level—when a group is sampled, its local parameters can be updated in a batch manner, and only global parameters shared between groups need to be stochastically updated. In this case, if the local parameters are of interest, then a final pass over all the observations must be done at the end of the algorithm to ensure that the local parameters have values which correspond to the most recent global parameters.

To illustrate this subtlety and provide concrete examples of both batch and stochastic variational inference, we now turn to discussing two popular latent variable models on which the work of this dissertation builds.

2.3 Latent Dirichlet Allocation

Probabilistic topic models discover the hidden thematic structure in large collection of documents; Latent Dirichlet allocation (LDA) (Blei et al., 2003) is the simplest topic model, on which many other models are based (Blei, 2012; Blei and Lafferty, 2009). LDA decomposes a collection of documents into *topics* and represents each document with a weighted subset of the topics.

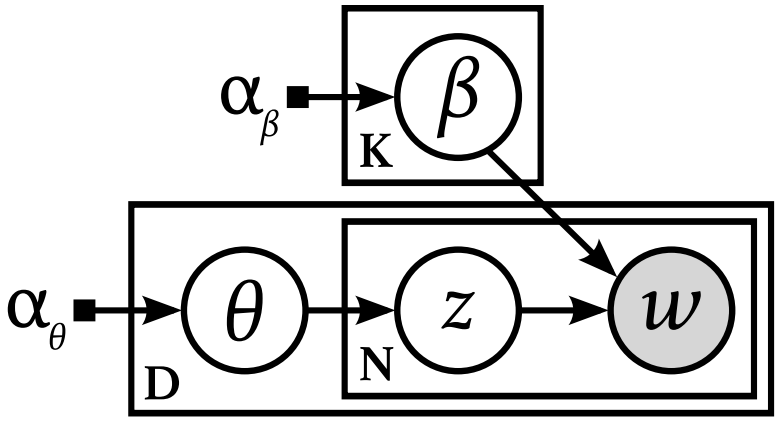


Figure 2.7: The graphical model for latent Dirichlet allocation (LDA). The variables include K topics β, local topical representations θ for each of the D documents, topic assignments z for each of the N words, and observed words w.

More formally, each topic β is a distribution over V vocabulary terms. Each document is represented as a distribution over the K topics: θ_d is a local K-dimensional topic vector for document d. Also at the document level is the latent variable z_{dn}, the topic assignment for the nth word of document d. A word's topic assignments z depend on the topic distributions for its respective document. In turn, the observed word w depends on its topic assignment z and the distribution of terms for the corresponding topic β_z. These dependencies are shown in Figure 2.7, the graphical model for LDA; the full generative process is shown in Figure 2.8. These define the joint distribution of the model:

$$p\left(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w} \mid \alpha_\beta, \alpha_\theta\right) = \prod_{k=1}^{K} p(\beta_k \mid \alpha_\beta) \prod_{d=1}^{D} \left[p(\theta_d \mid \alpha_\theta) \prod_{n=1}^{N_d} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \right]. \quad (2.5)$$

To infer the values of the latent variables ($\boldsymbol{\beta}$, $\boldsymbol{\theta}$, $\mathbf{z}$) of this model with variational inference, we approximate this posterior with a flexible family q. Using the mean field assumption, this family is

$$q\left(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z} \mid \boldsymbol{\lambda}\right) = \prod_{k=1}^{K} q(\beta_k \mid \lambda_k^\beta) \prod_{d=1}^{D} \left[q(\theta_d \mid \lambda_d^\theta) \prod_{n=1}^{N_d} q(z_{dn} \mid \lambda_{dn}^z) \right], \quad (2.6)$$

- for each topic $k = 1{:}K$,
  - draw topic distribution over vocabulary $\beta_k \sim \text{Dirichlet}_V(\alpha_\beta)$
- for each document $d = 1{:}D$,
  - draw local document topics $\theta_d \sim \text{Dirichlet}_K(\alpha_\theta)$
  - for each word $n = 1{:}N_d$,
    - draw word assignment $z_{dn} \sim \text{Categorical}_K(\theta_d)$
    - draw word $w_{dn} \sim \text{Categorical}_V(\beta_{z_{dn}})$

Figure 2.8: The generative process for latent Dirichlet allocation (LDA).

where $q(\beta_k)$ and $q(\theta_d)$ are Dirichlet-distributed, and $q(z_{dn})$ is a categorical distribution.

In addition to this approximation, we need the complete conditional distributions for each of the latent parameters. The joint distribution allows us to derive the following complete conditionals (see Appendix B), with the assumption that z_{dn} is a K-dimensional probability vector and w_{dn} is a V-dimensional indicator vector.

$$\beta_k \mid \boldsymbol{\theta}, \boldsymbol{z}, \mathbf{w}, \alpha_\beta, \alpha_\theta \sim \text{Dirichlet}_V\left(\alpha_\beta + \sum_{d=1}^{D}\sum_{n=1}^{N_d} z_{dn,k} w_{dn}\right) \tag{2.7}$$

$$\theta_d \mid \boldsymbol{\beta}, \boldsymbol{z}, \mathbf{w}, \alpha_\beta, \alpha_\theta \sim \text{Dirichlet}_K\left(\alpha_\theta + \sum_{n=1}^{N_d} z_{dn}\right) \tag{2.8}$$

$$z_{dn} \mid \boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{w}, \alpha_\beta, \alpha_\theta \sim \text{Categorical}_V(\theta_d \beta w_{dn}) \tag{2.9}$$

With the family q and the complete conditionals defined, we are able to iteratively update each parameter, as shown in the full variational inference procedure of Algorithm 1.

To adapt inference for a large quantity of documents, M documents are sampled at each iteration, as shown by Hoffman et al. (2013). Local document-specific parameters $\boldsymbol{\theta}$ and $\mathbf{z}$ are fit as in batch inference, but the variational parameters for global topics $\boldsymbol{\beta}$ are updated

Algorithm 1: Mean Field Variational Inference for LDA

Input: words w
Output: approximate posterior of latent parameters (topics $\boldsymbol{\beta}$, document topics $\boldsymbol{\theta}$, and word assignments $\mathbf{z}$) in terms of variational parameters $\boldsymbol{\lambda}$
Initialize $\mathbb{E}[\beta]$ to slightly random around uniform
Initialize $\mathbb{E}[\theta]$ to uniform
for *iteration* $i = 1 : M$ **do**
  set λ^{β} and λ^{θ} to respective priors
  set λ^{z} to zero
  for *each document* $d = 1 : D$ **do**
    for *each term* $n = 1 : N_d$) **do**
      set k-vector $\lambda^{z}_{dn} \propto \langle \mathbb{E}[\theta_{d,1}]\mathbb{E}[\beta_1], \ldots, \mathbb{E}[\theta_{d,K}]\mathbb{E}[\beta_K] \rangle w_{dn}$
      set $\mathbb{E}[z]_{d,n} = \lambda_{d,n}$
      update $\lambda^{\theta}_{d} \mathrel{+}= \mathbb{E}[z_{dn}]$
      update $\lambda^{\beta} \mathrel{+}= \mathbb{E}[z_{dn}] w_{dn}$
    end
    set $\mathbb{E}[\theta_d] \propto \lambda^{\theta}_{d}$
  end
  for *each topic* $k = 1 : K$ **do**
    set $\mathbb{E}[\beta_k] \propto \lambda^{\beta}_{k}$
  end
end
return $\boldsymbol{\lambda}$

stochastically. The contribution of each document ($\mathbb{E}[z_{dn}]w_{dn}$) is scaled as if the sample is representative of the entire collection, or by D/M, and added to intermediate variational parameters; then, the final variational parameters are a weighted average of the old parameters and the new intermediate parameters.

Researchers have developed alternative algorithms for LDA inference, including Markov chain Monte Carlo sampling (Steyvers and Griffiths, 2006; Newman et al., 2007) and other optimization-based variational approaches (Teh et al., 2006).

We now turn to a related model; this discussion will further illustrate model specification and variational inference, as well as provide the groundwork for models developed in this dissertation.

2.4 Poisson Factorization

Factorization approaches are used in many fields; recommendation systems are popular application (Koren et al., 2009). The setup for factorization is to frame the observed data as a matrix. In recommendation systems, one dimension of this matrix is the users and the other is items—the matrix is then filled with observations of users interacting with items, such as the number of times a user has clicked an item.

While less typical, factorization can also be used for topic modeling. In this context, one dimension of the matrix is documents, and the other is vocabulary terms; each cell contains the number of times a word occurs in a given document. Both recommendation and topic modeling applications are well-suited to non-negative matrix factorization (Lee and Seung, 2000), where both the observed matrix and the factor matrices have non-negative cell values. In this discussion, we will continue to use the language of topic modeling to elucidate the parallels with LDA.

Matrix factorization approaches most commonly assume that the cells of a matrix are Gaussian distributed (Salakhutdinov and Mnih, 2007), but they can also be assumed to be Poisson-distributed (Canny, 2004; Gopalan et al., 2015); the latter representation more accurate captures the structure of positive discrete observations such as word counts.

Poisson factorization (PF) represents the observations as w_{dv}; these are the total number of times that vocabulary word v occurs in document d.[5] The latent variables are K-dimensional representations θ_d for each document d and K-dimensional vocabulary term prevalences β_v. Like LDA, these parameters represent the global topics in terms of words and documents in terms of topics,[6] but unlike LDA, the these representations do not live on the unit simplex.

[5]For recommendation systems, observations are more commonly noted as r_{ui}, the number of interactions that user u has with item i, or the user's explicit or implicit rating of that item.

[6]Recommendation systems can be interpreted as capturing user preferences $\boldsymbol{\theta}$ and item attributes $\boldsymbol{\beta}$.

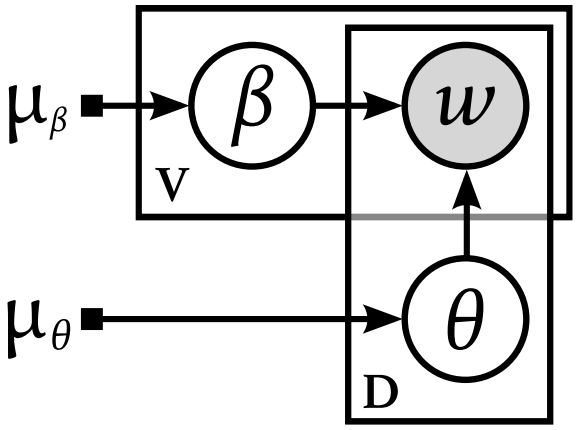


Figure 2.9: The graphical model for Poisson factorization (PF). The variables include β: V vocabulary word prevalences in the K topics, θ: local K-dimensional topic representations for each of the D documents, and observed word counts w. Hyper-parameters μ are composed of shape and rate components: $\mu = (\mu^s, \mu^r)$.

For a given document d and vocabulary term v, the word count w_{dv} depends on both the topical representation of the document θ_d and the term prevalences β_v. These dependencies are shown in Figure 2.9, the graphical model for PF; the full generative process is shown in Figure 2.10. These define the joint distribution of the model:

$$p\left(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{w} \mid \mu_\beta, \mu_\theta\right) = \prod_{k=1}^{K}\left[\prod_{v=1}^{V} p(\beta_{vk} \mid \mu_\beta) \prod_{d=1}^{D} p(\theta_{dk} \mid \mu_\theta)\right] \prod_{d=1}^{D}\prod_{v=1}^{V} p(w_{dv} \mid \theta_{dk}, \beta_{vk}). \quad (2.10)$$

To infer the values of the latent variables $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, we must again define a flexible family q to approximate this posterior; we define this family as

$$q\left(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}\right) = \prod_{k=1}^{K}\left[\prod_{v=1}^{V} q(\beta_{vk} \mid \lambda^{\beta}_{vk}) \prod_{d=1}^{D} p(\theta_{dk} \mid \lambda^{\theta}_{dk})\right], \quad (2.11)$$

where $q(\beta_{vk})$ and $q(\theta_{dk})$ are both gamma-distributed.[7]

[7]Throughout this dissertation, we use the shape and rate parameterization of the gamma distribution, or

$$\text{Gamma}(x \mid s, r) = \frac{r^s}{\Gamma(s)} x^{s-1} e^{-rx}.$$

- for each topic $k = 1{:}K$,
  - for each term topic $v = 1{:}V$,
    - draw topic distribution over vocabulary $\beta_{vk} \sim \text{Gamma}(\mu^s_\beta, \mu^r_\beta)$
- for each document $d = 1{:}D$,
  - for each topic $k = 1{:}K$,
    - draw local document topics $\theta_{dk} \sim \text{Gamma}(\mu^s_\theta, \mu^r_\theta)$
  - for each vocabulary term $v = 1{:}V$,
    - draw word counts $w_{dv} \sim \text{Poisson}\left(\theta_d^\top \beta_v\right)$

Figure 2.10: The generative process for Poisson Factorization (PF).

To obtain simple updates for each parameter, we first employ auxiliary latent variables z. These variables, when marginalized out, leave the original model intact. This construction depends on the additive property of the Poisson distribution. Specifically, if $x \sim \text{Poisson}(a+b)$ then $x = z_1 + z_2$ where $z_1 \sim \text{Poisson}(a)$ and $z_2 \sim \text{Poisson}(b)$. We apply this decomposition to the generative distribution for the word counts w, and define Poisson variables for each topic in the count:

$$w_{dv} = \sum_{k=1}^{K} z_{dvk}, \tag{2.12}$$

where

$$z_{dvk} \sim \text{Poisson}(\theta_{dk}\beta_{vk}). \tag{2.13}$$

The complete conditional distributions of the original latent parameters can then be derived using these auxiliary, or "helper," parameters.

$$, \beta_{vk} \mid \boldsymbol{\theta}, \mathbf{z}, \mathbf{w}, \mu_\beta, \mu_\theta \sim \text{Gamma}\left(\mu^s_\beta + \sum_{d=1}^{D} z_{dvk}, \mu^r_\beta + \sum_{d=1}^{D} \theta_{dk}\right) \tag{2.14}$$

$$\theta_{dk} \mid \boldsymbol{\beta}, \mathbf{z}, \mathbf{w}, \mu_\beta, \mu_\theta \sim \text{Gamma}\left(\mu^s_\theta + \sum_{v=1}^{V} z_{dvk}, \mu^r_\theta + \sum_{v=1}^{V} \beta_{vk}\right) \tag{2.15}$$

Algorithm 2: Mean Field Variational Inference for PF

Input: word counts w
Output: approximate posterior of latent parameters (global topics $\boldsymbol{\beta}$ and document representations $\boldsymbol{\theta}$) in terms of variational parameters $\boldsymbol{\lambda}$
Initialize $\mathbb{E}[\beta]$ to slightly random around uniform
Initialize $\mathbb{E}[\theta]$ to uniform
for *iteration* $i = 1 : M$ **do**
  set λ^{β} and λ^{θ} to respective priors
  update rate paramter $\lambda^{\theta,r}$ $+= \sum_{v=1}^{V} \mathbb{E}[\beta_v]$
  for *each document* $d = 1 : D$ **do**
    for *each vocabulary word* $v \in V(d)$[8] **do**
      set K-vector $\phi_{dv} \propto \langle \mathbb{E}[\theta_{d1}]\mathbb{E}[\beta_{v1}], \ldots, \mathbb{E}[\theta_{dK}]\mathbb{E}[\beta_{vK}] \rangle$
      set K-vector $\mathbb{E}[z_{dv}] = w_{dv} * \phi_{dv}$
      update shape parameter $\lambda_d^{\theta,s}$ $+= \mathbb{E}[z_{dv}]$
      update shape parameter $\lambda_v^{\beta,s}$ $+= \mathbb{E}[z_{dv}]$
    end
    set $\mathbb{E}[\theta_d] = \lambda_d^{\theta,s} / \lambda_d^{z,r}$
    update rate parameter $\lambda^{\beta,r}$ $+= \mathbb{E}[\theta_d]$
  end
  for *each vocabulary word* $v = 1 : V$ **do**
    set $\mathbb{E}[\beta_v] = \lambda_v^{\beta,s} / \lambda_v^{\beta,r}$
  end
end
return λ

We also need to define the conditional distribution for the auxiliary parameters:

$$z_{dv} \mid \boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{w}, \mu_\beta, \mu_\theta \sim \text{Multinomial}\left(w_{dv}, \phi_{dv}\right), \tag{2.16}$$

where

$$\phi_{dv} \propto \langle \theta_{d1}\beta_{v1}, \cdots, \theta_{dK}\beta_{vK} \rangle. \tag{2.17}$$

With this setup, we can iteratively update each parameter to construct the full variational inference procedure, shown in Algorithm 2.

To adapt this algorithm for a large corpora, documents are sampled at each iteration, as done

[8] $V(d)$ is the set of vocabulary indices for the collection of words in document d. We could also iterate over all V, but as zero word counts give $\mathbb{E}[z_{dv}] = 0 \; \forall v \notin V(d)$, the two are equivalent.

for LDA: document-specific parameters z_d and θ_d are updated as in the batch variant, and global topic parameters $\boldsymbol{\beta}$ are updated stochastically.

PF has also been extended to a Bayesian nonparametric version that learns the number of components (Gopalan et al., 2014) and a model that combines multiple signals, such as text and user behavior (Gopalan et al., 2014a).

Gopalan et al. (2015) showed that PF realistically captures patterns of user behavior for recommendation systems, lends itself to scalable algorithms for sparse data, and outperforms traditional matrix factorization based on Gaussian likelihoods (Gopalan et al., 2015; Salakhutdinov and Mnih, 2007).

2.4.1 Additivity

The additive property of Poisson distributed variable was mentioned in the previous section to assist with inference; here we discuss its application to modeling. To reiterate the property:

Property 2.4.1 (Additivity of Poisson random variables) *If* $x \sim Poisson(a + b)$*, then* $x = z_1 + z_2$ *where* $z_1 \sim Poisson(a)$ *and* $z_2 \sim Poisson(b)$.

When multiple factors occur in the parameterization of the Poisson, as in PF, then an observation can be attributed to these various components. For instance, a document with three occurrence of the word *bark* could have one instance attributed to a topic about dogs, and the other two instances attributed to a topic about trees; this attribution is observed via the auxiliary parameters $\mathbf{z}$ used during inference.

Additionally, models can be constructed with multiple terms parameterizing a Poisson distribution. For instance, if a document has author information and we wish to model author words, we can introduce a latent V-dimensional variable α_{a_d}, where a_d indicates the author

of document d. Then, we can extend the PF model to include these author biases:

$$w_{dv} \sim \text{Poisson}(\theta_d^\top \beta_v + \alpha_{a_d v}). \tag{2.18}$$

Now, word counts can be attributed not only to topics, but also to author biases. For example, if the word *bark* is used five times in a single document, we could infer that $\mathbb{E}[z_{dv}] = \langle 0.6, 1.0, 3.4 \rangle$. If these values map to a topic about dogs, a topic about trees, and the author's word inclinations, respectively, then this vector indicates that the author is using the word a little bit because they are discussing dogs and trees, but mostly because they like to use the word regardless of context (perhaps to indicate explosive sounds or brusque orders).

The base model can continue to be extended, adding new latent variables and corresponding terms to parameterize the Poisson. This structure allows additive Poisson models to be applied to problems where interpretability is important, as the model structure includes attribution, which lends itself well to interpretation.

While causal claims are not made in this dissertation, the ability to attribute observations to multiple latent components has the potential to be useful when investigating causal questions.

2.5 The Relationship Between LDA and PF

Latent Dirichlet allocation and Poisson factorization have many parallels (Canny, 2004; Paisley et al., 2014); the type of data they consider is identical, and both uncover latent local and global patterns with similar structure.

The similarities extend to the mathematical distributions used in each model. Specifically, the Dirichlet distribution is equivalent to the distribution of a normalized collection of

gamma variables—in LDA global topics $\boldsymbol{\beta}$ and document representations $\boldsymbol{\theta}$ are assumed to be Dirichlet distributed while in PF, global topics $\boldsymbol{\beta}$ and document representations $\boldsymbol{\theta}$ are assumed to be drawn from unnormalized gamma distributions.[9] Thus one of the main distinctions between the models is that LDA topics and document representations are on the probability simplex, whereas in PF they are not; the simplex representation is more desirable for interpretation.

The models also have subtle distinctions in their assumptions about the total number of words in each document, or document length. In the original LDA specification (Blei et al., 2003), the number of words in a document N_d is drawn from a Poisson distribution—this is typically marginalized out. Poisson factorization, in contrast, does not model the total number of words explicitly and this gamma representation for documents better accommodates documents of various lengths (Canny, 2004).

2.5.1 A Hybrid Model

We can draw on the strengths of each model to construct a hybrid model of both LDA and PF. Global topics $\boldsymbol{\beta}$ are drawn from Dirichlet distributions, allowing them to be true distributions over vocabulary words. Document representations $\boldsymbol{\theta}$ are drawn from gamma distributions, allowing for greater flexibility in document lengths; instead of θ_{dk} representing the proportion of document d about topic k, it represents the expected number of words assigned to topic k in document d. The full generative process of this hybrid approach is shown in Figure 2.11.

[9] See Appendix B for inference derivations that emphasize the mathematical similarity between the two representations.

- for each topic $k = 1{:}K$,
  - draw topic distribution over vocabulary $\beta_{\cdot,k} \sim \text{Dirichlet}_V(\alpha)$
- for each document $d = 1{:}D$,
  - for each topic $k = 1{:}K$,
    - draw local document topics $\theta_{dk} \sim \text{Gamma}(\mu^s, \mu^r)$
  - for each vocabulary term $v = 1{:}V$,
    - draw word counts $w_{dv} \sim \text{Poisson}\left(\theta_d^\top \beta_v\right)$

Figure 2.11: The generative process for a LDA/PF hybrid model.

This generative process specifies the joint distribution

$$p\left(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{w} \mid \alpha, \mu\right) = \prod_{k=1}^{K}\left[p(\beta_{\cdot,k} \mid \alpha) \prod_{d=1}^{D} p(\theta_{dk} \mid \mu)\right] \prod_{d=1}^{D} \prod_{v=1}^{V} p(w_{dv} \mid \theta_{dk}, \beta_{vk}). \quad (2.19)$$

To infer the posterior distribution over the hidden variables, we again construct a flexible approximating family,

$$q\left(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}\right) = \prod_{k=1}^{K}\left[q(\beta_{\cdot,k} \mid \lambda^{\beta}_{\cdot,k}) \prod_{d=1}^{D} p(\theta_{dk} \mid \lambda^{\theta}_{dk})\right], \quad (2.20)$$

and derive the complete conditional distributions for the latent parameters (see Appendix B). These derivations rely on auxiliary parameters $\mathbf{z}$, produced by the application of Property 2.4.1, which is identical to Equations (2.12) and (2.13). The complete conditionals for $\boldsymbol{\theta}$ and $\mathbf{z}$ are the same as shown in Equations (2.15) to (2.17), but the distribution for $\boldsymbol{\beta}$ is slightly modified:

$$\beta_{\cdot,k} \mid \boldsymbol{\theta}, \mathbf{z}, \mathbf{w}, \alpha, \mu \sim \text{Dirichlet}_V\left(\alpha + \sum_{d=1}^{D} \langle z_{d1k}, \cdots, z_{dVk} \rangle\right). \quad (2.21)$$

Algorithm 3 shows the full variational inference procedure for this model; it can be adapted

for a large number of observations as done for the PF inference procedure.

Algorithm 3: Mean Field Variational Inference for a LDA/PF Hybrid Model

Input: word counts w
Output: approximate posterior of latent parameters (global topics $\boldsymbol{\beta}$ and document representations $\boldsymbol{\theta}$) in terms of variational parameters $\boldsymbol{\lambda}$
Initialize $\mathbb{E}[\beta]$ to slightly random around uniform
Initialize $\mathbb{E}[\theta]$ to uniform
for *iteration* $i = 1 : M$ **do**
    set λ^{β} and λ^{θ} to respective priors
    update rate parameter $\lambda^{\theta,r} \mathrel{+}= \sum_{v=1}^{V} \mathbb{E}[\beta_v]$
    for *each document* $d = 1 : D$ **do**
        for *each vocabulary word* $v \in V(d)$[10] **do**
            set K-vector $\phi_{dv} \propto \langle \mathbb{E}[\theta_{d1}]\mathbb{E}[\beta_{v,1}], \ldots, \mathbb{E}[\theta_{dK}]\mathbb{E}[\beta_{v,K}] \rangle$
            set K-vector $\mathbb{E}[z_{dv}] = w_{dv} * \phi_{dv}$
            update shape parameter $\lambda_d^{\theta,s} \mathrel{+}= \mathbb{E}[z_{dv}]$
            update parameter $\lambda_v^{\beta} \mathrel{+}= \mathbb{E}[z_{dv}]$
        end
        set $\mathbb{E}[\theta_d] = \lambda_d^{\theta,s} / \lambda_d^{z,r}$
    end
    for *each topic* $k = 1 : K$ **do**
        set $\mathbb{E}[\beta_{\cdot,k}] \propto \lambda_k^{\beta}$
    end
end
return λ

2.6 Using Inferences for Exploration

When we introduced variational inference in Section 2.2, we made a brief note on how to use the the variational parameters $\boldsymbol{\lambda}$ returned by these optimization algorithms. These parameters define an approximate posterior distribution from which we can compute the expected values of the latent parameters; these expectations are used to explore the model and the original data.

However, the procedure for using these expectations for exploration is not always clear. This section surveys approaches to building and evaluating exploratory models. Using this

literature and the general visualization concepts described in Section 5.1, we propose five principles to guide the design and exploration of latent variable models in Chapter 5.

2.6.1 Modeling for Exploration

Exploration is about discovering relationships between constructs. For example, words obtain meaning in context (Condry, 2016); we cannot tell know if the word *plant* is a verb or a noun without the surrounding words. In the latent variable paradigm, we seek to explore the relationships between the latent and observed variables.

The challenge is that it is difficult to evaluate whether or not this exploration is successful. Computer science researchers tend to focus on the predictive ability of a model, but these objectives do not align with exploratory goals (Shmueli, 2010). At best, predictive evaluation proves a secondary use of an exploratory model; at worst, it introduces a "fictitious prediction problem" (Grimmer, 2013) to assuage reviewers that demand predictive evaluation or to bypass a more nuanced but onerous evaluation on the part of the researcher.

With topic models, the estimated probability of held-out documents is used to evaluate models (Wallach et al., 2009), but this and other traditional metrics are not correlated with human-evaluated coherence (Chang et al., 2009). One option is to intentionally optimize for semantic coherence (Mimno et al., 2011); another approach is to explicitly incorporate human input (Hu et al., 2014).

No matter the application, modeling is an elaborate process involving multiple iterations of making an attempt to represent the data-generating process and then critiquing the results (Blei, 2014). Human intuitions and feedback can be included at every stage; domain experts are invaluable in defining important concepts to include, and in critiquing, refining, and evaluating models.

The latent variable framework is particularly import in this process. Latent variables can

(and should) map to intuitive concepts. In the social sciences, researchers are often interested in constructs that are not directly observable, such as happiness (Diener, 2000), political ideals (Martin and Quinn, 2002), or sense of community (Glynn, 1981). We can design latent variables to capture these constructs and infer them from observed quantities.

When the object is to uncover a concrete measurement for an abstract construct, it is called a *measurement model*. One challenge is to determine whether or not a construct is valid; there are at least six ways to assess construct validity (Quinn et al., 2010). Validity and interpretability go hand in hand: if each latent variable has a corresponding valid construct, then it ensures that the individual variables and the model as a whole are interpretable.

Once we have a exploratory model fit to data, the next step is to investigate the results. This investigation allows researchers to verify that the model constructs are valid, to criticize the model such that it can be improved, and to understand the underlying data through the lens of the model.

We now turn to developing two exploratory models of human behavior (Chapters 3 and 4), after which we will describe general principles for designing and exploring latent variable models (Chapter 5).

3 Detecting and Characterizing Events

We can do nothing but scrutinize historical events
themselves if we want to discover what they are.

– Dean W. R. Matthews

Foreign embassies of the United States government communicate with each other and with the U.S. State Department through cabled message. The National Archive collects these documents in a running corpus, which traces the (unclassified) diplomatic history of the United States. It has collected, for example, about two million cables sent between 1973 and 1978.

Typically, a cable from this collection describes diplomatic "business as usual," such as arrangements for visiting officials, recovery of lost or stolen passports, or obtaining lists of names for meetings and conferences. For example, the embassies sent 8,635 cables during the week of April 21, 1975. Here is one, selected at random,

> Hoffman, UNESCO Secretariat, requested info from PermDel concerning an official invitation from the USG RE subject meeting scheduled 10-13 JUNE 1975, Madison, Wisconsin. Would appreciate info RE status of action to be taken in order to inform Secretariat. Hoffman communicating with Dr. John P. Klus RE list of persons to be invited.

But hidden in the corpus are also cables about important diplomatic events, the cables and events that are of primary interest to historians. During that same week, the United States was in the last moments of the Vietnam war and, on April 30, 1975, lost its hold on Saigon. This resulted in the end of the Vietnam War and a max exodus of refugees from the country. One of the cables around this event is

> GOA program to move Vietnamese Refugees to Australia is making little progress and probably will not cover more than 100-200 persons. Press comment on smallness of program has recognized difficulty of getting Vietnamese out of Saigon, but "Canberra Times" Apr 25 sharply critical of government's performance. [...] Labor government clearly hopes whole matter will somehow disappear.

Our goal in this chapter is to develop a method to help historians and political scientists wade through their collections, such as the 1970s cables, to find potentially important events, such as the fall of Saigon, and the primary sources around them. We develop *Capsule*, a probabilistic model for detecting and characterizing important events in large collections of historical communication.

Figure 3.1 illustrates Capsule's analysis of the two million cables from the National Archives. The y-axis is "eventness", a loose measure how strongly a week's cables deviate from the usual diplomatic chatter to discuss a matter that is common to many embassies. (This is described in detail in Section 3.2.)

The figure shows that Capsule detects many of the important moments during this five-year span, including the Air France hijacking and Israeli rescue operation "Operation Entebbe" (June 27–July 4, 1976), and the fall of Saigon (April 30, 1975). It also identifies other moments, such as the U.S. sharing lunar rocks with other countries (March 21, 1973) and the death of Mao Tse-tung (Sept. 9, 1976). Broadly speaking, Capsule gives a picture of the diplomatic history of these five years; it identifies and characterizes moments and source

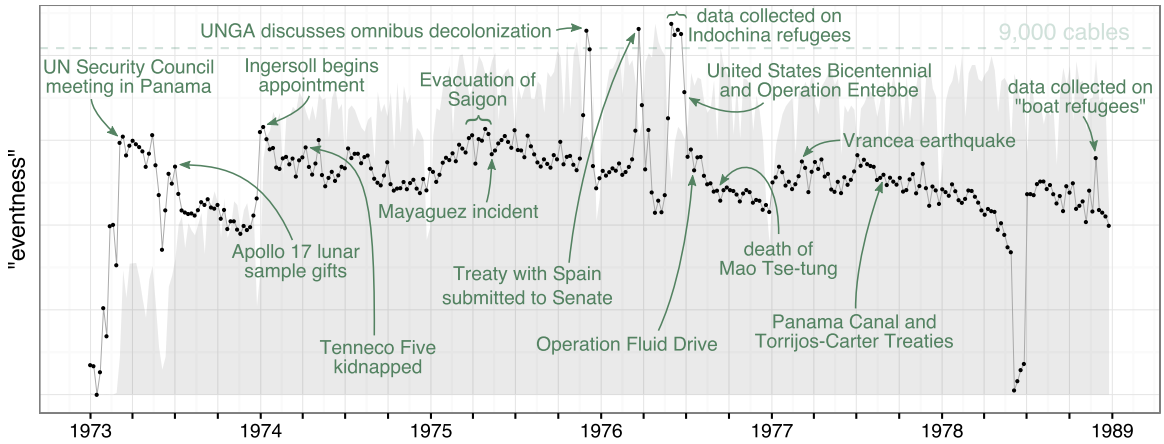


Figure 3.1: Measure of "eventness," or time interval impact on cable content (Eq. 3.2). Grey background indicates the number of cables sent over time. This comes from the model fit we discuss in Section 3.4. Capsule successful detects real-world events from National Archive diplomatic cables.

material that might be of interest to a historian.

The intuition behind Capsule is this: embassies write cables throughout the year, usually describing typical business such as the visiting of a government official. Sometimes, however, there is an important event, e.g., the fall of Saigon. When an event occurs, it pulls embassies away from their typical business to write cables that discuss what happened and its consequences. Thus Capsule effectively defines an "event" to be a moment in history when embassies deviate from what each usually discusses, and when each embassy deviates in the same way.

Capsule embeds this intuition into a Bayesian model. It uses hidden variables to encode what "typical business" means for each embassy, how to characterize the events of each week, and which cables discuss those events. Given a corpus, the corresponding posterior distribution provides a filter on the cables that isolates important moments in the diplomatic history. Figure 3.1 illustrates the mean of this posterior.

Capsule can be used to explore any corpora with the same underlying structure: text (or other discrete multivariate data) generated over time by known entities. This includes email, consumer behavior, social media posts, and opinion articles.

We present the model in Section 3.2, providing a formal model specification and give guidance on how to use the model posterior to detect and characterize real-worlds events (Chapter 5 presents a visualization scaffold for the model). In Section 3.4, we evaluate Capsule and explore its results on a collection of U.S. State Department cables and on simulated data.

3.1 Related work

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

In both univariate and multivariate settings, the goal is often that analysts want to predict whether or not rare events will occur (Weiss and Hirsh, 1998; Das et al., 2008). Capsule, in contrast, is designed to help analysts explore and understand the original data: our goal is interpretability, not prediction.

Events can also be construed as "change points" to mark when typical observations shift semi-permanently from one value to another (Guralnik and Srivastava, 1999; Adams and MacKay, 2007). Both varieties of events are important, but we focus on temporary shifts away from normal.

A common goal is to identify clusters of documents; these approaches are used on news articles (Zhao et al., 2012, 2007; Zhang et al., 2002; Li et al., 2005; Wang et al., 2007; Allan et al., 1998) and social media posts (VanDam, 2012; Lau et al., 2012; Jackoway et al., 2011; Sakaki et al., 2010; Reuter and Cimiano, 2012; Becker et al., 2010; Sayyadi et al., 2009). 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, usually weighted by tf-idf value (Fung et al., 2005; Kumaran and Allan, 2004; Brants et al., 2003; Das Sarma et al., 2011; Zhao et al., 2007, 2012); here, events are bursts in groups of terms.

Topic models (Blei, 2012) reduce the dimensionality of text data; they have been used to help detect events mentioned in social media posts (Lau et al., 2012; Dou et al., 2012) and posts relevant to monitored events (VanDam, 2012). We rely on topic models to characterize both typical content and events, but grouped observations can also be summarized directly (Peng et al., 2007; Chakrabarti and Punera, 2011; Gao et al., 2012).

In addition to text data over time, author (Zhao et al., 2007), news outlet (Wang et al., 2007), and spatial information (Neill et al., 2005; Mathioudakis et al., 2010; Liu et al., 2011) can be used to augment event detection. Capsule uses author information in order to characterize the typical concerns of authors.

Detecting and characterizing relationships (Schein et al., 2015; Linderman and Adams, 2014; Das Sarma et al., 2011) is related to event detection. When a message recipient is known, Capsule can use a sender-receiver pair in place of an author, but the model could be further tailored for network interactions.

3.2 The Capsule Model

In this section we develop the Capsule model for detecting and characterizing events. Capsule relies on text data sent between entities over time, and builds on topics models. We first give the intuition on Capsule, then formally specify the model. We also describe how we learn its hidden variables.

Consider an entity like the Bangkok American embassy, shown in Figure 3.2. 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

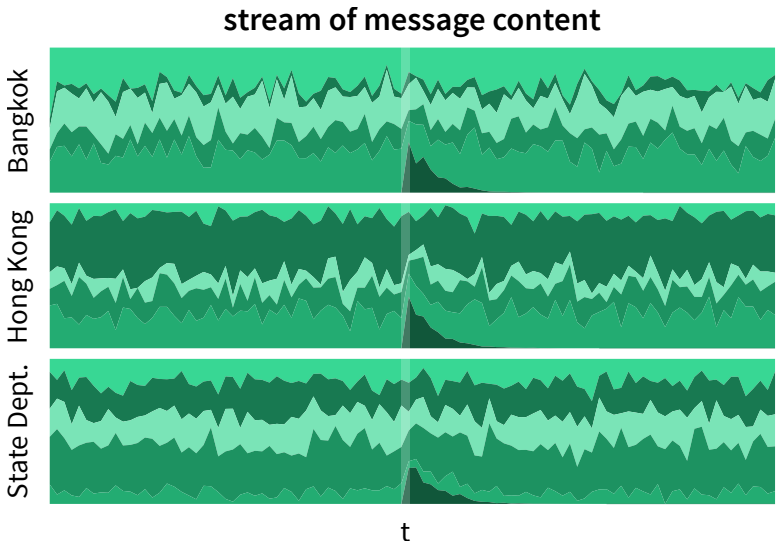


Figure 3.2: Cartoon intuition of Capsule; the y axis is the stacked proportion of messages about various subjects during a given time interval. The Bangkok embassy, Honk Kong embassy, and State Department all have typical concerns about which they usually send messages. When an events occurs at time t, the stream of message content alters to include the event, then fades back to "business as usual." Capsule discovers entities' typical concerns as well as the event occurrence and content.

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 t, an event occurs, such as the capture of Saigon during the Vietnam War. We do not directly observe that events occur, but we do observe the message stream. Using this stream, each event will be described as a distribution over the vocabulary, similar to how topics are distributions over these same terms. When an event occurs, the message content changes for multiple entities—significant events impact multiple parties simultaneously. The day following the capture of Saigon, for instance, the majority of the diplomatic cables sent by the Bangkok embassy and several other entities 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.

topic type	top terms
general	visit, hotel, schedule, arrival
entity	soviet, moscow, ussr, agreement
event	saigon, evacuation, vietnam, help

Table 3.1: Top vocabulary terms for examples of each of the three topic varieties; these three types of topics blend to form the distribution of each message. They come from the model fit we discuss in Section 3.4.

3.2.1 Model Specification

We now define Capsule in detail. Our data are *entities* sending *messages* over *time*. The observed variables are $w_{d,v}$, the number of times term v occurs in message d. The message is associated with an entity (or author) a_d and a time (or date) interval i_d.

We model each message with a bank of Poisson distributions, one per term in the vocabulary, $w_{d,v} \sim \text{Poisson}(\lambda_{d,v})$. The rate $\lambda_{d,v}$ blends the different influences on the content of the message, which are defined in terms of different types of *topics*. A topic, as in typical topic modeling (Blei et al., 2003; Canny, 2004; Gopalan et al., 2014b), is a distribution over terms.

Specifically, the message blends general topics about diplomacy (e.g., diplomats, communication) β_k, an entity topic that is specific to the author of the message (e.g., terms about France) η_{a_d},[1] and an event topic that is specific to the events of relevant recent weeks (e.g., terms about an international crisis) γ_t. Notice how messages share these topics in different configurations: all messages share the general topics; messages from the same entity share the entity topics; and messages from the same interval share the event topics.

Examples of these three types of topics are in Table 3.1—the general topic relates to planning travel, the entity topic captures words related to the U.S.S.R., and the event topic captures words related to the evacuation of Saigon toward the end of the Vietnam War.

[1] These entity-specific topics are similar to background topics (Paul and Dredze, 2012).

Each message blends its corresponding topics with different strengths, which are drawn per message. Each message represents a different mix of the events of recent weeks, entity-specific items, and general diplomacy.

Putting this together, the Poisson rate for term v in document d is

$$\lambda_{d,v} = \theta_d^\top \beta_v + \zeta_d \eta_{a_d,v} + \sum_{t=1}^{T} f(i_d, t)\epsilon_{d,t}\gamma_{t,v}, \tag{3.1}$$

where θ_d corresponds to strength of general diplomacy, ζ_d corresponds to strength of entity-specific concerns, and ϵ_d corresponds to strength of events; f is some function of decay. This function is important because events should not remain at their full strength indefinitely, but should decay over time. In our experiments, we find that exponential decay, as in Equation (3.21), performs well.

We place gamma priors on the topic strengths and Dirichlet priors on the topics. The distributions of general and entity topic strengths are defined hierarchically by entity, capturing the different topics that each entity tends to discuss. The prior on the entity strength is also defined hierarchically; different weeks are more or less "eventful." The graphical model is shown in Figure 3.3 and the generative process is in Figure 3.4.

There are connections between Capsule and recent work on Poisson processes. In particular, we can interpret Capsule as a collection of related discrete time Poisson processes with random intensity measures. Further, marginalizing out the event strength prior reveals that word use from one entity can "excite" word use in another, which suggests a close relationship to Hawkes processes (Hawkes, 1971).

Given a collection of messages, posterior inference uncovers the different types of topics and how each message exhibits them. We will see below, how inferences about the event strengths enable us to filter the corpus to find important messages.

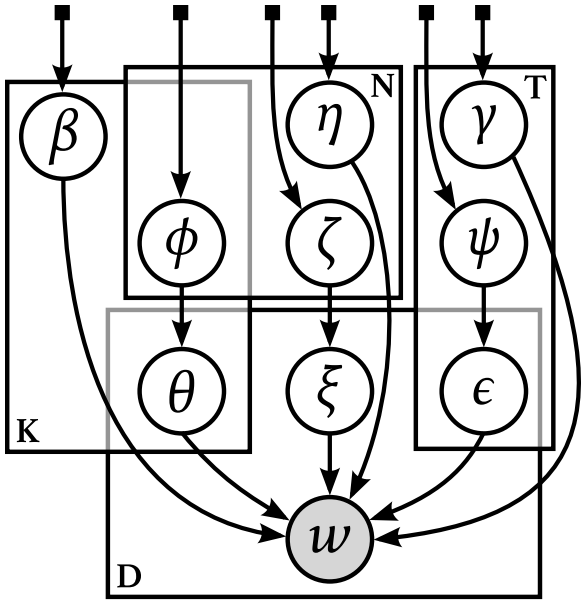


Figure 3.3: The graphical model for Capsule. Observed words w depend on general topics β, entity-specific topics η, and event topics γ, as well as document representations θ, ξ, and ϵ. Variables ϕ and ζ represent entity concerns (with general topics and entity-specific topics, respectively) and ψ represents the event strength of a given time interval. Hyper-parameters are indicated by black squares, but not labeled for visual simplicity.

3.2.2 Detecting and Characterizing Events

Once we estimate the posterior distribution of the Capsule parameters, described in the following section, we can use the expectations of the latent parameters to explore the original data. To detect events, we consider the proportion of the document about an event, and take a weighted average of these proportions:

$$m_t = \frac{1}{\sum_{d=1}^{D} f(i_d, t)} \sum_{d=1}^{D} \frac{f(i_d, t)\mathbb{E}[\epsilon_{d,t}]}{\zeta_d + \sum_{j=1}^{T} f(i_d, j)\mathbb{E}[\epsilon_{d,j}] + \sum_{k=1}^{K} \mathbb{E}[\theta_{d,k}]}. \tag{3.2}$$

This measure of "eventness" provides an estimate of the proportion of words that are related to a real-world event in that interval. Figure 3.1 shows events detected with this metric.

Given an identified event, we can characterize it in terms of its top terms under γ, but we can also use weighted event relevancy parameters $f(i_d, t)\epsilon_{d,t}$ to sort documents; Section 3.4

- for each time step $t = 1{:}T$,
  - draw interval description over vocabulary (event topic) $\gamma_t \sim \text{Dirichlet}_V(\alpha)$
  - draw interval strength $\psi_t \sim \text{Gamma}(s_\psi, r_\psi)$
- for each entity $n = 1{:}N$,
  - draw entity-specific topics over vocabulary $\eta_n \sim \text{Dirichlet}_V(\alpha)$
  - draw entity-specific topic strength $\xi_n \sim \text{Gamma}(s_\xi, r_\xi)$
- for each topic $k = 1{:}K$,
  - draw general topic distribution over vocabulary $\beta_k \sim \text{Dirichlet}_V(\alpha)$
  - for each entity $n = 1{:}N$,
    - draw general entity concern $\phi_{n,k} \sim \text{Gamma}(s_\phi, r_\phi)$
- for each document $d = 1{:}D$ sent at time i_d by author a_d,
  - draw local entity concern $\zeta_d \sim \text{Gamma}(s_\zeta, \xi_{a_d})$
  - for each topic $k = 1{:}K$,
    - draw local entity concern $\theta_{d,k} \sim \text{Gamma}(s_\theta, \phi_{a_d,k})$
  - for each time $t = 1{:}T$,
    - draw local interval relevancy $\epsilon_{d,t} \sim \text{Gamma}(s_\epsilon, \psi_t)$
  - for each vocabulary term $v = 1{:}V$,
    - set $\lambda_{d,v} = \theta_d^\top \beta_v + \zeta_d \eta_{a_d} + \sum_{t=1}^{T} f(i_d, t)\epsilon_{d,t}\gamma_{t,v}$
    - draw word counts $w_{d,v} \sim \text{Poisson}\left(\lambda_{d,v}\right)$

Figure 3.4: The generative process for Capsule.

explores relevant documents for events found in the National Archive diplomatic cables data. In addition to detecting and characterizing events, Capsule can be used to explore entity concerns and the general themes in a given collection.

3.3 Variational Inference for Capsule

In order to use the Capsule model to explore the observed documents, we must compute the posterior distribution. Conditional on the observed word counts w, our goal is to compute the posterior values of the hidden parameters—general topics $\boldsymbol{\beta}$, entity topics $\boldsymbol{\eta}$, event topics $\boldsymbol{\gamma}$, entity concerns $\boldsymbol{\phi}$ (for general topics) and $\boldsymbol{\xi}$ (for their own topic), overall event strengths $\boldsymbol{\psi}$, and document-specific strengths for general topics $\boldsymbol{\theta}$, entity topics $\boldsymbol{\zeta}$, and event topics $\boldsymbol{\epsilon}$.

As for many Bayesian models, the exact posterior for Capsule is not tractable to compute and it must be approximated. In this section, we develop an approximate inference algorithm for Capsule based on variational methods (see Section 2.2).

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,

$$\mathcal{L}(q) \;=\; \mathbb{E}_q\big[\log p(w, \psi, \gamma, \phi, \beta, \xi, \eta, \theta, \epsilon, \zeta) \;-\; \log q(\psi, \gamma, \phi, \beta, \xi, \eta, \theta, \epsilon, \zeta)\big]. \quad (3.3)$$

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

$$q(\psi, \gamma, \phi, \beta, \xi, \eta, \theta, \epsilon, \zeta) = \prod_{d=1}^{D} \left[q(\zeta_d \mid \lambda_d) \prod_{k=1}^{K} q(\theta_{d,k} \mid \lambda^{\theta}_{d,k}) \prod_{t=1}^{T} q(\epsilon_{d,t} \mid \lambda^{\epsilon}_{d,t}) \right]$$
$$\prod_{t=1}^{T} \left[q(\gamma_t \mid \lambda^{\gamma}_t) q(\psi_t \mid \lambda^{\psi}_t) \right] \prod_{n=1}^{N} \left[q(\xi_n \mid \lambda^{\xi}_n)\, q(\eta_n \mid \lambda^{\eta}_n) \right] \prod_{k=1}^{K} \left[q(\beta_k \mid \lambda^{\beta}_k) \prod_{n=1}^{N} q(\phi_{n,k} \mid \lambda^{\phi}_{n,k}) \right] \tag{3.4}$$

The variational distributions for the topics $q(\gamma)$, $q(\beta)$ and $q(\eta)$ are all Dirichlet-distributed with free variational parameters λ^{γ}, λ^{β}, and λ^{η} respectively. Similarly, the variational distributions $q(\psi)$, $q(\phi)$, $q(\xi)$, $q(\theta)$, $q(\epsilon)$, and $q(\zeta)$ are all gamma-distributed with corresponding free variational parameters λ^{ψ}, λ^{ϕ}, λ^{ξ}, λ^{θ}, λ^{ϵ} and λ^{ζ}. For these gamma-distributed variables, each free parameter λ has two components: shape s and rate r.

The expectations under q, which are needed to maximize the ELBO, have closed form analytic updates–we update each parameter in turn, following standard coordinate ascent variational inference techniques, as the Capsule model is specified with the required conjugate relationships that make this approach possible Ghahramani and Beal (2001).

To obtain simple updates, we first rely on auxiliary latent variables z. These variables, when marginalized out, leave the original model intact. We apply 2.4.1 to the word count rate in Equation (3.1) and define Poisson variables for each component of the word count:

$$z^{\mathcal{K}}_{d,v,k} \sim \text{Poisson}(\theta_{d,k}\beta_{k,v}),$$

$$z^{\mathcal{E}}_{d,v} \sim \text{Poisson}(\zeta_d \eta_{a_d,v}),$$

$$z^{\mathcal{T}}_{d,v,t} \sim \text{Poisson}\left(f(i_d, t)\epsilon_{d,t}\gamma_{t,v}\right).$$

The $\mathcal{K}$, $\mathcal{E}$, and $\mathcal{T}$ superscripts indicate the contributions from general, entity, and event

topics, respectively. Given these variables, the total word count is deterministic:

$$w_{d,v} = \sum_{k=1}^{K} z^{\mathcal{K}}_{d,v,k} + z^{\mathcal{E}}_{d} + \sum_{t=1}^{T} z^{\mathcal{T}}_{d,v,t}.$$

Coordinate-ascent variational inference is derived from complete conditionals, i.e., the conditional distributions of each variable given the other variables and observations. These conditionals define both the form of each variational factor and their updates. The following are the complete conditional for each of the gamma- and Dirchlet-distributed latent parameters. The notation $D(n)$ is used for the set of documents sent by entity n; $D(t)$ is the set of documents sent impacted by events at time t (e.g., all documents after the event in the case of exponential decay).

$$\gamma_t \mid \mathbf{W}, \psi, \phi, \xi, \beta, \eta, \theta, \epsilon, \zeta, z \quad \sim \quad \text{Dirichlet}_V \left(\alpha_\gamma + \sum_{d=1}^{D} \langle z^{\mathcal{T}}_{d,1,t}, \cdots, z^{\mathcal{T}}_{d,V,t} \rangle \right) \tag{3.5}$$

$$\eta_n \mid \mathbf{W}, \psi, \phi, \xi, \beta, \gamma, \theta, \epsilon, \zeta, z \quad \sim \quad \text{Dirichlet}_V \left(\alpha_\eta + \sum_{d \in D(n)} \langle z^{\mathcal{E}}_{d,v}, \cdots, z^{\mathcal{E}}_{d,v} \rangle \right) \tag{3.6}$$

$$\beta_k \mid \mathbf{W}, \psi, \phi, \xi, \gamma, \eta, \theta, \epsilon, \zeta, z \quad \sim \quad \text{Dirichlet}_V \left(\alpha_\beta + \sum_{d=1}^{D} \langle z^{\mathcal{K}}_{d,1,k}, \cdots, z^{\mathcal{K}}_{d,V,k} \rangle \right) \tag{3.7}$$

$$\psi_t \mid \mathbf{W}, \phi, \xi, \beta, \gamma, \eta, \theta, \epsilon, \zeta, z \quad \sim \quad \text{Gamma} \left(s_\psi + |D(t)| s_\epsilon, r_\psi + \sum_{d \in D(t)} \epsilon_{d,t} \right) \tag{3.8}$$

$$\xi_n \mid \mathbf{W}, \psi, \phi, \beta, \gamma, \eta, \theta, \epsilon, \zeta, z \quad \sim \quad \text{Gamma}\left(s_\xi + |D(n)|s_\xi, r_\xi + \sum_{d \in D(n)} \zeta_d\right) \tag{3.9}$$

$$\phi_{n,k} \mid \mathbf{W}, \psi, \xi, \beta, \gamma, \eta, \theta, \epsilon, \zeta, z \quad \sim \quad \text{Gamma}\left(s_\phi + |D(n)|s_\theta, r_\phi + \sum_{d \in D(n)} \theta_{d,k}\right) \tag{3.10}$$

$$\theta_{d,k} \mid \mathbf{W}, \psi, \phi, \xi, \beta, \gamma, \eta, \epsilon, \zeta, z \quad \sim \quad \text{Gamma}\left(s_\theta + \sum_{v=1}^{V} z^{\mathcal{K}}_{d,v,k}, \phi_{a_d,k} + \sum_{v=1}^{V} \beta_{k,v}\right) \tag{3.11}$$

$$\epsilon_{d,t} \mid \mathbf{W}, \psi, \phi, \xi, \beta, \gamma, \eta, \theta, \zeta, z \quad \sim \quad \text{Gamma}\left(s_\epsilon + \sum_{v=1}^{V} z^{\mathcal{T}}_{d,v,t}, \psi_t + f(i_d, t) \sum_{v=1}^{V} \gamma_{t,v}\right) \tag{3.12}$$

$$\zeta_d \mid \mathbf{W}, \psi, \phi, \xi, \beta, \gamma, \eta, \theta, \epsilon, z \quad \sim \quad \text{Gamma}\left(s_\zeta + \sum_{v=1}^{V} z^{\mathcal{E}}_{d,v}, \xi_{a_d} + \sum_{v=1}^{V} \eta_{a_d,v}\right) \tag{3.13}$$

The complete conditional for the auxiliary variables has the form

$$z_{d,v} \mid \psi, \phi, \xi, \beta, \gamma, \eta, \theta, \epsilon, \zeta \sim \text{Mult}(w_{d,v}, \omega_{d,v}),$$

where

$$\omega_{d,v} \propto \langle \theta_{d,1}\beta_{1,v}, \cdots, \theta_{d,K}\beta_{K,v}, \zeta_d \eta_{a_d,v}, f(i_d, 1)\epsilon_{d,1}\gamma_{1,v}, \cdots, f(i_d, T)\epsilon_{d,T}\gamma_{T,v} \rangle. \tag{3.14}$$

Intuitively, these variables allocate the data to one of the entity concerns or events, and thus can be used to explore the data.

Given these conditionals, the algorithm sets each parameter to the expected conditional parameter under the variational distribution. The mean field assumption guarantees that this expectation will not involve the parameter being updated. Algorithm 4 shows our variational inference algorithm.

This algorithm uses the notation $\boldsymbol{\lambda}$ to refer to the set of variational parameters,

$$\boldsymbol{\lambda} = \{\lambda^{\gamma}, \lambda^{\eta}, \lambda^{\beta}, \lambda^{\psi}, \lambda^{\xi}, \lambda^{\phi}, \lambda^{\theta}, \lambda^{\epsilon}, \lambda^{\xi}\}.$$

The notation $V(d)$ is the set of vocabulary indices for the collection of words in document d. We could also iterate over all V, but as zero word counts give $\mathbb{E}[z_{d,v}] = 0 \; \forall v \notin V(d)$, the two are equivalent.

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. Source code is available at `https://github.com/ajbc/capsule`.

3.4 Evaluation

In this section we explore the performance of Capsule on simulated data and a collection of over 2 million U.S. State Department diplomatic cables from the 1970s.

3.4.1 Results on Simulated Data

Prior to exploring Capsule results on data of historical interest, we provide a quantitative assessment of the model on simulated data.

We generated ten data sets, each with 100 time steps, 10 general topics, and 100 entities. Each simulation contained about 20,000 documents and followed the generative process

Algorithm 4: Variational Inference for Capsule

Input: word counts w
Output: approximate posterior of latent parameters in terms of variational parameters $\boldsymbol{\lambda}$
Initialize $\mathbb{E}[\beta]$ to slightly random around uniform
Initialize $\mathbb{E}$[all other parameters] to uniform
for *iteration* $m = 1 : M$ **do**
  set all $\boldsymbol{\lambda}$ to respective priors, excluding $\lambda^{\theta,rate}$, $\lambda^{\zeta,rate}$, and $\lambda^{\epsilon,rate}$, which are set to 0
  update $\lambda^{\theta,rate} \mathrel{+}= \sum_V \mathbb{E}[\beta_v]$
  for *each document* $d = 1 : D$ **do**
    for *each term* $v \in V(d)$[2] **do**
      set $(K+T+1)$-vector $\omega_{d,v}$ as shown in eq. (3.14), using $\mathbb{E}$ of parameters
      set $(K+T)$-vector $\mathbb{E}[z_{d,v}] = w_{d,v} * \omega_{d,v}$
      update $\lambda_d^{\theta,shape} \mathrel{+}= \mathbb{E}[z_{d,v}^{\mathcal{K}}]$ [eq. (3.11)]
      update $\lambda_d^{\epsilon,shape} \mathrel{+}= \mathbb{E}[z_{d,v}^{\mathcal{K}}]$ [eq. (3.12)]
      update $\lambda_d^{\zeta,shape} \mathrel{+}= \mathbb{E}[z_{d,v}^{\mathcal{E}}]$ [eq. (3.13)]
      update $\lambda_v^{\beta} \mathrel{+}= \mathbb{E}[z_{d,v}^{\mathcal{K}}]$ [eq. (3.7)]
      update $\lambda_v^{\gamma} \mathrel{+}= \mathbb{E}[z_{d,v}^{\mathcal{T}}]$ [eq. (3.5)]
      update $\lambda_v^{\eta} \mathrel{+}= \mathbb{E}[z_{d,v}^{\mathcal{E}}]$ [eq. (3.6)]
    end
    set $\lambda_d^{\theta,rate} = \mathbb{E}[\phi_{a_d}] + \sum_v \mathbb{E}[\beta]$ [eq. (3.11)]
    set $\lambda_d^{\epsilon,rate} = \mathbb{E}[\psi] + f \sum_v \mathbb{E}[\gamma]$ [eq. (3.12)]
    set $\lambda_d^{\zeta,rate} = \mathbb{E}[\xi_{a_d}] + \sum_v \mathbb{E}[\eta]$ [eq. (3.13)]
    set $\mathbb{E}[\theta_d] = \lambda_d^{\theta,shape} / \lambda_d^{\theta,rate}$
    set $\mathbb{E}[\epsilon_d] = \lambda_d^{\epsilon,shape} / \lambda_d^{\epsilon,rate}$
    set $\mathbb{E}[\zeta_d] = \lambda_d^{\zeta,shape} / \lambda_d^{\zeta,rate}$
    update $\lambda_{a_d}^{\phi,shape} \mathrel{+}= s_\theta$ [eq. (3.10)]
    update $\lambda_t^{\psi,shape} \mathrel{+}= s_\epsilon \forall t : f(i_d, t) \neq 0$ [eq. (3.8)]
    update $\lambda_{a_d}^{\xi,shape} \mathrel{+}= s_\eta$ [eq. (3.9)]
    update $\lambda_{a_d}^{\phi,rate} \mathrel{+}= \theta_d$ [eq. (3.10)]
    update $\lambda^{\psi,rate} \mathrel{+}= \epsilon_d$ [eq. (3.8)]
    update $\lambda_{a_d}^{\xi,rate} \mathrel{+}= \zeta_d$ [eq. (3.9)]
  end
  set $\mathbb{E}[\phi] = \lambda^{\phi,shape} / \lambda^{\phi,rate}$
  set $\mathbb{E}[\beta_k] = \lambda^{\beta_{k,v}} / \sum_v \lambda^{\beta_k} \forall k$
  set $\mathbb{E}[\xi] = \lambda^{\xi,shape} / \lambda^{\xi,rate}$
  set $\mathbb{E}[\eta_n] = \lambda^{\eta_{n,v}} / \sum_v \lambda^{\eta_n} \forall n$
  set $\mathbb{E}[\psi] = \lambda^{\psi,shape} / \lambda^{\psi,rate}$
  set $\mathbb{E}[\gamma_t] = \lambda^{\gamma_{t,v}} / \sum_v \lambda^{\gamma_t} \forall t$
end
return λ

assumed by Capsule, as shown in Figure 3.4.

To evaluate event detection, we created a ranked list of all time intervals and computed the overlap between a method and the simulated ground at every threshold; this generates an curve under which we can compute the area and normalized based on ideal performance—we refer to this metric as event detection AUC:

$$\text{event detection AUC} = \frac{\sum_{i=1}^{T} |\text{Truth}_i \cap \text{Model}_i|}{\sum_{i=1}^{T} i}, \tag{3.15}$$

where Model_i is a set of the top i most eventful intervals according to the model, and Truth_i is the known set of the top i most eventful intervals. As the data is simulated, we can order all intervals by their known "eventness"—this metric captures how well the model recovers the true ordering.

The most successful of the baseline methods for event detection were based on absolute error in word count relative to the mean. This can be computed for all words:

$$\text{word count deviation} = \sum_{v=1}^{V} \left[\sum_{d=1}^{D} \text{abs} \left(w_{d,v} - \frac{1}{|D|} \sum_{d=1}^{D} w_{d,v} \right) \right], \tag{3.16}$$

and can also be weighted by tf-idf,

$$\text{tf-idf word deviation} = \sum_{v=1}^{V} \text{tf-idf}(v) \left[\sum_{d=1}^{D} \text{abs} \left(w_{d,v} - \frac{1}{|D|} \sum_{d=1}^{D} w_{d,v} \right) \right]. \tag{3.17}$$

We also considered metrics that computed deviations on the entity and document level, but the simplest overall metrics performed best.

Figure 3.5 shows that Capsule[3] outperforms these approaches for event detection. We also consider an "event only" model—this is a model that only uses the interval-related subset of Capsule's parameters; comparing to this shows that is it important to model "business

[3]The model was set with the same number of topics $K = 10$ and exponential decay f used to simulate the data. More details on the decay function surround its formal definition in Equation (3.21).

as usual" for improved event detection. LDA based approaches like average deviation from mean in topic space (Dou et al., 2012) do not perform well for event detection as deviations in topic space are too coarse to provide a meaningful signal.

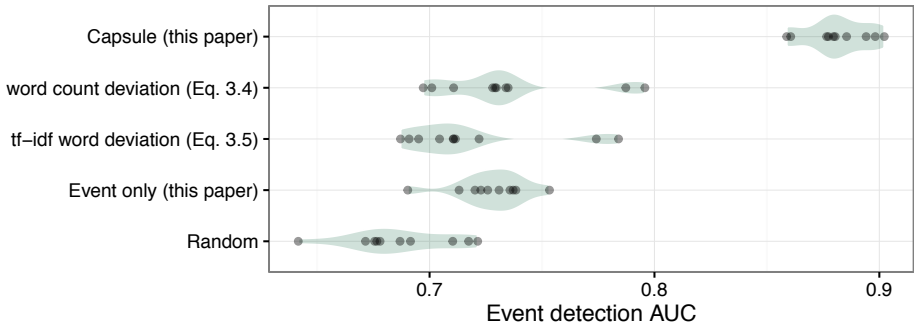


Figure 3.5: Event detection performance on ten simulated datasets; each dot is the performance on a single dataset, and the shaded green describes the distribution of the performances. Capsule detects events better than comparison methods.

Once events have been identified, our next task is to identify relevant documents; to evaluate this, we calculate precision of recovering the top N documents, or

$$\text{precision@N} = \frac{|(\text{set of true top } N \text{ docs}) \cap (\text{set of model top } N \text{ docs})|}{N}. \tag{3.18}$$

Both Capsule and its event-only partial model outperform all comparison methods in terms of document recovery. For Capsule, average precision at 10 documents was 0.44; the event-only model had average precision of 0.09. LDA performed slightly worse than the event-only model, and the other comparison methods (similar to Equations 3.16 and 3.17) recovered zero relevant documents–equivalent to random.

Model Sensitivity. We assessed the sensitivity of our model to three different decay functions f: exponential, linear, and step functions. We simulated data for each function and then fit Capsule using every permutation of f and multiple settings for event decay

duration. We considered a step function,

$$f(i_d, t) = \begin{cases} 1, & \text{if } t \leq i_d < t + \tau \\ 0, & \text{otherwise,} \end{cases} \tag{3.19}$$

as well as linear decay,

$$f(i_d, t) = \begin{cases} 1 - \frac{i_d - t}{\tau}, & \text{if } t \leq i_d < t + \tau \\ 0, & \text{otherwise.} \end{cases} \tag{3.20}$$

and an exponential decay function:

$$f(i_d, t) = \begin{cases} 0, & \text{if } t \leq i_d < t + \tau \\ \exp\left\{\frac{-(i_d - t)}{\tau/5}\right\}, & \text{otherwise.}[4] \end{cases} \tag{3.21}$$

We used duration $\tau = 3$ and simulated ten data sets for each of the three functions f. In fitting the models, we also considered all three functions f and varied the decay duration τ from 1 to 5. Figure 3.6 shows the results of these experiments, using both event detection and document recovery metrics discussed previously.

As expected, the model performs best when the model decay function matches the function used to generate the data. For both event detection and document recovery, the exponential decay was least sensitive to the setting of duration τ used in fitting the data; it was also the least sensitive to the function used in simulating the data. In exploring results on the real-world cable data, we found that the exponential decay provided the most interpretable results.

[4] Unlike the linear and step functions, the exponential function could be evaluated for any time interval t after a document's appearance at i_d; the function is truncated for computational reasons. The mean lifetime of this exponential decay is the duration τ is divided by 5—this ensures that 99.3% of the area under the curve is reached before the function is truncated at duration τ.

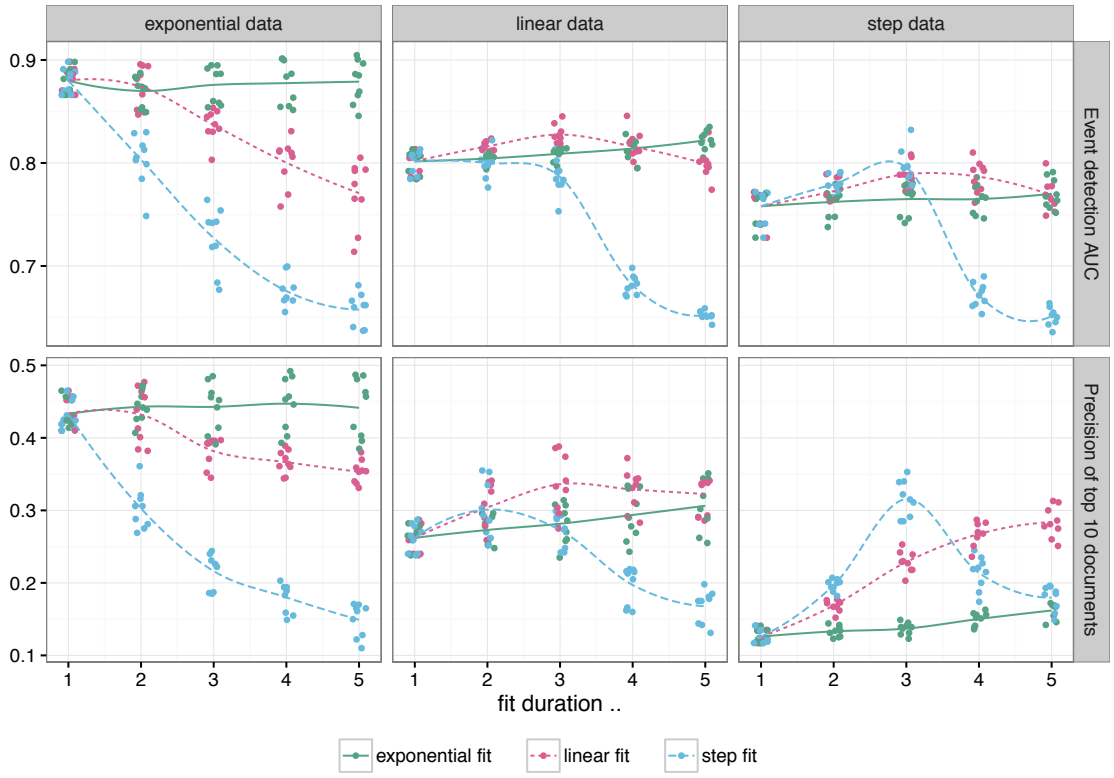


Figure 3.6: Assessment of model parameter sensitivity on simulated data–Capsule performs best when the model decay function matches the function used to generate the data. The exponential decay is least sensitive to the setting of duration τ and the true function f.

3.4.2 Results on U.S. State Department Diplomatic Cables

As Capsule is intended to be used to explore large collections of documents, we must demonstrate its use in that context. This sections describes and explores the application of Capsule to a real-world collection of diplomatic messages.

Data. The National Archive collects communications between the U.S. Sate Department and its embassies. We obtained a collection of these diplomatic messages from the History Lab at Columbia,[5] which received them from the Central Foreign Policy Files at the National

[5] http://history-lab.org

Archives. The communications in this data set were sent between 1973 and 1978.

In addition to the text of the cables themselves, each document is supplemented with information about who sent the cable (e.g., the State Department, the U.S. Embassy in Saigon, or an individual by name), who received the cable (often multiple entities), and the date the cable was sent. We used a vocabulary of size 6,293 and omitted cables with fewer than three terms, resulting in a collection of 2,021,852 messages sent between 22,961 entities. We selected a weekly duration for the time intervals, as few cables were sent on the weekends.

Model Settings. We fit Capsule with $K = 100$ general topics and using the exponential decay f, shown in Equation (3.21), with event duration $\tau = 4$. With these settings on the cables data, fitting the model takes about one hour per iteration.[6]

Quantitative Results. The History Lab at Columbia provided a list of 39 real-world events in the time period covered by the cables data; they validated that these events were present in at least one of six reputable collections of events, such as the Office of the Historian list of milestones.[7]

We ran Capsule and baseline comparison methods to recover these events, and used the nDCG metric to evaluate the methods. The nDCG metric is discounted cumulative gain,

$$\text{DCG} = \sum_{j=1}^{T} \frac{\mathbf{1}[\text{interval at rank } j \text{ in known events}]}{\log j}, \tag{3.22}$$

divided by the ideal DCG value, or

$$\text{nDCG} = \frac{\text{DCG}}{\text{ideal DCG}}. \tag{3.23}$$

As shown in Table 3.2, Capsule outperforms the baselines.

Additionally, we can compute held-out validation data likelihood on the model and each of

[6]Our algorithm is batch–we consider each data point for every iteration. Modifying the algorithm to stochastically sample the data would reduce the time required to achieve an equivalent model fit.

[7]https://history.state.gov/milestones/1969-1976

Method	nDCG
Capsule	0.693
Average tf-idf weighted word count deviation	0.652
Average unweighted word count deviation	0.642
Single term maximum tf-idf weighted deviation	0.561
Random (10k ave)	0.557
Single term maximum unweighted deviation	0.555

Table 3.2: Evaluation of Capsule and comparison baselines on a collection of 39 real-world events. Capsule performs best.

Model	LL at 10 iterations	LL at convergence
Full Capsule	-1.62e7	-1.52e7
Entity Topics Only	-1.64e7	–
General Topics Only	-1.71e7	-1.53e7
Event Only	-1.79e7	–

Table 3.3: Log likelihood (LL) computed on validation data at 10 iterations and at convergence—the event only and entity only models are small enough that they converge with very few iterations. The full Capsule model achieves the lowest log likelihood in both cases.

its component parts; Table 3.3 shows that the full Capsule model captures the data better than any of its component parts individually.

Model Exploration. The evaluations to this point are useful in validating that Capsule captures its intended constructs, but the objective of the model is not prediction; rather, it is to be used as a scaffold to explore large collections of documents. We now turn to exploring the cables data using Capsule.

We begin our exploration by detecting events using Capsule. With Equation (3.2) as our metric of "eventness," we consider this metric over time, which is shown in Figure 3.1. Here, high values—often peaks—correspond to real-worlds events, several of which are labeled.

One of the tallest peak occurs the week of December 1, 1975, during which the United Nations General Assembly (UNGA) discussed omnibus decolonization. As discussed in

$f * \epsilon$	Date	Entity	Subject
4.60	1975-12-05	Canberra	30th UNGA: Item 23, Guam, Obmibus Decolonization and ...
4.26	1975-12-05	Mexico	30th UNGA-Item 23: Guam, Omnibus Decolonization and ...
4.21	1975-12-06	State	30th UNGA-Item 23: Guam, Omnibus Decolonization and ...
4.11	1975-12-03	Dakar	30th UNGA: Resolutions on American Samoa, Guam and ...
4.08	1975-12-04	Monrovia	30th UNGA: Item 23: Resolutions on decolonization and A...

Table 3.4: Top documents for the time interval of week December 1, 1975, when the UN discussed decolonization resolutions; Capsule recovers relevant documents related to this real-world event. Typos intentionally copied from original data.

$f * \epsilon$	Date	Entity	Subject
5.06	1975-05-15	Sofia	Seizure of US merchant vessel by Cambodian forces
5.05	1975-05-15	Dar es Salaam	Seizure of U.S. merchant vessel by Cambodian forces
4.92	1975-05-16	Lusaka	Seizure of US merchant vessel by Cambodian forces
4.61	1975-05-13	Zagreb	Waiver request for INS Vienna visas Eagle name check...
4.59	1975-05-15	State	eizure of US merchant Vessel by Cambodian forces

Table 3.5: Top documents for the week during which the S.S. Mayaguez was captured. Capsule identifies documents relevant to the real-world event. Typos intentionally copied from original data.

Section 3.2, we sort documents by their weighted event relevancy parameters $f(i_d, t)\epsilon_{d,t}$ to find cables that reflect an event. Table 3.4 shows the top cables for this discussion. Capsule accurately identifies this real-world event and recovers relevant cables.

Another notable event was the seizure of the S.S. Mayaguez, an American merchant vessel, in May of 1975—at the end of the Vietnam War. The top documents for this week are shown in Table 3.5. We can inspect individual documents to confirm their relevancy and learn more about the events. For instance, the content of the most relevant document, according to Capsule, is as follows.

```
In absence of MFA Chief of Eighth Department Avramov, I informed American
desk officer Yankov of circumstances surrounding seizure and recovery of merchant
ship Mayaguez and its crew. Yankov promised to inform the Foreign Minister of
US statement today (May 15). Batjer
```

$f * \epsilon$	Date	Entity	Subject
6.86	1976-07-07	Cairo	Possible SC meeting on Israeli rescue operation
6.18	1976-07-10	Kuwait	Media reaction to Bicentennial summary
6.15	1976-07-06	Damascus	Syria condemns Israeli operation to free Air France ...
5.91	1976-07-08	Tel Aviv	Passengers comment on Air France hijacking
5.89	1976-07-06	Stockholm	Possible SC meeting on Israeli rescue operation
5.38	1976-07-09	Nicosia	Bicentennial activities in Cyprus
5.09	1976-07-11	State	Security Council debate on Entebbe events CONFID...
4.77	1976-07-09	State	Travel of Peter M. Storm, House Budget Committee
4.76	1976-07-06	Jidda	Weekly Saudi Editorial Summary (June 30-July 6)
4.68	1976-07-08	Lusaka	SWAPO President seeks assessment of Kissinger-Vor...
4.56	1976-07-07	Stockholm	Ugandan role in Air France hijacking
4.45	1976-07-06	Karachi	Transitional quarter funding for RSS travel
4.43	1976-07-06	Athens	Bicentennial anniversary in Greece
4.37	1976-07-08	Damascus	Beirut travel
4.34	1976-07-10	State	Status of Mrs. Bloch
4.17	1976-07-07	Hong Kong	Hong Kong Communist press denounces Israeli resc...
4.12	1976-07-08	Dar es Salaam	President Nyerere's fourth of July messages
4.09	1976-07-10	Moscow	Pravda and Krasnaya Zvezda on Entebbe rescue oper...

Table 3.6: Top documents for the week after the US bicentennial celebration and Operation Entebbe. Capsule identifies documents relevant to both these real-world events.

A third week of interest occurs in early July of 1976. On July 4th, the US celebrated its Bicentennial, but on the same day, Israeli forces completed a hostage rescue mission—an Air France flight from Tel Aviv had been hijacked and taken to Entebbe, Uganda. This event, like many events, is mostly discussed the week following the real-world event; relevant cables are shown in Table 3.6. The cable from Stockholm describing the "Ugandan role in Air France hijacking" begins with the following content, which reveals further information about the event.

> 1. We provided MFA Director of Political Affairs Leifland with Evidence of Ugandan assistance to hijackers contained in Ref A. After reading material, Leifland described it a "quite good", and said it would be helpful for meeting MFA has scheduled for early this morning to determine position GOS will take at July 8 UNSC consideration of Israeli Rescue Operation. ...

top terms
church, vatican, catholic, bishop, pope, ford, cardinal, ban, religious, archbishop
program, university, grant, education, school, post, institute, research, center, american
security, council, terrorist, threat, sc, sabotage, protective, herein, unsc, honour
visit, hotel, schedule, arrival, arrive, depart, please, meet, day, room
labor, union, strike, ilo, employment, federation, afl cio, trade, worker, confederation
bank, credit, loan, investment, finance, payment, financial, eximbank, opic, central
law, case, court, legal, investigation, arrest, justice, sentence, trial, attorney
party, government, election, opposition, national, leader, campaign, vote, support, anti
tax, company, pay, lease, compensation, exemption, repatriation, income, taxation, fee
oil, petroleum, opec, crude, gulf, price, exploration, refinery, energy, company
israel, arab, israeli, middle, egypt, peace, plo, cairo, egyptian, lebanon
radio, television, broadcast, allotment, appropriation, obligation, zero, warc, transmitter, network
india, indian, pakistan, delhi, goi, ocean, bangladesh, transit, pakistani, afghan
turkish, turkey, cyprus, greek, greece, athens, ankara, morocco, cypriot, algeria
aid, relief, emergency, usaid, disaster, donor, wfp, sahel, ifad, unicef
aircraft, team, flight, clearance, transport, civair, aviation, traffic, charter, cargo
soviet, moscow, press, ussr, soviet union, american, one, war, communist, article
sea, zone, marine, maritime, fish, coastal, continental, territorial, mile, fishery

Table 3.7: Top vocabulary terms for a selection of general topics, one per row, according to topic distributions β_k. Capsule identifies general diplomatic themes that can be relevant to any entity.

Capsule assumes that only one event occurs in each time interval—this example is a clear violation of this assumption, but it also demonstrates that the model successfully captures both events, even when they overlap.

In addition to events, Capsule can be used to explore the general themes of a corpus and entities' typical concerns. Examples of general topics of conversation are shown in Table 3.7 and entity-exclusive topics are shown in Table 3.8; these show us how entity topics absorb location-specific words, preventing these terms from overwhelming the general topics.

These exploratory results show that our model is successfully capturing when multiple entities are discussing the same subjects and that our model can be used to explore the underlying data by providing a structured scaffold from which to view the data.

entity	top terms
Ankara	turkish, turkey, ankara, government, cyprus, greek, party, one, time
Athens	greek, athens, greece, gog, government, cyprus, turkish, press, minister
Auckland	new zealand, company, box, trade, contact, opportunity, united states
Baghdad	iraqi, iraq, goi, arab, state, regime, ministry, government, party
Berlin	berlin, frg, german, senat, time, bonn, trade, one, agreement
Bern	swiss, bern, federal, bank, snb, gold, end, interest, national
Brussels	belgian, belgium, brussels, government, firestone, european, ministry
Budapest	hungarian, hungary, trade, mudd, one, time, puja, well, policy
Buenos Aires	argentine, argentina, goa, us, hill, government, one, press, police
Cairo	egyptian, cairo, egypt, arab, israeli, israel, peace, agreement, president
Canberra	australian, australia, goa, government, minister, whitlam, end, dfa, time
Dakar	senegalese, president, african, summary, conference, end, support, one
Dar es Salaam	tangov, salaam, tanzanian, spain, president, government, african, one
Guayaquil	ecuador, ecuadorean, port, congen, one, tuna, local, time, boat
Islamabad	pakistan, gop, government, one, party, minister, general, opposition, ppp
Paris	paris, france, rush, french, one, government, amconsul, quai, european
Jerusalem	jerusalem, bank, israeli, us, israel, plo, one, arab, unifil
Jidda	saudi, jidda, saudi arabia, prince, us, fahd, one, time, government
Johannesburg	black, africa, african, trade, union, police, labor, one, committee
Kabul	afghan, government, goa, minister, one, pakistan, regime, time, ministry
Lima	peru, gop, lima, peruvian, dean, minister, general, marcona, government
Lisbon	portugal, portuguese, gop, lisbon, government, party, summary, minister
London	london, british, government, fco, labor, agreement, one, washdc, summary
Madrid	spanish, spain, madrid, one, govt, general, committee, government, time
Nairobi	kenya, nairobi, marshall, embassy, kenyan, unep, le, ref, state
Oslo	norwegian, norway, soviet, government, minister, ministry, policy
Ottawa	canadian, canada, goc, ottawa, us, extaff, government, minister, federal
Peking	chinese, peking, uslo, china, people, teng, one, trade, delegation, hong
Phnom penh	penh, phnom, khmer, rice, fank, enemy, cambodia, government, dean
Prague	czechoslovak, goc, czech, trade, embassy, one, mfa, time, cssr
Quito	ecuador, ecuadorean, gulf, government, minister, bloomfield, general, one
Sao Paulo	paulo, brazil, state, brazilian, president, government, congen, one, do
Seoul	korea, korean, rok, rokg, seoul, park, government, president, time
Singapore	singapore, asean, minister, government, one, prime, comment, vietnam
Sofia	bulgarian, trade, one, agreement, american, visit, committee, party
Sydney	australia, australian, one, general, american, state, government, post
Tokyo	japan, japanese, tokyo, fonoff, summary, miti, end, diet, time
Taipei	taiwan, groc, china, chinese, government, american, one, local, republic
The Hague	dutch, netherlands, hague, government, minister, party, stoel, mfa, one
USUN New York	committee, usun, priority, report, draft, resolution, sc, comite, rep, new york
Vancouver	canada, government, canadian, british, columbia, pipeline, federal, editorial
Zagreb	yugoslav, yugoslavia, croatian, fair, belgrade, american, one, ina, summary
Zurich	swiss, congen, consulate, general, american, bern, dollar, shipment

Table 3.8: Top vocabulary terms for a selection of entities according to entity-exclusive topics η_n. Capsule identifies entity-specific themes and interests.

3.5 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 comparison methods and explored its results on a real-world datasets. We anticipate that Capsule and its visualization (presented in Chapter 5) can be used by historians, political scientists, and others who wish to investigate events in large text corpora.

4 Social Poisson Factorization

People exercise an unconscious selection

in being influenced.

– T. S. Eliot

Until this point, we have focused on the analysis of human behavior based on observations of text. We turn now to investigate a different variety of human behavior: individuals interacting with media or purchasing items. We consider the problem of how best to recommend content to consumers, which obviously has applications in advertising, but is also relevant to economics, psychology, and sociology.

Recommendation has become a core component in our online experience, such as when we watch movies, read articles, listen to music, and shop. Given information about what a user has consumed (e.g., items viewed, marked as "favorites," or rated), the goal of recommendation is to suggest a set of unobserved items that she will like.

Most recommendation systems aim to make personalized suggestions to each user based on similar users' histories. To solve this problem, matrix factorization algorithms are the workhorse methods of choice to solve this problem (Koren et al., 2009; Su and Khoshgoftaar, 2009). Factorization algorithms use historical data to uncover recurring patterns of consumption, and then describe each user in terms of their varying preferences for those patterns. For example, the discovered patterns might include art supplies, holiday decora-

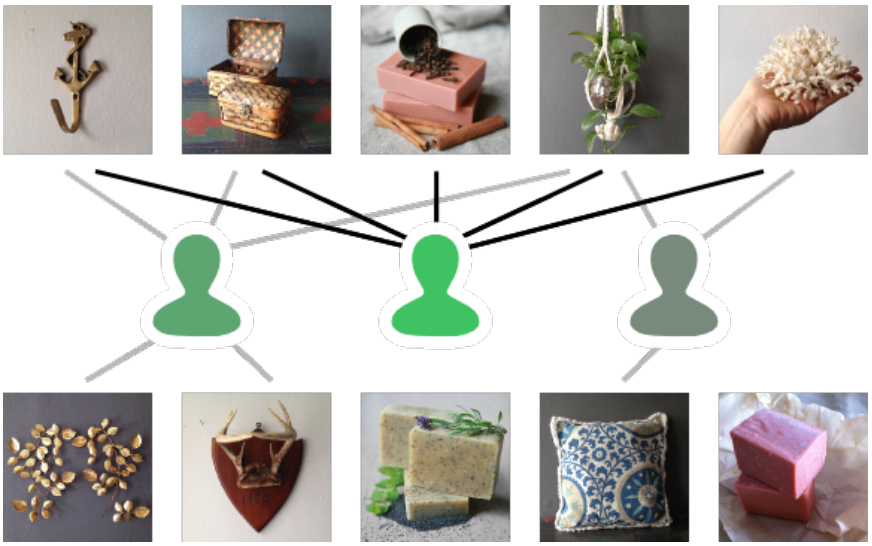


Figure 4.1: Observed and recommended items[1] for an Etsy user. The user is shown in the center, with friends on the sides. The top row is training items and the bottom row is the top recommendations from our model (SPF). Some items are recommended because they are favorites of the friends, and others because they match the general preferences of the user.

tions, and vintage kitchenware; and each user has different preferences for each category. To perform recommendation, factorization algorithms find unmarked items of each user that are characteristic of her preferences.

Many applications of recommendation contain an additional source of information: a social network. This network is increasingly available at the same platforms on which we read, watch, and shop. Examples include Etsy, Instagram, and various social readers. Researchers have found that users value the opinions of their friends for discovering and discussing content (Johnstone and Katz, 1957; Volz, 2006), and online access to their network can reinforce this phenomenon.

Factorization approaches, however, cannot exploit this information. They can capture that you may enjoy an item because it matches your general preferences, but they cannot capture that you may enjoy another because your friend enjoyed it. Knowing your connections and what items your friends like should help better predict what you will enjoy.

[1] Etsy product images courtesy of Amber Dubois and Ami Lahoff. Used with permission.

In this chapter we develop *social Poisson factorization* (SPF), a Bayesian factorization method that accounts for the social aspect of how users consume items. (SPF is based on Poisson factorization (Section 2.4), a model that is particularly suited for implicit data.) SPF assumes that there are two signals driving each user's clicks: her latent preferences for items (and the latent attributes of each) and the latent "influence" of her friends.[2] From observed data—which contains both click histories and a social network—SPF infers each user's preferences and influences. Subsequently, it recommends items relating both to what a user is likely to be interested in and what her friends have clicked.

Figure 4.1 gives the intuition. The user is in the center. She clicked on items (on the top, connected to the user), has friends (to either side), and those friends have clicked on items too (top and bottom, connected to each friend). From this data, we can learn both about her preferences (e.g., for handmade soap) and about how much she is influenced by each of her friends (e.g., more strongly by the friend on the left). SPF recommends items on the bottom, based on both aspects of the data. It is important to be able to explain the origins of recommendations to users (Herlocker et al., 2000), and SPF can tell the user why an item was recommended: it can indicate friends ("you always trust Sally") and general item attributes ("you seem to like everything about ninjas") to describe the source of recommendations.

We use the language of users clicking on items. This is just a convenience—our model applies just as easily for users purchasing, rating, watching, reading, and marking as a favorite. Our goal is to predict which of the unclicked items a user will want to click.

In this chapter, we develop the mathematical details behind the model (Section 4.2), derive efficient learning algorithms (based on variational inference) for estimating it from data (Section 4.3), and evaluate it on six real-world data sets (Section 4.4). In all cases, our social

[2]There is a large body of research literature on peer influence (Leskovec et al., 2006; Crandall et al., 2008; Shang et al., 2011). In this work we use the term to indicate the latent change in consumption due to social connections.

recommendation outperformed both traditional factorization approaches (Gopalan et al., 2015; Salakhutdinov and Mnih, 2007) and previous recommendation methods that account for the network (Guo et al., 2015; Jamali and Ester, 2010; Ma et al., 2009, 2008; Yang et al., 2013).

4.1 Related Work

We first review previous research on using social networks to help recommend items to users. A crucial component of SPF is that it infers the influence that users have with each other. In previous work, some systems assume that user influence (sometimes called "trust") is observed (Massa and Avesani, 2007). However, trust information beyond a binary yes/no is onerous for users to input, and thus observing trust beyond "following" or "friending" is impractical in a large system. Others assume that trust is propagated (Andersen et al., 2008) or computed from the structure of the network (Golbeck and Hendler, 2006). This is limited in that it ignores user activity, which can reveal the trust of a user for some parts of the network over others; SPF captures this idea. Information diffusion (Du et al., 2013; Guille et al., 2013) also relies on user activity to describe influence, but focuses on understanding the spread of information in a more global sense than we desire. A final alternative is to compute trust from rating similarities between users (Fazeli et al., 2014). However, performing this computation in advance of fitting the model confounds general preference similarity with instances of influence—two people with the same preferences might read the same books in isolation.

Other research has included social information directly into various collaborative filtering methods. Zhao et al. (2014) incorporate the network into pairwise ranking methods. Their approach is interesting, but one-class ranking methods are not as interpretable as factorization, which is important in many applications of recommender systems (Herlocker et al., 2000).

Ma et al. (2008), Purushotham et al. (2012), and Yang et al. (2013) have explored how traditional factorization methods can exploit network connections. For example, many of these models factorize both user-item data and the user-user network. This brings the latent preferences of connected users closer to each other, reflecting that friends have similar tastes. Ma et al. (2009) and Ye et al. (2012) incorporate this idea more directly by including friends' latent representations in computing recommendations made for a user.

Our model has a fundamentally different approach to using the network to form recommendations. It seeks to find friends with different preferences to help recommend items to a user that are outside of her usual taste. For example, imagine that a user likes an item simply because many of her friends liked it too, but that it falls squarely outside of her usual preferences. Models that adjust their friends' overall preferences according to the social network do not allow the possibility that the user may still enjoy this anomalous item. As we show in Section 4.4, using the social network in this way performs better than these previous approaches.

4.2 Social Poisson Factorization

In this section we develop social Poisson factorization (SPF). SPF is a model for recommendation; it captures patterns in user activity using traditional signals—latent user preferences and latent item attributes—and estimates how much each user is influenced by his or her friends' observed clicks. From its estimate of influence, SPF recommends clicked items by influential friends even when they are not consistent with a user's factorization-based preferences.

We first review Poisson factorization in the context of recommendation systems and give the intuition on our model. Then, we formally specify our model, describe how to form recommendations, and discuss how we learn the hidden variables.

4.2.1 PF for Recommendation

SPF is based on Poisson factorization (PF) (Gopalan et al., 2015), a variant of probabilistic matrix factorization for recommendation. Section 2.4 describes the model in detail. Here, we review PF with an eye towards recommendation.

Let r_{ui} be the count of how many times user u clicked item i.[3] PF assumes that an observed count r_{ui} comes from a Poisson distribution. Its rate is a linear combination of a non-negative K-vector of user preferences θ_u and a non-negative K-vector of item attributes β_i,

$$r_{ui} \sim \text{Poisson}(\theta_u^\top \beta_i).$$

The user preferences and item attributes are hidden variables with Gamma priors. (Recall that the Gamma is an exponential family distribution of positive values.) Given a matrix of observed clicks, posterior inference of these hidden variables reveals a useful factorization: latent attributes describe each item and latent preference describe each user. These inferences enable personalized recommendations.

4.2.2 SPF Intuition

In many settings, users are part of an online social network that is connected to the same platforms on which they engage with items. For some, such as Etsy, these networks are innate to the site. Others may have external data, e.g., from Facebook or LinkedIn, about the network of users.

We build on PF to develop a model of data where users click on items and where the same users are organized in a network. Social Poisson factorization (SPF) accounts for both the latent preferences of each user and the click patterns of her neighbors.

[3] The theory around PF works on count data, but Gopalan et al. (2015) showed that it works well empirically with implicit recommendation data, i.e., censored counts, as well.

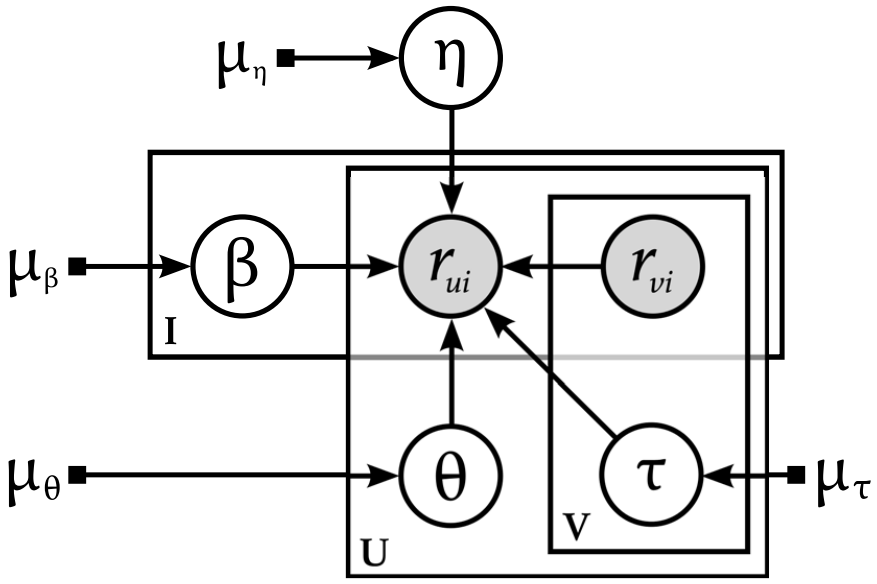


Figure 4.2: A conditional directed graphical model of social Poisson Factorization (SPF) to show considered dependencies. For brevity, we refer to the set of priors a and b as μ; for example, $\mu_\theta = (a_\theta, b_\theta)$. These hyper-parameters are fixed.

Consider the user whose items are shown in Figure 4.1. The intuition behind SPF is that there can be two reasons that a user might like an item. The first reason is that the user's general preferences match with the attributes of the item; this is the idea behind Poisson factorization (and other factorization approaches). For example, the user of Figure 4.1 may inherently enjoy handmade soap. A second reason is that the user has a friend who likes the item, or perhaps a collection of friends who all like it. This possibility is not exposed by factorization, but captures how the user might find items that are outside of her general preferences. Without learning the influence of friends in Figure 4.1, the system could easily interpret the basket as a general preference and recommend more baskets, even if the user does not usually like them.

SPF captures this intuition. As in PF, each user has a vector of latent preferences. However, each user also has a vector of "influence" values, one for each of her friends. Whether she likes an item depends on both signals: first, it depends on the affinity between her latent preferences and the item's latent attributes; second, it depends on whether her influential friends have clicked it.

4.2.3 Model Specification

We formally describe SPF. The observed data are user behavior and a social network. The behavior data is a sparse matrix $\mathbf{R}$, where r_{ui} is the number of times user u clicked on item i. (Often this will be one or zero.) The social network is represented by its neighbor sets; $N(u)$ is the set of indices of other users connected to u. Finally, the hidden variables of SPF are per-user K-vectors of non-negative preferences θ_u, per-item K-vectors of non-negative attributes β_i, and per-neighbor non-negative user influences τ_{uv}. Loosely, τ_{uv} represents how much user u is influenced by the clicks of her neighbor, user v. (Note we must set the number of components K. Section 4.4 studies the effect of K on performance; usually we set it to 50 or 100.)

Conditional on the hidden variables and the social network, SPF is a model of clicks r_{ui}. Unlike many models in modern machine learning, we specify the joint distribution of the entire matrix $\mathbf{R}$ by the conditionals of each cell r_{ui} given the others,

$$r_{ui} \mid r_{-u,i} \sim \text{Poisson}\left(\theta_u^\top \beta_i + \textstyle\sum_{v \in N(u)} \tau_{uv} r_{vi}\right), \tag{4.1}$$

where $r_{-u,i}$ denotes the vector of clicks of the other users of the ith item.[4] This equation captures the intuition behind the model, that the conditional distribution of whether user u clicks on item i is governed by two terms. The first term, as we said above, is the affinity between latent preferences θ_u and latent attributes β_i; the second term bumps the parameter up when trustworthy neighbors v (i.e., those with high values of τ_{uv}) also clicked on the item. Figure 4.2 shows the dependencies between the hidden and observed variables as a conditional graphical model.

To complete the specification of the variables, we place gamma priors on all of the hidden

[4] We are specifying an exponential family model conditionally, as described in Section 2.1.3. Here we have a improper conditional model with the specification defining a pseudo-likelihood (Besag, 1975).

variables. We chose the hyper-parameters of the gammas so that preferences, attributes, and influences are sparse. (See Section 4.4 for details.)

4.2.4 Forming Recommendations with SPF

We have specified a probabilistic model of hidden variables and observed clicks. Given a $U \times M$ click matrix $\mathbf{R}$ and a $U \times U$ social network $\mathbf{N}$, we analyze the data by estimating the posterior distribution of the hidden preferences, attributes, and influences $p(\theta_{1:U}, \beta_{1:M}, \tau_{1:U} \mid \mathbf{R}, \mathbf{N})$. The following section describes our algorithm for estimating this posterior, which places high probability on configurations of preferences, attributes, and influence values that best describe the observed clicks within the social network.

From this posterior, we can form predictions for each user and each of their unclicked items. For a user u and an unclicked item j, we compute

$$\mathbb{E}\left[r_{uj}\right] = \mathbb{E}\left[\theta_u\right]^\top \mathbb{E}\left[\beta_j\right] + \sum_{v \in N(u)} \mathbb{E}\left[\tau_{uv}\right] r_{vj}, \tag{4.2}$$

where all expectations are with respect to the posterior. For each user, we form recommendation lists by making predictions for the user's set of unclicked items and then ranking the items by these continuous-valued predictions. This is how we can use SPF to form a recommendation system.

4.3 Variational Inference for SPF

Social PF enjoys the benefits of Poisson factorization and accounts for the network of users. However, using SPF requires computing the posterior. Conditioned on click data and a social network, our goal is to compute the posterior user preferences, item attributes, and latent influence values.

As we have encountered previously, the exact posterior for SPF is not tractable to compute and we approximate the posterior based on variational methods (see Section 2.2). With our algorithm, we can approximate posterior expectations with very large click and network data (see Section 4.4).

Like in previous chapters, we use the mean-field variational family, where each latent variable is independent and governed by its own variational parameter. The latent variables are the user preferences θ_u, item attributes β_i, and user influences τ_{uv}. The variational family is

$$q(\theta, \beta, \tau) = \prod_{u,k} q(\theta_{uk}|\lambda^{\theta}_{uk}) \prod_{i,k} q(\beta_{ik}|\lambda^{\beta}_{ik}) \prod_{u,v} q(\tau_{uv}|\lambda^{\tau}_{uv}). \tag{4.3}$$

This is a flexible family. For example each cell of each user's preference vector θ_{uk} is associated with its own variational parameter λ^{θ}_{uk}. Thus, when fit to be close to the model's posterior, the variational parameters can capture each user's unique interests, each item's unique attributes, and each friend's unique influence value.

With the family in place, variational inference solves the following optimization problem,

$$q^*(\theta, \beta, \tau) = \arg\min_q \mathrm{KL}\left(q(\theta, \beta, \tau)||p(\theta, \beta, \tau \mid \mathbf{R}, \mathbf{N})\right) \tag{4.4}$$

Note that the data—the clicks and the network—enter the variational distribution through this optimization. Finally, we use the resulting variational parameters of $q^*(\cdot)$ as a proxy for the exact posterior. This lets us use SPF to perform recommendation.

Our algorithm fits the parameters of the variational distribution in Equation (4.3) so that it is close in KL divergence to the posterior. We use coordinate ascent, iteratively updating each parameter while holding the others fixed. This goes uphill in the variational objective and converges to a local optimum (Bishop, 2006).

To obtain simple updates, we first construct auxiliary latent variables z using Property 2.4.1;

we apply this decomposition to the conditional click count distribution in Equation (4.1). We define Poisson variables for each term in the click count:

$$z^M_{uik} \sim \text{Poisson}(\theta_{uk}\beta_{ik}) \quad z^S_{uiv} \sim \text{Poisson}\left(\tau_{uv} r_{vi}\right).$$

The M and S superscripts indicate the contributions from matrix factorization (general preferences) and social factorization (influence), respectively. Given these variables, the click count is deterministic,

$$r_{ui} \mid r_{-u,i} = \sum_{k=1}^{K} z^M_{uik} + \sum_{v=1}^{V} z^S_{uiv},$$

where $V = |N(u)|$ and the index v selects a friend of u (as opposed to selecting from the set of all users).

We now require the complete conditional distributions for each variable; these conditionals define both the form of each variational factor and their updates. For the Gamma variables—the user preferences, item attributes, and user influence—the conditionals are

$$\theta_{uk} \mid \beta, \tau, z, \mathbf{R}, \mathbf{N} \sim \text{Gamma}\left(a_\theta + \sum_i z^M_{uik},\ b_\theta + \sum_i \beta_{ik}\right) \tag{4.5}$$

$$\beta_{ik} \mid \theta, \tau, z, \mathbf{R}, \mathbf{N} \sim \text{Gamma}\left(a_\beta + \sum_u z^M_{uik},\ b_\beta + \sum_u \theta_{uk}\right) \tag{4.6}$$

$$\tau_{uv} \mid \theta, \beta, z, \mathbf{R}, \mathbf{N} \sim \text{Gamma}\left(a_\tau + \sum_i z^S_{uiv},\ b_\tau + \sum_i r_{vi}\right). \tag{4.7}$$

The complete conditional for the auxiliary variables is
$z_{ui} \mid \theta, \beta, \tau, \mathbf{R}, \mathbf{N} \sim \text{Mult}\left(r_{ui}, \phi_{ui}\right)$ where

$$\phi_{ui} \propto \left\langle \theta_{u1}\beta_{i1},\ \cdots,\ \theta_{uK}\beta_{iK}, \tau_{u1} r_{1i},\ \cdots,\ \tau_{uV} r_{Vi} \right\rangle. \tag{4.8}$$

(Intuitively, these variables allocate the data to one of the factors or one of the friends.) Each

variational factor is set to the same family as its corresponding complete conditional.

Given these conditionals, the algorithm sets each parameter to the expected conditional parameter under the variational distribution. (Thanks to the mean field assumption, this expectation will not involve the parameter being updated.) Note that under a gamma distribution, $\mathbb{E}[\lambda] = \lambda_a/\lambda_b$, where λ_a and λ_b are shape and rate parameters. For the auxiliary variables, the expectation of the indicator is the probability, $\mathbb{E}[z_{ui}] = r_{ui} * \phi_{ui}$.

Algorithm 5 shows our variational inference algorithm. It is $O(N(K + V))$ per iteration, where N is the number of recorded user-item interactions (click counts, ratings, etc.). K is the number of latent factors, and V is the maximum user degree. (Note that both K and V are usually small relative to N.) In Section 4.4 we empirically compare the runtime of SPF with competing methods. We can modify the algorithm to sample users and update the variables stochastically (Hoffman et al., 2013); this approach scales to much larger datasets than competing methods.

To assess convergence, we use the change in the average click log likelihood of a validation set.

Source code for Algorithm 5 is available at `https://github.com/ajbc/spf`. We now turn to an empirical study of SPF.

4.4 Empirical Study

In this section we study the performance of SPF. We compared SPF to five competing methods that involve a social network in recommendation (Guo et al., 2015; Jamali and Ester, 2010; Ma et al., 2009, 2008; Yang et al., 2013) as well as two traditional factorization approaches (Gopalan et al., 2015; Salakhutdinov and Mnih, 2007). Across six real-world datasets, our methods outperformed all of the competing methods (Figure 4.3). We also demonstrate how to use SPF to explore the data, characterizing it in terms of latent factors

Algorithm 5: Mean field variational inference SPF

Initialize $\mathbb{E}[\theta]$, $\mathbb{E}[\beta]$ randomly
for *each user* u **do**
  for *each friend* $v \in N(u)$ **do**
    Set $\lambda^{\tau,b}_{u,v} =$ prior $b_\tau + \sum_i r_{vi}$ [eq. (4.7)]
  end
end
while *Model has not converged,* $\Delta \log \mathcal{L} > \delta$ **do**
  Initialize global $\lambda^{\beta,a}$ to prior a_β for all items and all factors
  for *each user* u **do**
    while *User parameters have not converged,* $\Delta[\theta_u] + \Delta[\tau_u] > \delta'$ **do**
      Initialize local $\lambda^{\beta,a}$ to 0 for all items and factors **Initialize** preferences $\lambda^{\theta,a}_u$ to prior a_θ for all factors **Set** $\lambda^{\theta,b}_u =$ prior $b_\theta + \sum_i \mathbb{E}[\beta_i]$ [eq. (4.5)]
      Initialize influence $\lambda^{\tau,a}_{user}$ to prior a_τ for all friends
      for *each (item* i*, click count* r*)* $\in clicks_u$ **do**
        Set ϕ_{ui} from $\mathbb{E}[\theta_u]$, $\mathbb{E}[\beta_i]$, $\mathbb{E}[\tau_u]$, and r_i [eq. (4.8)]
        Set $\mathbb{E}[z_{ui}] = r * \phi_{ui}$
        Update $\lambda^{\theta,a}_u \mathrel{+}= \mathbb{E}[z^M_{ui}]$ [eq. (4.5)]
        Update $\lambda^{\tau,a}_u \mathrel{+}= \mathbb{E}[z^S_{ui}]$ [eq. (4.7)]
        Update local $\lambda^{\beta,a}_i \mathrel{+}= \mathbb{E}[z^M_{ui}]$ [eq. (4.6)]
      end
      Set $\mathbb{E}[\theta_u] = \lambda^{\theta,a}_u / \lambda^{\theta,b}_u$
      Set $\mathbb{E}[\tau_u] = \lambda^{\tau,a}_u / \lambda^{\tau,b}_u$
    end
    Update global $\lambda^{\beta,a} \mathrel{+}=$ local $\lambda^{\beta,a}$
  end
  Set $\lambda^{\beta,b} =$ prior $b_\beta + \sum_u \mathbb{E}[\theta_u]$ [eq. (4.6)]
  Set $\mathbb{E}[\beta] = \lambda^{\beta,a} / \lambda^{\beta,b}$
end

and social influence. Finally, we assess sensitivity to the number of latent factors and discuss how to set hyper-parameters on the prior distributions.

4.4.1 Datasets, Methods, and Metrics

Datasets and preprocessing. We studied six datasets. Table 4.1 summarizes their attributes. The datasets are:

- *Ciao* (ciao.co.uk) is a consumer review website with an underlying social network.

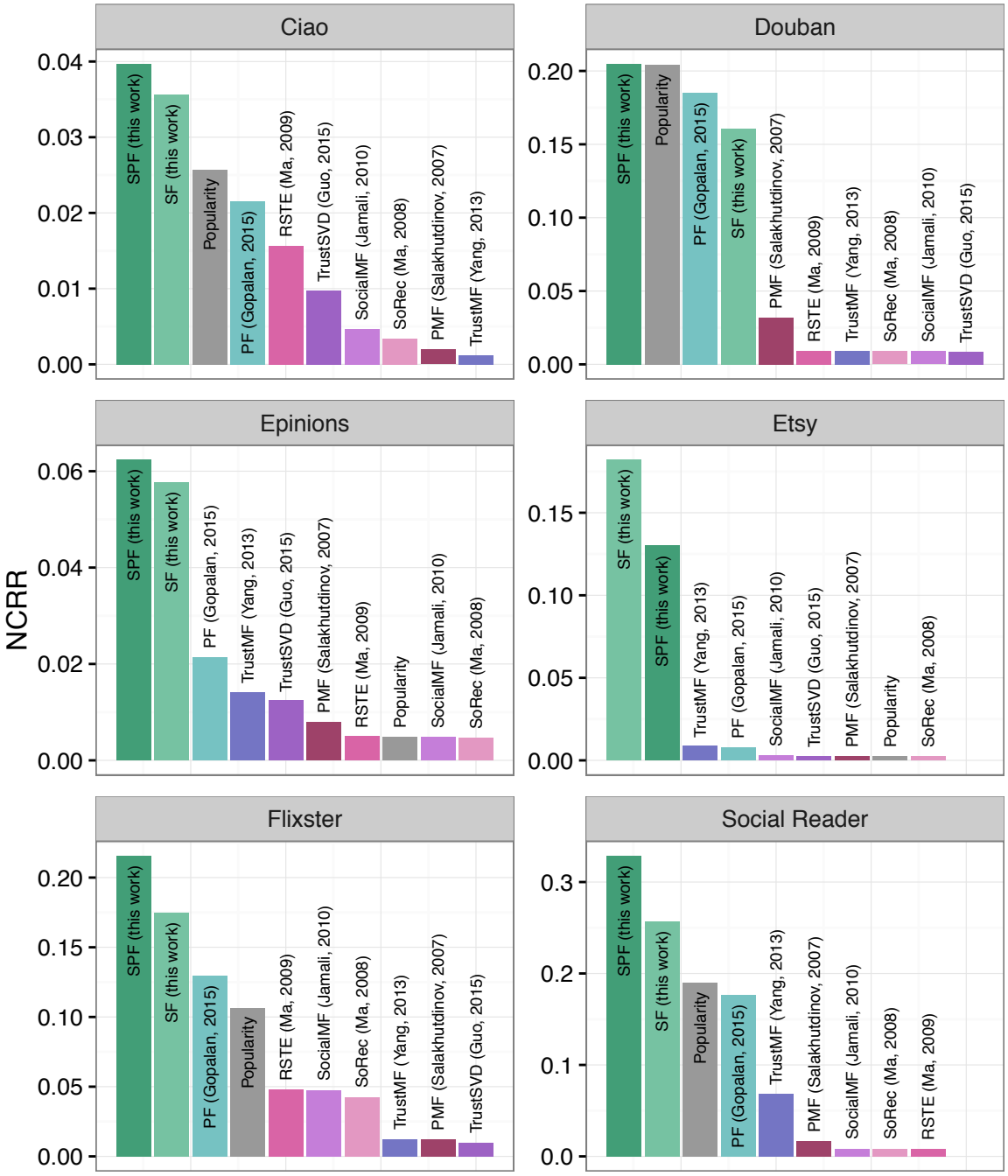


Figure 4.3: Performance of various methods on all six datasets, measured as NCRR averaged over users with held-out data. The Poisson-based factor models (PF and SPF) use $K = 40$ on Ciao, $K = 125$ on Epinions, $K = 100$ on Etsy, and $K = 50$ on Flixster, Douban, and Social Reader. Similar K values are used for competing models, but some perform best with lower K, in which case those settings are used. Models are sorted by performance. RSTE was omitted on Etsy data due to long run time and TrustSVD was omitted on Social Reader data due to difficulty in finding appropriate parameter settings. SPF outperforms all competing methods, except on Etsy, where our alternate model SF achieves top performance.

Guo et al. (2014) crawled DVD ratings and trust values for a small dataset of 7K users and 98K items.

- *Epinions* (epinions.com) is another consumer reviews website where users rate items and mark users as trustworthy. Our data source was Massa and Avesani (2007) and consists of 39K users and 131K items.

- *Flixster* (flixster.com) is a social movie review website crawled by Jamali and Ester (2010). We binarized ratings, thresholding at 3 or above, resulting in 132K users and 42K items.

- *Douban* (douban.com) is a Chinese social service where users record ratings for music, movies, and books; it was crawled by Ma et al. (2011). It contains 129K users and 57K items.

- *Etsy* (etsy.com) is a marketplace for handmade and vintage items, as well as art and craft supplies. Users may follow each other and mark items as favorites. This data was provided directly by Etsy, and culled to users who have favorited at least 10 items and have at least 25% of their items in common with their friends; we omitted any items with fewer than 5 favorites. This is a large dataset of 40K users and 5.2M items.

- *Social Reader* is a dataset from a large media company that deployed a reader application on a popular online social network. The data contains a friendship network and a table of article clicks. We analyzed data from April 2-6, 2012, only including users who read at least 3 articles during that time. It contains 122K users and 6K items.

These datasets include both explicit ratings on a star scale and binary data. Content consumption is binary when the data is implicit (a news article was viewed) or when the system only provides a binary flag (favoriting). With implicit data, non-Poisson models require us to subsample 0's so as to differentiate between items; in these instances, we randomly sampled negative examples such that each user has the same number of positive and negative ratings.

	Ciao	Epinions	Flixster	Douban	S. Reader	Etsy
# of users	7,375	39,307	131,542	129,097	121,950	39,862
# of items	97,540	130,786	41,878	56,862	6,153	5,201,879
# interactions	270,427	639,775	6,740,332	16,207,151	489,735	18,650,632
% interactions	0.038%	0.012%	0.122%	0.221%	0.065%	0.009%
interaction type	5-star	5-star	binary	5-star	binary	binary
network type	directed	directed	undirected	undirected	undirected	directed
# network edges	56,267	176,337	488,869	1,323,828	100,175	4,761,437
network density	0.103%	0.011%	0.006%	0.016%	0.001%	0.300%
% shared	25.0%	36.0%	62.3%	51.0%	50.1%	30.8%

Table 4.1: Attributes of each data source, post-curation. User-item interactions are non-zero clicks, favorites, or ratings. Percent shared is the average percentage of items users have in common with their friends. Data sources were chosen for their diversity of attributes.

Note that Poisson-based models implicitly analyze the full matrix without needing to pay the computational cost of analyzing the zeros (Gopalan et al., 2015).

For each dataset, we preprocessed the network. We removed network connections where the users have no items in common. Note this advantages both SPF and comparison models (though SPF can learn the relative influence of the neighbors).

Our studies divided the data into three groups: approximately 10% of 1000 users' data are held-out for post-inference testing, 1% of all users' data are used to assess convergence of the inference algorithm, and the rest is used to train. One exception is Ciao, where we used 10% of all users' data to test.

Competing methods. We compared SPF to five competing models that involve a social network in recommendation: RSTE (Ma et al., 2009), TrustSVD (Guo et al., 2015), SocialMF (Jamali and Ester, 2010), SoRec (Ma et al., 2008), and TrustMF (Yang et al., 2013).[5] We also include probabilistic Gaussian matrix factorization (PMF) (Salakhutdinov and Mnih, 2007), because it is a widely used recommendation method. For each of these, we used the parameter settings that achieved best performance according to the example fits published on the LibRec website.

[5]We used the LibRec library (librec.net) for all competing methods.

We can think of SPF having two parts: a Poisson factorization component and a social component (see Equation (4.1)). Thus we also compared SPF to each of these components in isolation, Poisson factorization (Gopalan et al., 2015) (PF) and *social factorization* (SF). SF is the influence model without the factorization model.[6] We note that SF is a contribution of this work as well.

Finally we compare to two baselines, ordering items randomly and ordering items by their universal popularity.

Metrics. We evaluate these methods on a per-user basis. For each user, we predict clicks for both held-out and truly unclicked items, and we rank these items according to their predictions. We denote the user-specific rank to be rank_{ui} for item i and user u. A better model will place the held-out items higher in the ranking (giving smaller rank_{ui} values on held-out items). We now introduce the *normalized cumulative reciprocal rank* (NCRR) metric to gauge this performance.

Reciprocal rank (RR) is an information retrieval measure; given a query, it is the reciprocal of the rank at which the first relevant document was retrieved. (Larger numbers are better.) Users "query" a recommender system similarly, except that each user only has one query (e.g., "what books should I read?") and they care not just about the first item that's relevant, but about finding as many relevant items as possible.

Suppose user u has held out items $\mathcal{D}_u$.[7] We define the cumulative reciprocal rank to be:

$$\text{CRR}_u = \sum_{i \in \mathcal{D}_u} \frac{1}{\text{rank}_{ui}}.$$

CRR can be interpreted as the ease of finding all held-out items, as higher numbers indicate

[6] Social factorization has a technical problem when none of a user's friends has clicked on an item; the resulting Poisson cannot have a rate of zero. Thus we add a small constant $\epsilon = 10^{-10}$ to the rate in social factorization's model of clicks.

[7] With binary data this is simply the full set of heldout items. When items have non-binary ratings, we threshold the set such to include only highly rated items (4 or 5 in a 5-star system).

that the held-out items are higher in the list. For example, a CRR of 0.75 means that the second and fourth items are in the held-out set, or are relevant to the user.

CRR behaves similarly to discounted cumulative gain (DCG), except it places a higher priority on high-rank items by omitting the log factor—it can be thought of as a harsher variant of DCG. Like DCG, it can be also be normalized. The normalized cumulative reciprocal rank (NCRR) is

$$\text{NCRR}_u = \frac{\text{CRR}_u}{\text{ideal CRR}_u},$$

where the ideal variant in the denominator is the value of the metric if the ranking was perfect. To evaluate an entire model, we can compute average NCRR over all users, $\frac{1}{U}\sum_u \text{NCRR}_u$. We will use this metric throughout this section.

Performance measured by NCRR is consistent with performance measured by NDCG, but NCRR is more interpretable. Simple reciprocals are easier to understand than the reciprocal of the log.

Note we omit root-mean-square error (RMSE) as a metric. Improvements in RMSE often do not translate into accuracy improvements for ranked lists (Amatriain et al., 2012; Cremonesi et al., 2010; Loiacono et al., 2014; Singh et al., 2014), especially with binary or implicit data. Our end goal here is item recommendation and not rating prediction—"which movie should I watch next?" is inherently a ranking problem—thus we treat the predictions as means to an end.

4.4.2 Performance and Exploration

We evaluate SPF by considering overall performance and performance as a function of user degree. We also show how to explore the data using the algorithm.

Performance. Figure 4.3 shows the performance of SPF against the competing methods: the previous methods that account for the social network, social factorization (SF), Poisson

factorization (PF), and the popularity baseline. (We do not illustrate the random baseline because it is far below all of the other methods.) SPF achieves top performance on five of the datasets. On the one remaining dataset, Etsy, the social-only variant of our model (SF) performs best.

Notice the strong performance of ranking by popularity. This highlights the importance of social factorization. It is only social Poisson factorization that consistently outperforms this baseline.

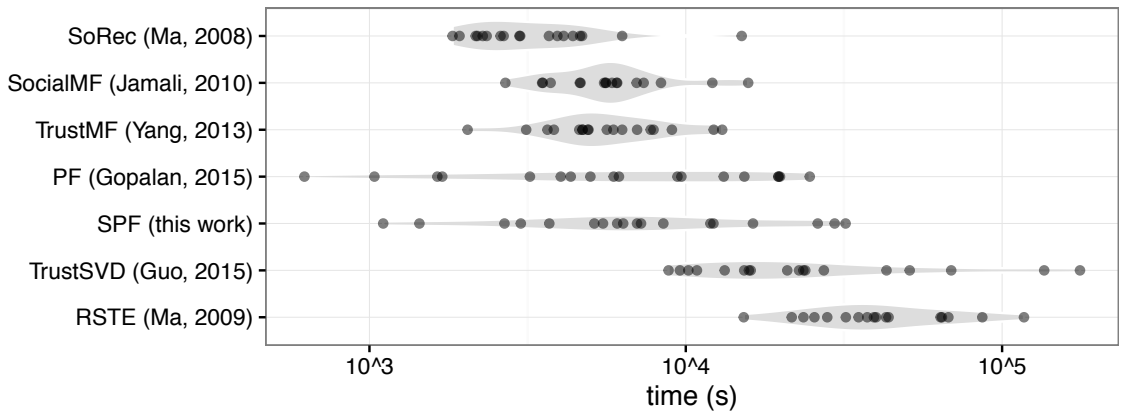


Figure 4.4: Training and testing runtimes for multiple models on Ciao data, with the number of latent factors K ranging from 1 to 500. Each dot represents a full cycle of training and evaluating. SPF performs with average runtime.

We measured runtime with the Ciao data set to get a sense for the relative computational costs. Figure 4.4 shows the runtime for all of the methods at various values of K. The Poisson models are average in terms of runtime.

Finally, using the Ciao and Epinions data, we break down the performance of SPF, SF, and PF as a function of the degree of each user; the results are shown in Figure 4.5.[8] All models perform better on high-degree users, presumably because these are higher activity users as well. Overall, SPF performs better than SF because of its advantage on the large number of low-degree users.

Interpretability. It is important to be able to explain the origins of recommendations to

[8]Smoothed with GAM. http://www.inside-r.org/r-doc/mgcv/gam

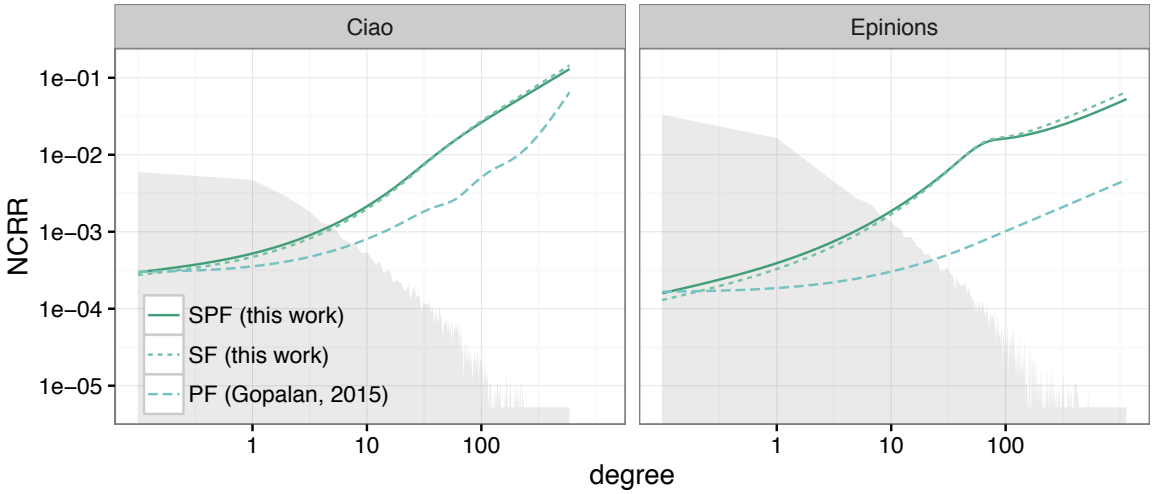


Figure 4.5: Performance on Ciao and Epinions broken down as a function of degree; grey in background indicates density of users. SPF and SF perform similarly, with SPF doing slightly better on a large number of low-degree users and SF doing better on a low number of high-degree users.

users (Herlocker et al., 2000). Items recommended with SPF have the advantage of interpretability. In particular, we use auxiliary variables (see Sections 2.4.1 and 4.3) to attribute each recommendation to friends or general preferences; we then use these attributions to explore data.

When items are recommended because of social influence, the system may indicate a friend as the source of the recommendation. Similarly, when items are recommended because of general preferences, the system may indicate already clicked items that exhibit that preference. On the Etsy data, learned item factors included coherent groupings of items such as mugs, sparkly nail polish, children's toys, handmade cards, and doll clothes. Thus, SPF explains the recommended the handmade soap in Figure 4.1 as coming from general preferences and the others items as coming from social influence. The social and preference signals will not always be cleanly separated; SPF attributes recommendations to sources probabilistically.

Figure 4.6 shows how the proportion of social attribution (as opposed to general preference attribution) changes as a function of user degree on Ciao and Epinions. We observe that

Epinions attributes a larger portion of behavior to social influence, controlled for user degree. Similarly, we can compute the contribution of users to their friends' behavior. 4.7 shows social contribution as a function of indegree; here we see that Epinions users with higher indegree have lower social contribution than low-indegree users.

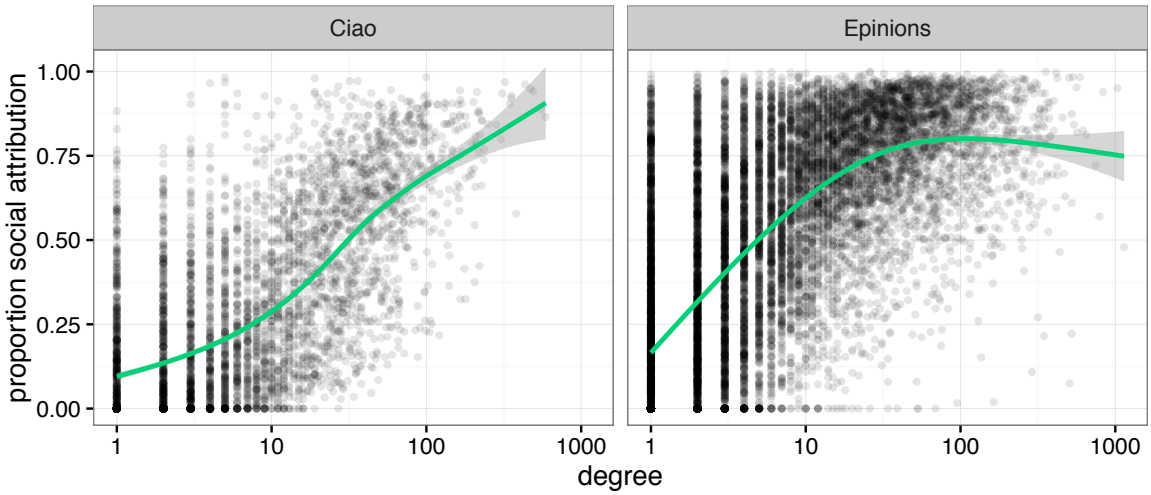


Figure 4.6: The proportion of social attribution (vs. general preference attribution) as a function of user degree. Attributions are calculated on all training data from Ciao and Epinions. Epinions attributes a larger portion of rating to social influence.

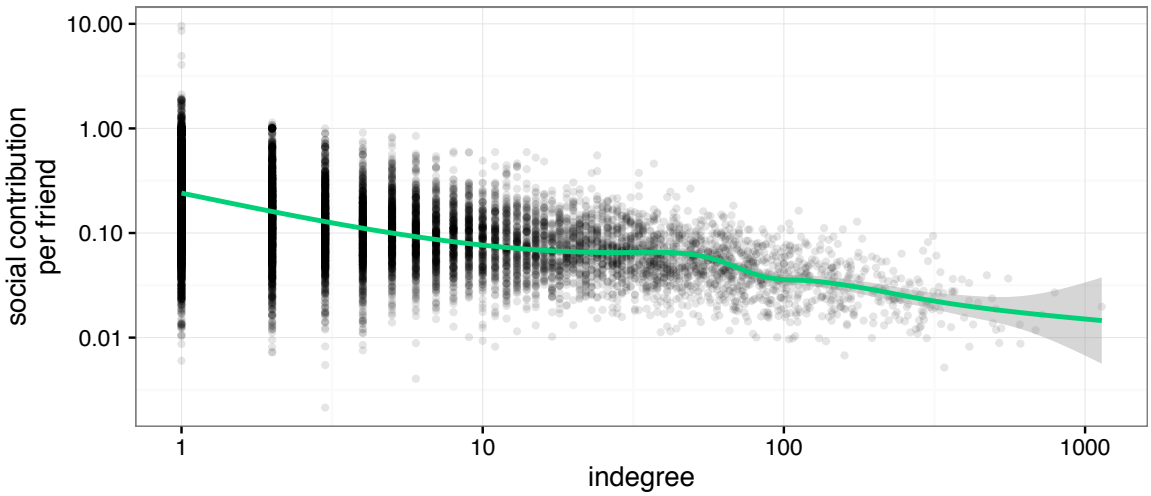


Figure 4.7: Contribution to friends' behavior as a function of indegree, calculated on all Epinions training data. Users with higher indegree have lower social contribution.

4.4.3 Experimental Details

The details of our methods requires some decisions: we must choose the number of latent factors K and set the hyper-parameters.

Choosing the number of latent factors K. All factorization models, including SPF, require the investigator to select of the number of latent factors K used to represent users and items. We evaluated the sensitivity to this choice for the Ciao dataset. (We chose this dataset because of its smaller size; ranking millions of items for every user is computationally expensive for any model.) Figure 4.8 shows per-user average NCRR K varies from 1 to 500; SPF performs best on the Ciao dataset with $K = 40$, though is less sensitive to this choice than some other methods (such as PF).

Hyperparameters. We also must set the hyper-parameters to the gamma priors on the latent variables. The gamma is parameterized by a shape and a rate. We followed Gopalan et al. (2015) and set them to 0.3 for the priors on latent preferences and attributes. We set the hyper-parameters for the prior on user influences to $(2, 5)$ in order to encourage the model to explore explanation by social influence. In a pilot study, we found that the model was not sensitive to these settings.

Does learning influence matter? We can easily fix each user-friend influence at 1, giving us local popularity among a user's social connections. We compared fitted influence against fixed influence on both Ciao and Epinions and found that SPF with fitted influence performs best on both datasets.

In the case of cold-start users, where we know the user's social network but not their click counts on items, SPF will perform equivalently to SF with fixed influence. SPF in this cold-start user scenario performs better than competing models.

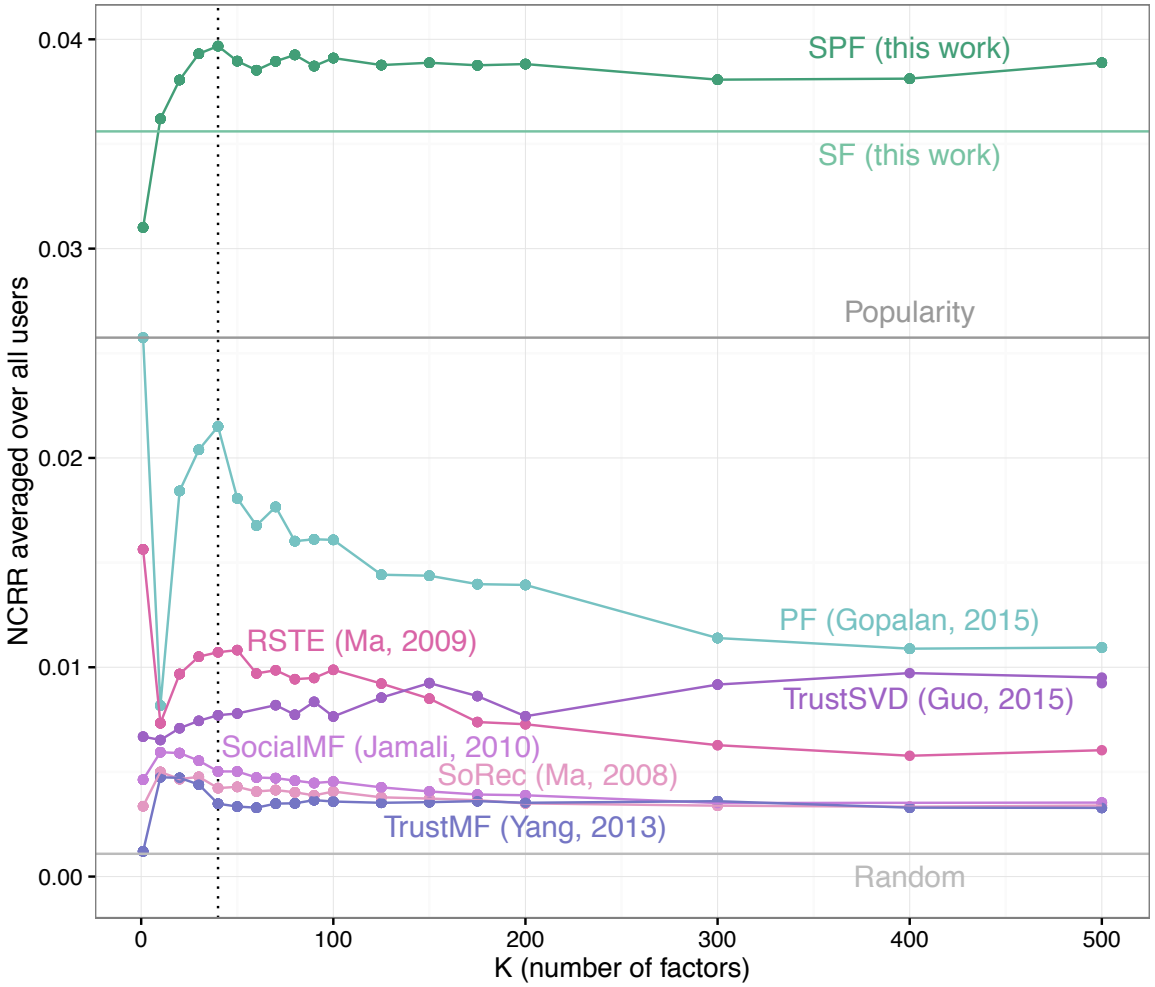


Figure 4.8: Model performance on Ciao data (measured as NCRR averaged over all users) as a function of number of latent factors K. The dotted vertical line at $K = 40$ indicates the best performance for Poisson family models.

4.5 Discussion

We presented social Poisson factorization, a Bayesian model that incorporates a user's latent preferences for items with the latent influences of her friends. We demonstrated that social Poisson factorization improves recommendations even with noisy online social signals. Social Poisson factorization has the following properties: (1) It discovers the latent influence that exists between users in a social network, allowing us to analyze the social dynamics. (2) It provides a source of explainable serendipity (i.e., pleasant surprise due to novelty). (3) It

enjoys scalable algorithms that can be fit to large data sets.

We anticipate that social Poisson factorization will perform well on platforms that allow for and encourage users to share content. Examples include Etsy, Pinterest, Twitter, and Facebook. We note that our model does not account for time—when two connected users both enjoy an item, one of them probably consumed it first. Future work includes incorporating time, hierarchical influence, and topical influence.

5 Exploring Latent Variable Models

Too often diagrams rely solely on one type of data
or stay at one level of analysis.

– Edward R. Tufte

The previous two chapters have presented latent variable models of human behavior and their results on real-world data. In this chapter, we focus on the general process of exploring latent variable models. Once we have a exploratory model fit to data, investigating the fit allows researchers to verify that the model constructs are valid, to criticize the model such that it can be improved, and to understand the underlying data through the lens of the model.

We begin with an overview of visualization concepts, then use these concepts in defining principles for exploring latent variable methods. Finally, we present two example visualizations to demonstrate these principles.

5.1 Visualization Concepts

Whether the goal is to criticize, validate, or understand, the key to exploring a model is visualization.

There are an abundance of techniques for visualizing observed data (Tukey, 1977; Cleveland, 1993; Telea, 2014; Chen et al., 2007; Fayyad et al., 2002). While some of these consider

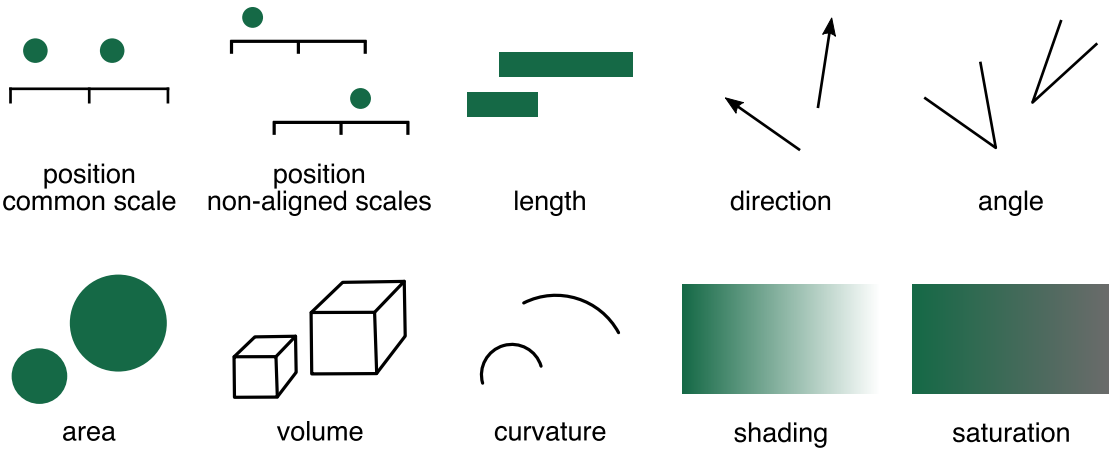


Figure 5.1: Graphical elements, as described by Cleveland and McGill (1984), in approximate order of chances of correct perception by a viewer (top row leads to more accurate perception).

simple models, little work has been done to specifically address visualization for latent variable models. Our goal here is to highlight some basic concepts of visualization that we can use to formulate principles for visualizing latent variable models in the following section.

We begin with the fundamental graphical elements used to convey information. Cleveland and McGill (1984) studied ten graphical elements, shown in Figure 5.1, and found that certain elements conveyed information more accurately to viewers. In order from most to least accurate, these elements are

1. position along a common scale
2. position along non-aligned scales
3. length, direction, angle
4. area
5. volume, curvature
6. shading, color saturation.

Thus, one should prefer elements higher on this list when generating a visualization.

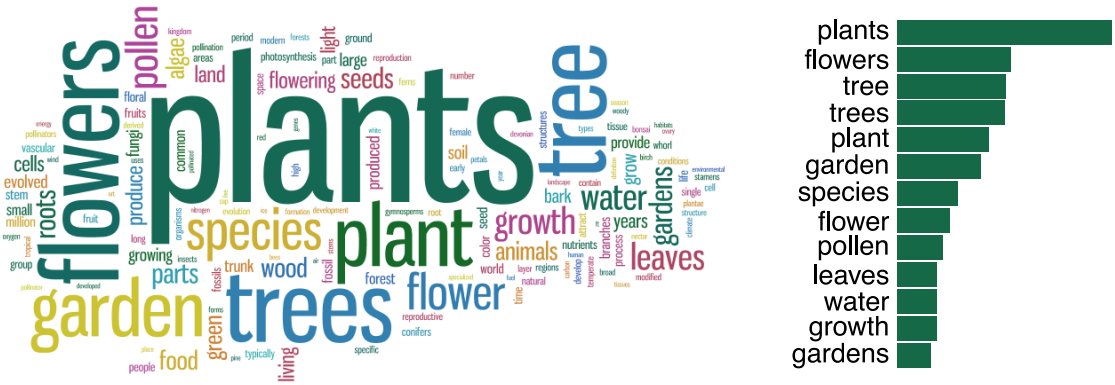


Figure 5.2: Two visualizations of a topic. On the left is a word cloud[1] with location, color, and orientation mapped to random attributes, which may be confusing (e.g., "are words related to trees marked in blue?"). On the right is simple ordered list (truncated for space) with bars indicating the prominence of each word. While the word cloud is better for interpretation a quick glance, the ordered list coveys detailed information more clearly—for example, the word *plants* is nearly twice as prevalent as any other word, which is clear with the bar chart but not with the word cloud.

Wilkinson (2005) builds on these elements to construct a "grammar of graphics," which describes the process of making a complete graphic. In it, Wilkinson emphasizes how to transform and map data to these visual elements—the key art in generating visualizations is selecting which data or model attributes to display and how they should be mapped to graphical elements. In creating the aesthetic attributes of these mappings, it is essential to avoid "chartjunk" (Tufte and Graves-Morris, 1983), or unnecessary ornament. As an example, Figure 5.2 demonstrates two ways of visualizing a topic distribution, one with aesthetic elements mapped randomly. Ornamentation and arbitrary mappings distract from the main message of a graphic.

That message may be conveyed in multiple levels of representation, each with its own scope of detail. Tufte (1991) discusses the importance of including both micro and macro aspects in a visualization, "[s]implicity of reading derives from the context of detailed and complex information, properly arranged. A most unconventional strategy is revealed: *to clarify, add detail.*" Observers of a visualization should be able to examine both large-scale patterns and local details.

[1] Generated by `http://www.wordle.net`.

A broader design perspective is that the manner in which one interacts with an object should be so obvious that the user need not even think about it (Norman, 2013). Visualizations should be similarly intuitive—an investigator should not need to contemplate if their initial impression of a visualization is correct.

Above all, it should be remembered that graphics provide evidence for decision making (Tufte and Robins, 1997). In generating visualizations, the question of interest should be immediately clear, if not the answer as well.

5.2 Principles for Exploring Latent Variable Models

Building on this foundation of modeling visualization approaches (Sections 2.6.1 and 5.1), this section defines a set of five principles that help us in constructing models for exploration and then visualizing the results of those models. The principles are as follows.

1. The questions to be answered must be clear. This principle is applicable to both developing latent variable models and constructing visualizations based on models. It involves placing the model (or visualization) in a larger context and defining the function of the model in that context.

2. Each latent variable must map to an intuitive concept. As interpretation involves the discovery of relationships, it is impossible to interpret one variable's relationship with another variable when of of them is ambiguously or arbitrarily defined. If each latent variable maps to a meaningful concept, then the relationships between those concepts can be explored.

3. Each graphical element must be meaningful. When graphical elements are mapped to random values, investigators may question the meaning of these graphical elements, as shown in Figure 5.2. Additionally, the graphical elements must be chosen with care such that they successfully convey the intended meaning—mapping to graphical elements such as text size can be ambiguous: is the viewer supposed to derive meaning from the text height

or the area the text covers?

4. Model results must be displayed in conjunction with the original data. By portraying both the learned model parameters and the original data, a visualization creates a two-tiered view of the problem. The model provides high level summaries of the data, and the original data provides low-level details. Depending on the model specification, the data can also be augmented by local parameters of the model.

5. Interactions must be obvious. If there are physical interactions to a visualization (e.g., a mouse click to display more information), then these interactions should be easy to discover and not require explanation outside of simple visual cues (e.g., the element changes color when the cursor hovers over it). This also applies to static visualizations and model output more generally: it should be clear how to investigate and validate the model results to find the answers to the questions required by our first principle.

These principles are intentionally broad such that they can be applied to latent variable models in general. To solidify them with a concrete example, we now apply them in constructing visualization frameworks for topic models (Section 2.3) and Capsule (Chapter 3).

5.3 Visualizing Topic Models

Probabilistic topic models (Chapter 2) are a set of machine learning tools that discover the hidden thematic structure in a collection of documents; they find salient themes and represent each document as a combination of themes. However, topic models are high-level statistical tools. A user must scrutinize numerical distributions to understand and explore their results; the raw output of the model is not enough to create an easily explored corpus.

We propose a method for using a fitted topic model to organize, summarize, visualize, and interact with a corpus. With our method, users can explore the corpus, moving be- tween high level discovered summaries (the "topics") and the documents themselves, as Figure 5.3

illustrates.

Our design is centered around the idea that the model both summarizes and organizes the collection. Our method translates these representations into a visual system for exploring a collection, but visualizing this structure is not enough. The discovered structure induces relationships—between topics and articles, and between articles and articles—which lead to interactions in the visualization.

Thus, we have three main goals in designing the visualization: summarize the corpus for the user; reveal the relationships between the content and summaries; and, reveal the relationships across content. We aim to present these in a ways that are accessible and useful to a spectrum of users, not just machine learning experts.

Our method can be applied to any collection from a topic model fit to its documents.[2] We describe the details of our method in the rest of the chapter. First, we survey prior work on corpus browsers. Second, we discuss our interactive visualization method, describing our design choices for visualizing a corpus with the output of a topic modeling algorithm. Third, we point to our open source implementation of the method and describe several use cases of the resulting corpus navigators. Finally, we explore a set of qualitative user reviews from a pilot study.

5.3.1 Related Work

Simple electronic corpus browsers exist in most operating systems; they allow users to list, sort, and search documents and their meta-data. However, using these tools can be unwieldy when a user does not have a specific search query or a good understanding of the corpus.

[2]There are many open source implementation of topic modeling algorithms, e.g. http://www.cs.princeton.edu/~blei/lda-c, http://cran.r-project.org/web/packages/lda/ and http://mallet.cs.umass.edu/topics.php.

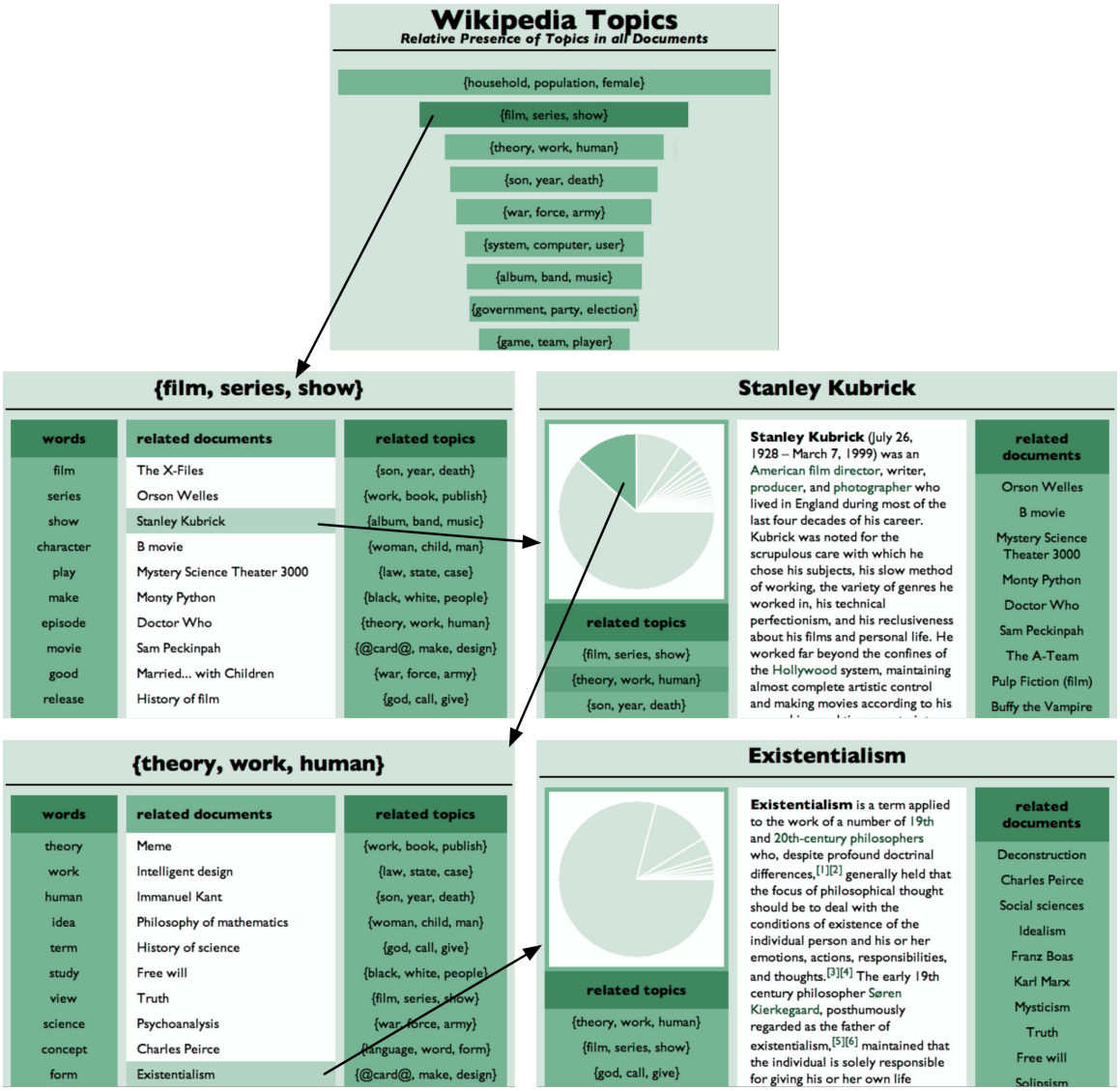


Figure 5.3: Navigating Wikipedia with a topic model. Beginning at the top, we see a set of topics, each of which is a theme discovered by a topic modeling algorithm. We click on a topic about film and television. We choose a document associated with this topic, which is the article about film director Stanley Kubrick. The page about this article includes its content and the topics that it is about. We explore a related topic about philosophy and psychology, and finally view a related article about Existentialism. This browsing structure—the themes and how the documents are organized according to them—is created by running a topic modeling algorithm on the raw text of Wikipedia and visualizing its output.

Researchers have proposed several solutions to the problem of understanding large document corpora. Examples include Exemplar-based Visualization (Chen et al., 2009), FacetAtlas (Cao et al., 2010), and ThemeRiver (Havre et al., 2000). These visualizations help users understand the corpus as a whole, but they do not provide a more detailed exploration of individual documents.

At the document level, researchers have pursued several methods for visually summarizing individual documents. Examples include Phrase Nets (Van Ham et al., 2009), Document Cards (Strobelt et al., 2009), and the Word Tree (Wattenberg and Viégas, 2008). These visualizations accomplish their goals, but have no mechanism for giving the context of the analyzed documents within the larger corpus. Our visualization provides both a high-level summary of the corpus and links between the summary and individual documents.

In some document visualization problems, the structure of the collection is given. There has been much research on using *facets*, or meta-data, to innovate the browsing experience (Hearst, 2008; Lee et al., 2009; Thai and Handschuh, 2010). However, many corpora do not contain meta-data, and it can be difficult to obtain. The approach we present visualizes a discovered structure in a corpus, without requiring human annotation. Our technique is tailored to the structure that topic models discover.

Previous topic modeling research has focused on building new topic models to capture more structure—examples include those that represent time series (Blei and Lafferty, 2006), authorship (Rosen-Zvi et al., 2004), or citation (Chang and Blei, 2009)—and also on improving the algorithms for fitting data to a topic model (Newman et al., 2007; Teh et al., 2006; Hoffman et al., 2010), but topic models are mainly used for exploratory analysis. There has been little research on how to best visualize and interact with an analyzed collection.

Topic model researchers have typically used topic browsers for exploring the fitted model in order to evaluate model algorithms (Newman et al., 2006)[3]; TopicNets (Gretarsson et al.,

[3]See also: http://bit.ly/browser-blei, http://bit.ly/browser-rexa, and http://bit.ly/

2012) and the Topic Browser (Gardener et al., 2010) are the most notable of these topic browsers. These visualizations emphasize topics; the document content is rendered to the side or as an external link rather than integrated into the browser. Further, these browsers include little visual representation: they rely mostly on links and numbers to convey meaning.

We present a way of using topic models to help learn about and discover items in a corpus. With this goal it is important to forgo jargon and difficult-to-interpret numbers, and to emphasize interaction and visualizations that are meaningful to many types of users. Our navigator presents the output of a topic model in an interface that illuminates a given corpus to non-technical users.

5.3.2 Visualizing a Topic Model

How can we visualize a corpus through the lens of a topic model? Our goals are to use the topic model to summarize the corpus, reveal the relationships between documents and the discovered summary, and reveal the relationships between the documents themselves. We applied our visualization to 100,000 articles from Wikipedia, which we will use as a running example. (We also visualized 3,000 articles from the New York Times and 61,000 US Federal Cases. Others have applied our visualizer to 20,000 articles from the ArXiv, which is a large repository of scientific preprints.)

LDA decomposes a collection of documents into *topics*—biased probability distributions over terms—and represents each document with a (weighted) subset of the topics. When fit to a set of documents, the topics are interpretable as themes in the collection, and the document representations indicate which themes each document is about. Thus, the learned topics summarize the collection, and the document representations organize the corpus into overlapping groups.

`browser-czdml`.

For example, the most frequent words in the topic {*film, series, show*} in Figure 5.3 are *film*, *series*, *show*, *character*, *play*, *make*, *episode*, and *movie*; Wikipedia articles that exhibit this topic include *The X-files*, *Orson Welles*, and *History of film*. In contrast to machine learning algorithms for classification and prediction, topic models are an unsupervised method. The documents are not labeled with meta-data, e.g., "related to film." Rather, the topics and how the documents exhibit them are discovered by the algorithm.

In advance of building the visualization, the user must collect the documents and run a topic modeling algorithm on them. Our visualization uses both the observed data and the inferred topic model variables.[4] The topic model variables are the topics β_k, each of which is a distribution over a vocabulary, and the topic proportions θ_d, one for each document and each of which is a distribution over the topics.

The latent and observed variables of a topic model are numerous, and their relationships are complex. Thus, we use multiple views to illuminate the structure. We created a basic navigator that fully represents a corpus through the lens of an LDA analysis. In this section, we explain our design choices.

Visualizing the Elements of a Topic Model. The navigator contains two main kinds of pages: one for displaying the discovered topics and the other for presenting the documents. There are also overview pages, which illustrate the overall structure of the corpus; they are a launching point for browsing.[5]

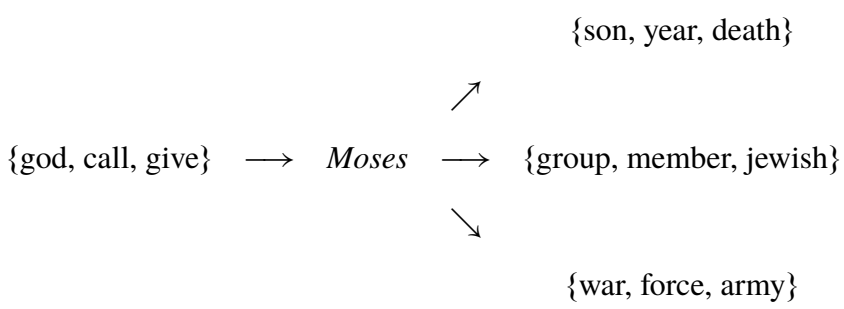
These pages display the corpus and the discovered structure. But this is not sufficient—we also use the topic model inferences to present connections between the documents and topics. With these connections, a user can move between summary and document-level presentations. Limiting a user to a summary-level presentation of the corpus gives the approach of previous topic model visualizations; limiting them to document-level is simply viewing the original

[4]Note that the we use variables to indicate their posterior expectations. This is to make the notation simple.

[5]Additionally, term pages integrate the topic model with a traditional index of the collection, and are presented in a similar format to the topic pages.

corpus.

Hence, in our visualization every element on a page links a user to a new view. With these links, a user can easily traverse the network of relationships in a given corpus. For example, from a topic page a user can link to view a specific document. This document might link to several topics, each of which the user can explore. If we use the topic three words to represent a topic, we can show an example of this browsing experience.



We illustrated another navigation example in Figure 5.3.

An advantage of our design is that every type of relationship has a representation and an interaction. This illuminates the structure of corpus to a user and helps her navigate that structure. Further, note that any individual variable may occur in multiple views; all relationships are many-to-many. The topic {*food, make, wine*} is related to documents titled *Tea* and *Migraine*, and so would appear on both their respective pages.

Topic Pages. Topics summarize the corpus. In the output of an inference algorithm, they are probability distributions over the vocabulary. But topics tend to be sparse, and so a good visual representation is as a set of words that have high probability (as opposed to a traditional view of a distribution, such as a bar graph). Given such a set, users can often conceive meaning in a topic model (Chang et al., 2009). For example, one can intuitively glean from the three words {*school, student, university*} (Figure 5.5) that this topic is about education and academics. (Our visualization might also reveal uninterpretable topics, which indicates a misfit to the data. Techniques like those of Newman et al. (2010); Mimno and

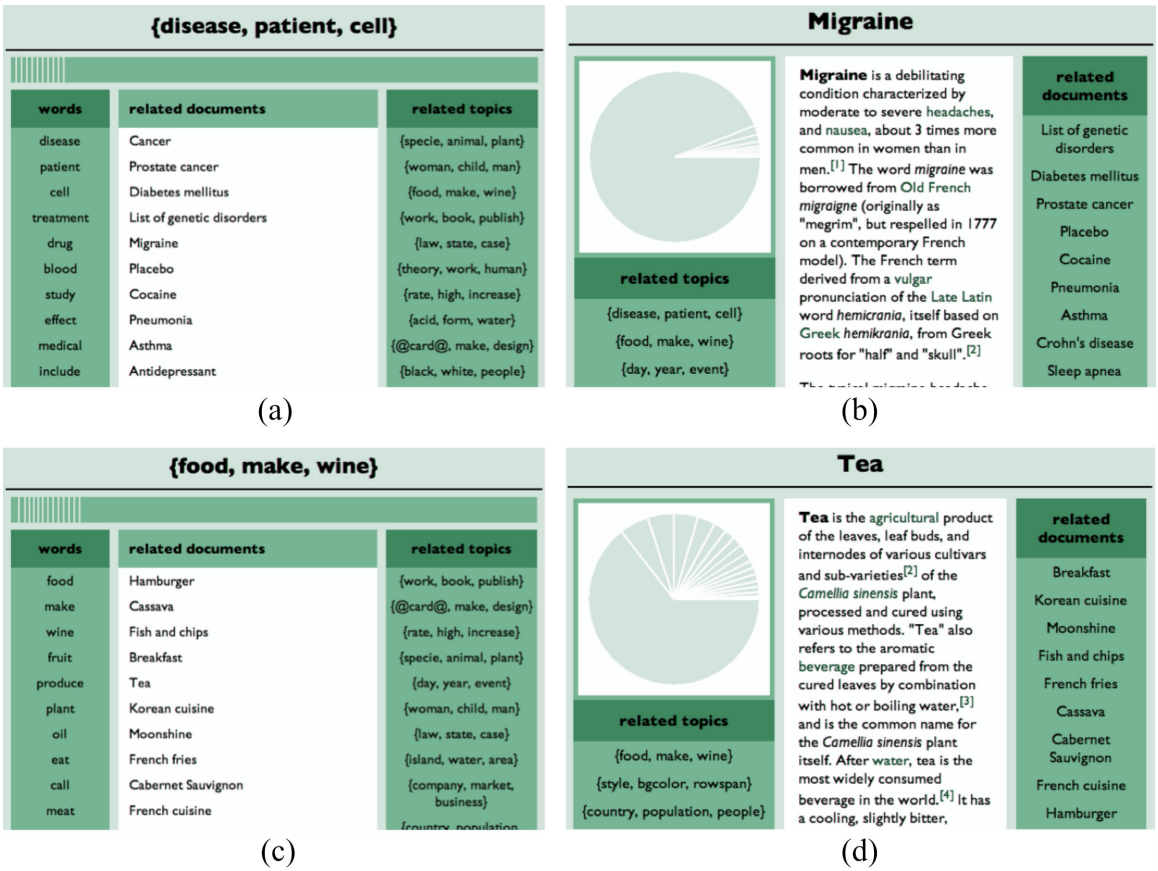


Figure 5.4: Topic pages and document pages from the navigator of Wikipedia. **(a)** A view for a topic on diseases and medicine. **(b)** A document view for the Wikipedia article titled *Migraine*. This exhibits the {*disease, patient, cell*} topic shown in (a) at a very high percentage and the topic {*food, make, wine*} in (c) at a low percentage. **(c)** A view for a topic on food and cooking. **(d)** A document view for the Wikipedia article titled *Tea*. This exhibits the {*food, make, wine*} topic in (c) strongly.

Blei. (2011) might be used to prune these topics.)

We illustrate example topic pages in Figures 5.4 and 5.5, (a) and (c). In these pages, the terms are represented as a list of words in the left column, ordered by their topic-term probability $\beta_{k,v}$.[6] We chose not to scale the term text by probability because it conveys meaning imprecisely; most attributes are perceived on non-linear scales (Wilkinson, 2005).

The center column of the view lists documents that exhibit the topic, ordered by inferred topic proportion θ_{dk}. Documents are rendered by their titles on this page and each links to

[6]Each vocabulary term also links to a term page.

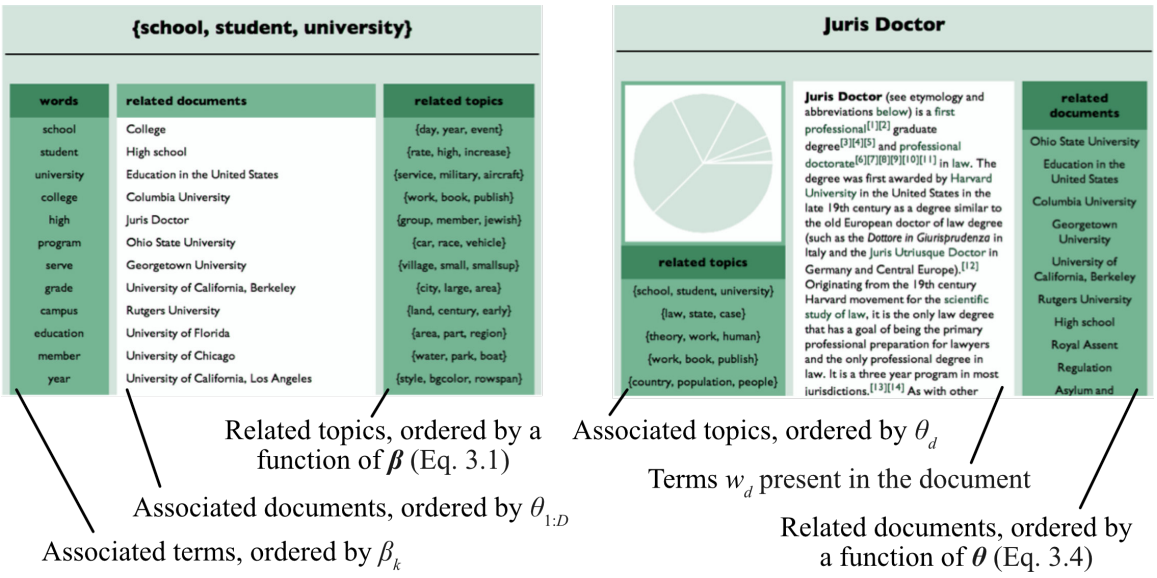


Figure 5.5: A topic page and document page from the navigator of Wikipedia. We have labeled how we compute each component of these pages from the output of the topic modeling algorithm.

its corresponding document page; this enables a user to move from the high-level topic view to the low-level document view. We can see that the list of documents related to {*food, make, wine*} are tied to food and eating.

$$\{\text{food, make, wine}\} \longrightarrow \begin{cases} \nearrow \textit{Breakfast} \\ \textit{Korean cuisine}. \\ \searrow \textit{French fries} \end{cases}$$

The topic in Figure 5.5 titled {*school, student, university*} is related to articles on general concepts such as *College* and *Education in the United States* but also to articles on specific institutions, like *Columbia University* and *Ohio State University*. These relationships between topics and documents were discovered by the topic model.

Finally, related topics are also listed with corresponding links, allowing a user to explore the high-level topic space. Topic similarity is not inferred directly with LDA, but can be computed from the topic distributions that it discovers. Related topics are shown in the right

column of the topic page by pairwise topic dissimilarity score

$$\xi_{ij} = \sum_{v=1}^{V} \mathbf{1}_{\mathbb{R}_{\neq 0}}(\beta_{i,v}) \mathbf{1}_{\mathbb{R}_{\neq 0}}(\beta_{j,v}) \left| \log(\beta_{i,v}) - \log(\beta_{j,v}) \right| \tag{5.1}$$

where the indicator function is defined as

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases} \tag{5.2}$$

This is related to the average log odds ratio of the probability of each term in the two topics. This metric finds topics that have similar distributions. Note that there are other ways to define topic similarity, for example by looking at co-occurrences of topics within documents (Blei and Lafferty, 2007).

Continuing with our example topic from Figure 5.4 (a), this metric scores the following topics highly.

{specie, animal, plant}

{disease, patient, cell} → {woman, child, man}

{food, make, wine}

Since the original topic relates to matters of health, it makes sense that the related topics cover a spectrum of concepts from the natural world to human lifestyles.

Document Pages. Document pages render the original corpus, providing a low-level view. In the case of the Wikipedia navigator, HTML content is drawn directly from Wikipedia articles, as shown in Figures 5.4 and 5.5, (b) and (d). The ArXiv navigator discussed in Section 5.3.3 incorporates meta-data about the articles as well; this is shown in right section of Figure 5.7.

We supplement each document by showing the topics that it exhibits, where a topic is rendered as its top three most probable words. These text-rendered topics are listed in the left column of each page and ordered by their topic proportions θ_{dk}. In addition to this related topics column, topics are also displayed in a pie chart, showing their respective proportions within the document. Pie slices highlight the topic titles below on hover and vice-versa, and both lead to the appropriate topic page when clicked. Since the topic proportions θ_d sums to one, this is a depiction of document descriptions $\boldsymbol{\theta}$ that matches human intuition.

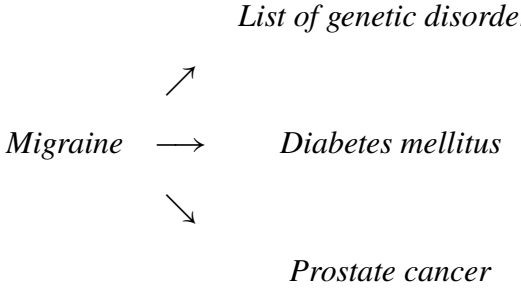
Glancing at the pie chart of the document page in Figure 5.5, one sees that the *Juris Doctor* article is roughly a third about academia, a third about law, and a third about other topics. Any other depiction, such as a bar chart, would require numerical annotation to be as specific. Every rendering of a topic links to its respective page, allowing a user to shift to a high-level topic view.

Finally, documents are associated with similar documents. Like topic similarity, document similarity is not inferred directly with LDA, but is defined by the topic proportions:

$$\sigma_{ij} = \sum_{k=1}^{K} \mathbf{1}_{\mathbb{R}_{\neq 0}}(\theta_{ik}) \mathbf{1}_{\mathbb{R}_{\neq 0}}(\theta_{jk}) \left| \log(\theta_{ik}) - \log(\theta_{jk}) \right|. \tag{5.3}$$

This metric says that a document is similar to other documents that exhibit a similar combination of topics.

In the example shown in Figure 5.4 (b), the article is related to other documents as follows.



Related documents, like all other relationships, link to their respective pages, allowing a user to explore the documents.

Overview Pages. Overview pages are the entry points to exploring the corpus. In the simplest of these pages, we rank the topics by their relative presence in the corpus and display each in a bar with width proportional to the topic's presence score p_k: the sum of the topic proportions for a given topic over all documents,

$$p_k = \sum_{d=1}^{D} \theta_{dk}. \tag{5.4}$$

Examples of this view can be found in Figure 5.6. From this figure, we see that many documents are related to the topic {*household, population, female*}; this is consistent with our observations of the corpus, which includes many Wikipedia articles on individual cities, towns, and townships. Similarly, the high scoring of the {*film, series, show*} topic is likely due to the number of articles dedicated to particular movies and television shows. One of the lowest scoring topics by this scale is {*water, park, boat*}, which has a narrow scope: outdoor recreation.

We have created additional overview pages—these give users alternative entry points to variable pages, which may be found in any of our demonstration navigators.

5.3.3 Implementation and example use

We provide an open source implementation of the topic modeling visualization. There are three steps in applying our method to visualizing a corpus: (1) run LDA inference on the corpus to obtain posterior expectations of the latent variables (2) generate a database and (3) create the web pages to navigate the corpus.

Any open-source LDA package can be used; we used LDA-C.[7] (Using an alternative pack-

[7] http://www.cs.princeton.edu/~blei/lda-c

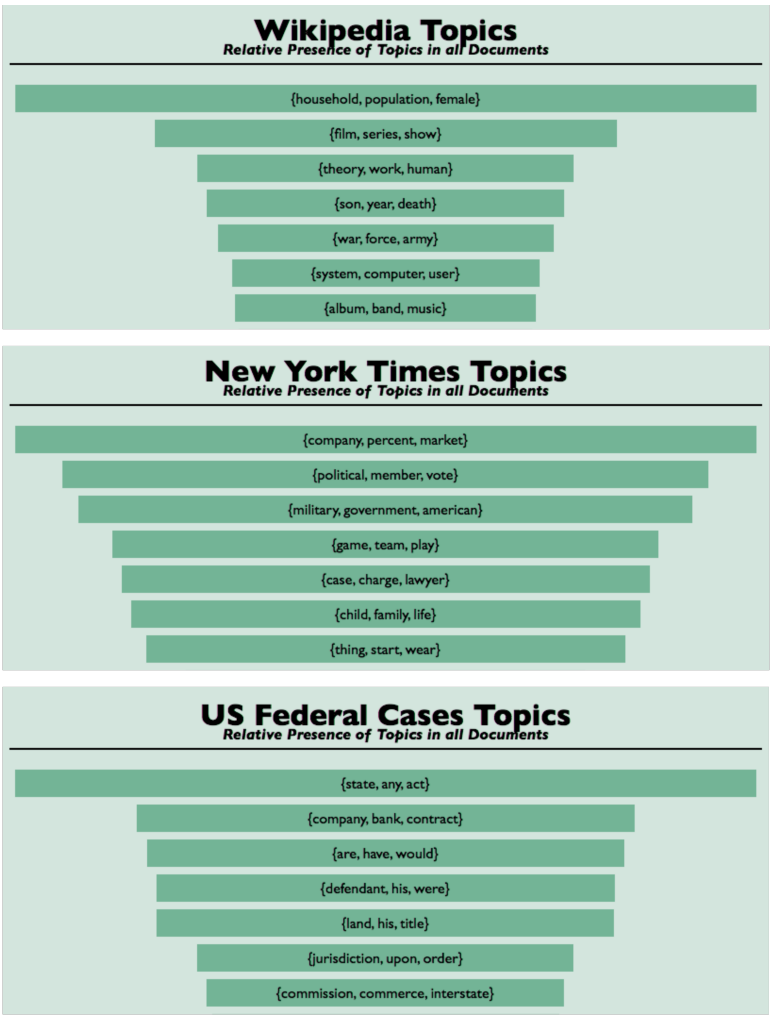


Figure 5.6: Topic overviews from a visualization of Wikipedia (top), the New York Times (center), and US Federal cases (bottom). All of these navigators are online (see the Section 5.3.3).

age might require changes to our provided database generator script.) We implemented the remainder of the pipeline in python. It can be found at `https://github.com/ajbc/tmve-original`.

We also created an alternative version that generates pages as requested (or "lazily") using Django[8] as an alternative, though it excludes some of the similarity links. With this variant, models can also be viewed as the model runs, meaning that a user can start exploring a corpus almost immediately with a model like online LDA (Hoffman et al., 2010) which

[8] `https://www.djangoproject.com`

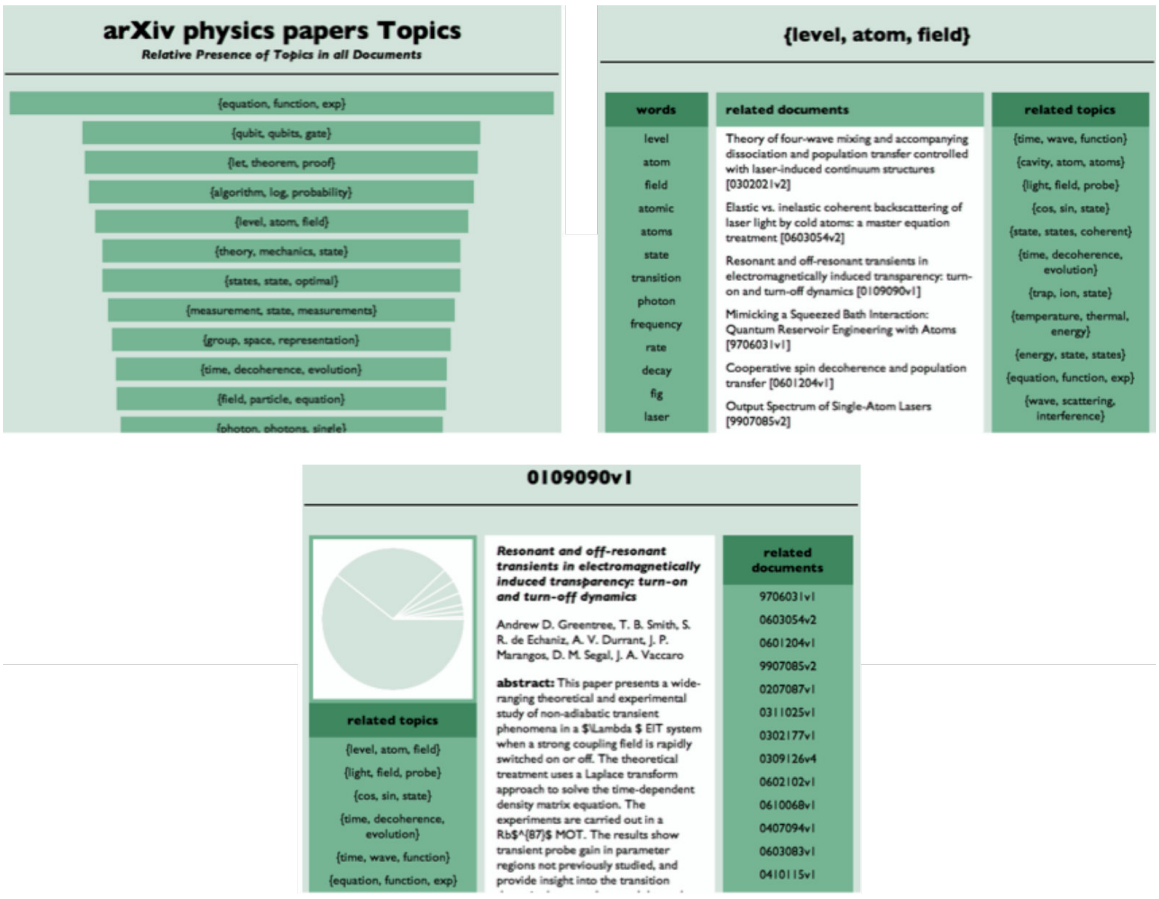


Figure 5.7: A navigator of the arXiv of Physics preprints, generated by Ivan Savov using our open source implementation of the topic model visualization.

is able to scale to millions of documents. The source for this variant may be found at https://github.com/ajbc/tmv.

There are three examples of navigators using our visualization.

- **100,000 Wikipedia articles.** We analyzed Wikipedia articles with a 50-topic LDA model; the navigator can be found at http://bit.ly/wiki100. All figures in this chapter are drawn from this demonstration unless noted otherwise.
- **61,000 US Federal Cases.** We created a navigator of US Federal Cases[9] with 30 topics generated with LDA. A page form the resulting navigator can be seen in Figure 5.6; it can be found in full at http://bit.ly/case-demo.

[9] Obtained via http://www.infochimps.com/datasets/text-of-us-federal-cases; link no longer active.

- **New York Times.** We also applied our method to a corpus of 3,000 New York Times articles, generating 20 topics with LDA. The resulting navigator can be seen in Figure 5.6; it can be found in full at `http://bit.ly/nyt-demo`.
- **ArXiv.** One week after the source code was released we received links to a navigator of arXiv (a large archive of scientific preprints) that was generated using our code with few adaptions. The results can be seen in Figure 5.7; the full browser is no longer available online.

5.3.4 Preliminary User Study

We conducted a preliminary user study on seven individuals, asking for qualitative feedback on the Wikipedia navigator. In general, the reviews were positive, all noting the value of presenting the high-level structure of a corpus in addition its low-level content. One reviewer claimed that it was organized similarly to his own way of thinking.

Six individuals responded that they discovered connections that would have remained obscure by using Wikipedia traditionally. For example, one user explored articles about economics and discovered countries with inflation or deflation problems of which he had previously been unaware.

All of the reviewers preferred a search when looking for something specific; the negative feedback we received focused on our lack of integrating search. We acknowledge that searching is an important feature for browsing a large corpus and that it should be included to complete the system.

The only other negative feedback was due to the small scope of 100,000 Wikipedia articles: reviewers were unable to find detailed information on narrow subjects like music synthesizers or the evolutionary history of cats. This is a problem with the corpus as selected rather than the browsing structure we implemented.

While our navigator of 100,000 Wikipedia articles was no replacement for Wikipedia, three of the individuals stated that they would like to see Wikipedia supplemented with a topical structure or would use that structure if it existed on Wikipedia; the remainder of the reviewers implied that such a system would be useful in general.

5.4 Visualizing Capsule

Like LDA, Capsule (Chapter 3) also represents topics as distributions over words, and can use many similar elements to visualize and explore the model.

In constructing a visualization for Capsule, we created overview pages for events (Figure 5.8), entities (Figure 5.9), and general topics. These pages serve as launching points to investigate the corpus.

From here, users can investigate specific event, entity, and general topic pages that display ordered lists of relevant terms and relevant document. For entity and event pages, we scraped Wikipedia to provide descriptions for the entities and any real-world events that occurred in a given time interval. Users can also navigate to document pages, as shown in Figure 5.10).

Source code for this visualization is available at `https://github.com/ajbc/capsule-viz` and a live demo is present at `http://www.princeton.edu/~achaney/capsule/`.

5.5 Discussion

In this chapter, we have proposed five principles for constructing exploratory models and then visualizing the results of those models. We then demonstrated these principles by creating navigators for LDA and Capsule that summarize the corpus for the user and reveal

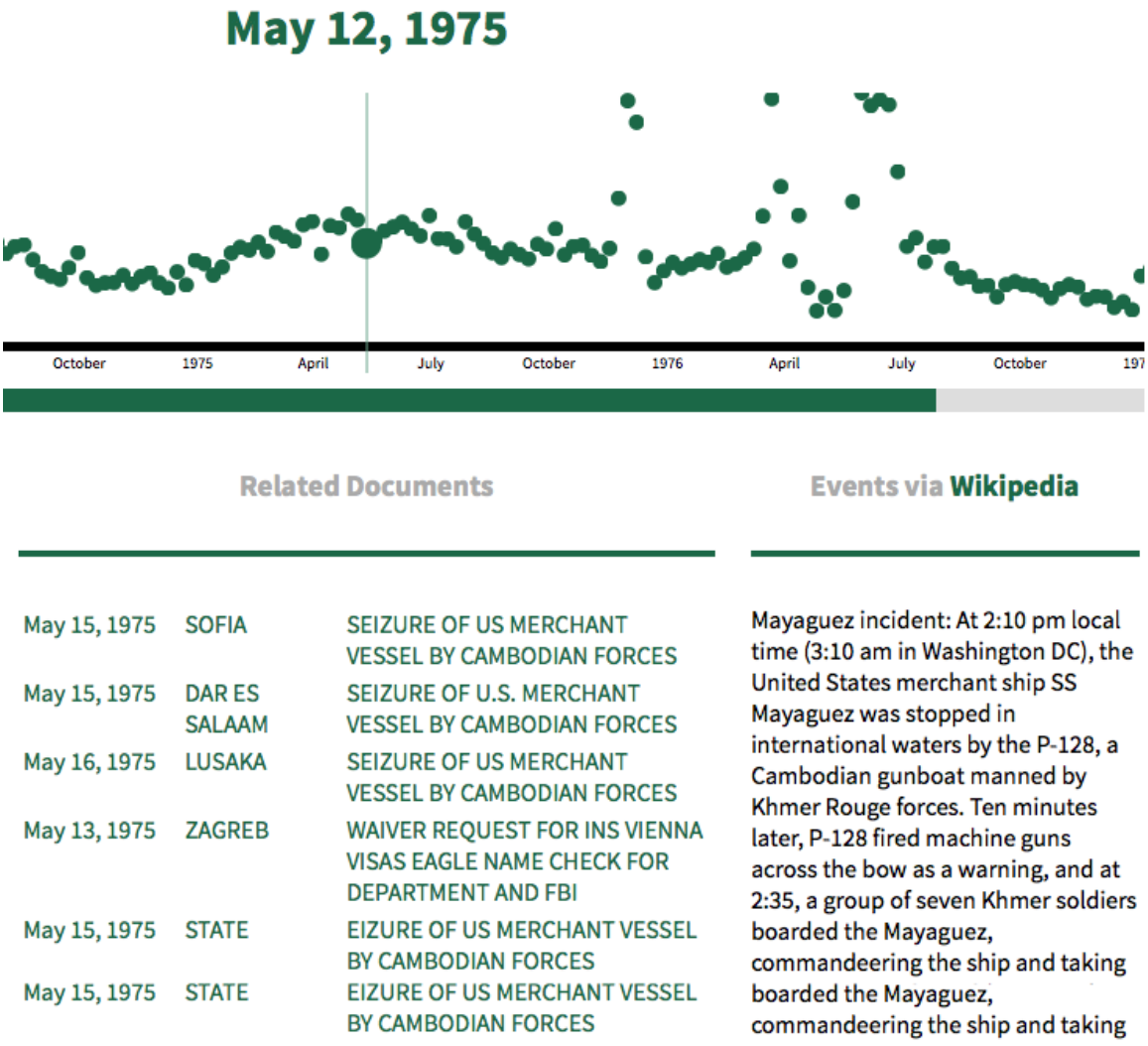


Figure 5.8: A screen-shot the Capsule visualization of US State Department cables. The event overview view allows users to select a time interval, which then displays the top terms, most relevant documents, and real-world events scraped from Wikipedia for that time interval.

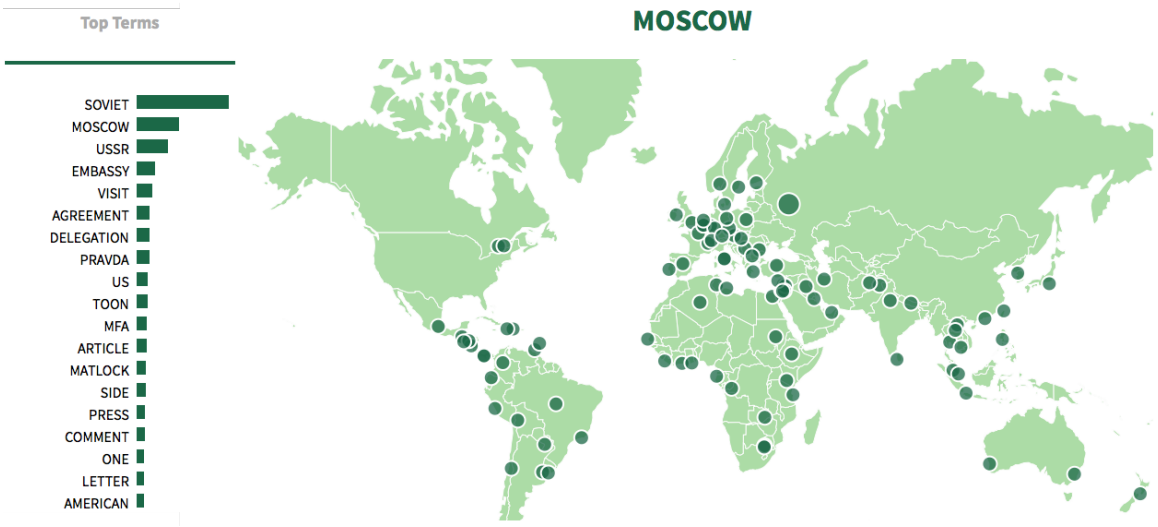


Figure 5.9: A screen-shot of Capsule visualization of US State Department cables. The entity overview allows users to select entities on a map, which then displays the top terms. most relevant documents, and a description scraped from Wikipedia for that entity.

Figure 5.10: A screen-shot of Capsule visualization of US State Department cables. This view allows users to investigate a given document. For here, one may navigate to event, entity, or general topic pages that are relevant to the document (right column).

relationships between and across content and summaries. Overview pages allow users to understand the corpus as a whole before delving into more specific exploration via individual variable pages. We have achieved this with navigator designs that illuminates a given corpus to non-technical users; understanding our navigators do not require an understanding of the details of the underlying models.

We see potential for our visualizations to have many applications. LDA can be applied to scientific, historical, web, and news articles, all of which would benefit from an accompanying navigator. Capsule can similarly be applied to email and blog posts.

These visualizations may be extended and tailored to specific extensions of the models, but the principles for exploring models apply to any latent variable model.

6 Conclusion

End? No, the journey doesn't end here.

– J. R. R. Tolkien

This dissertation has presented three pieces of work that relate to statistical modeling of human behavior and the exploration of model results. We developed two additive Poisson models—Capsule in Chapter 3 and SPF in Chapter 4—for human-centered applications and described how to attribute observed behavior to sources of influence. We also presented visualization based on an underlying statistical model as a first-class research problem, and provided five principles in Section 5.2 to guide the construction of these systems. We demonstrated these principles with exploratory tools for LDA and Capsule in Chapter 5, and with static visualizations for SPF in Chapter 4. To conclude, we review the contributions of this dissertation and point to directions for future work.

6.1 Contributions

In this section, we itemize the contributions presented throughout this dissertation.

- In Chapter 2, we presented latent Dirichlet allocation (Section 2.3) and Poisson factorization (Section 2.4) in a way that enabled us to emphasize the connections between the models (Section 2.5.1). Our contribution was presenting how these approaches

can be combined into a hybrid model that draws on the strengths of each model.

- In Chapter 3, we developed the Capsule model for detecting and characterizing events.
- We also derived variation inference algorithm for Capsule and released accompanying source code.
- In Chapter 4, we developed social Poisson factorization (SPF), another additive Poisson model like Capsule. SPF incorporates the ratings of friends (and not just friends' general preferences) in providing personalized recommendations.
- We derived a variational inference algorithm for SPF and released accompanying source code.
- For additive Poisson models in general, as well as Capsule and SPF in particular, we outlined how to attribute observations to different latent variables. While we did not make any causal claims in this dissertation, this additive framework is ripe for causal inference.
- In Chapter 5, we introduced a set of five principles for model exploration and visualization.

    1. The questions to be answered must be clear.
    2. Each latent variable must map to an intuitive concept.
    3. Each graphical element must be meaningful.
    4. Model results must be displayed in conjunction with the original data.
    5. Interactions must be obvious.

- In Chapter 5 we presented a visualization pipeline for LDA in keeping with thee principles, and released accompanying source code and demonstrations.
- To explore the results of Capsule, we also developed a navigator in Chapter 5 and

released source code for this visualization.

6.2 Future Directions

We have established a foundation of research which can be built on in many ways. Future work includes both specific tasks and addressing broad questions.

The Capsule model and corresponding visualization can be developed further—the model can be adapted to include network interactions; for example, one entity's involvement in an event could spawn its ally to be involved. The model's relationship with Poisson processes, especially excitatory processes, can also be explored.

For social Poisson factorization, the model can be extended to include some notion of time; when time is observed, it would introduce an order into friends consuming the same content. This implies directionality to influence which would result in a well-defined joint. This would open the door to investigators making causal claims with future adaptions of the model.

In terms of specific tasks for visualizing topic models, visualizations could be adapted to include topic modeling extensions. Other models incorporate time series of topics (Blei and Lafferty, 2007), hierarchies of topics (Blei et al., 2010), authorship (Rosen-Zvi et al., 2004), document impact (Gerrish and Blei, 2010), and models where the data determine the number of topics (Teh et al., 2007). Visualization should exist to accommodate a variety of topic models and variables that might be added to the analysis.

More broadly, visualizations can be created for any exploratory latent variable model. We should ask: how can we include the full posterior in our visualizations and explorations, not just the expectation of the mean for each parameter? Bayesian models are especially valuable in cases where estimating the full posterior distribution (or even simply uncertainly for certain parameters) is important; visualizations can be adapted and developed to reflect

this importance.

This dissertation lays the groundwork for many directions of future research. One open avenue is to explore formal notions of causality with scalable machine learning algorithms. An obvious application of this is understanding the impact of recommender systems; more broadly, there are many opportunities to understand influences on human behavior using massive data and latent variable models. The models and exploration approaches presented in this dissertation are the first steps in this line of research.

APPENDICES

A | Prior Publications

Much of the work presented here has been presented and published previously; this appendix itemizes these prior versions of the work. This dissertation presents these research themes jointly, providing both a broader context for the work and deeper investigations for each of the component problems.

Chapter 3: Detecting and Characterizing Events

- Allison J.B. Chaney, Hanna Wallach, David M. Blei, and Matthew Connelly. *Detecting and Characterizing Events.* Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2016. (Also referenced in Chapter 5.)
- Allison J.B. Chaney, Hanna Wallach, and David M. Blei. *Who, What, When, Where, and Why? A Computational Approach to Understanding Historical Events Using State Department Cables.* Sixth Annual New Directions in Analyzing Text as Data Conference (Text as Data), 2015.

Chapter 4: Social Poisson Factorization

- Allison J.B. Chaney, David M. Blei, and Tina Eliassi-Rad. *A Probabilistic Model for Using Social Networks in Personalized Item Recommendation.* Proceedings of the 9th ACM Conference on Recommender Systems (RecSys), 2015.
- Allison J.B. Chaney, Prem Gopalan, and David M. Blei. *Poisson Trust Factorization for Incorporating Social Networks into Personalized Item Recommendation.* NIPS

Workshop: What Difference Does Personalization Make?, 2013.

Chapter 5: Exploring Latent Variable Models

- Allison J.B. Chaney and David M. Blei. *Visualizing Topic Models.* Proceedings of the Sixth Annual International AAAI Conference on Web and Social Media (ICWSM), 2012.

B Complete Conditional Derivations

This appendix contains the derivation of complete conditionals for the topic distributions $\boldsymbol{\beta}$ described in Chapter 2 to illustrate the similarities between LDA and PF. Derivations for document representations $\boldsymbol{\theta}$ are similar but not shown. Word-specific topic assignments $\mathbf{w}$ are also not shown.

Derivation of LDA β Updates

$$
\begin{aligned}
p(\beta_k \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z}) &\propto p(\mathbf{w}, \boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{z}) \\
&= \prod_{d=1}^{D} \left[p(\theta_d \mid \alpha_\theta) \prod_{n=1}^{N_d} [p(z_{dn} \mid \theta_d)\, p(w_{dn} \mid \beta, z_{dn})] \right] \prod_{k=1}^{K} p(\beta_k \mid \alpha_\beta) \\
\log p(\beta_k \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z}) &\propto \sum_{d=1}^{D} \left[\log p(\theta_d \mid \alpha_\theta) + \sum_{n=1}^{N_d} [\log p(z_{dn} \mid \theta_d) + \log p(w_{dn} \mid \beta, z_{dn})] \right] + \\
&\quad \sum_{k=1}^{K} \log p(\beta_k \mid \alpha_\beta) \\
&\propto \sum_{d=1}^{D} \sum_{n=1}^{N_d} \log p(w_{dn} \mid \beta, z_{dn}) + \log p(\beta_k \mid \alpha_\beta) \\
&= \sum_{d=1}^{D} \sum_{n=1}^{N_d} \log \mathrm{Cat}(w_{dn} \mid \beta_{z_{dn}}) + \sum_{v=1}^{V} (\alpha_\beta - 1) \log \beta_{k,v} - \log \mathbf{B}(\bar{\alpha}_\beta) \\
&\propto \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbf{1}[z_{dn} = k] \log \mathrm{Cat}(w_{dn} \mid \beta_k) + \sum_{v=1}^{V} (\alpha_\beta - 1) \log \beta_{k,v} \\
&= \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbf{1}[z_{dn} = k] \sum_{v=1}^{V} w_{dn,v} \log \beta_{k,v} + \sum_{v=1}^{V} (\alpha_\beta - 1) \log \beta_{k,v}
\end{aligned}
$$

$$= \sum_{v=1}^{K} \log \beta_{k,v} \left[\alpha_\beta - 1 + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbf{1}[z_{dn} = k] w_{dn,v} \right]$$
$$\propto \log \text{Dirichlet}_V \left(\alpha_\beta + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbf{1}[z_{dn} = k] w_{dn,v} \right)$$

Thus:

$$\beta_k \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z} \sim \text{Dirichlet}_V \left(\alpha_\beta + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbf{1}[z_{dn} = k] w_{dn,v} \right) \tag{B.1}$$

This is equivalent to:

$$\beta'_{k,v} \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z} \sim \text{Gamma} \left(\alpha_\beta + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbf{1}[z_{dn} = k] w_{dn,v}, 1 \right) \tag{B.2}$$

$$\beta_{k,v} = \frac{\beta'_{k,v}}{\sum_{j=1}^{V} \beta'_{k,j}} \tag{B.3}$$

Derivation of PF Updates

$$\begin{aligned}
p(\beta_{k,v} \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z}) &\propto p(\mathbf{w}, \boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{z}) \\
&= \prod_{d=1}^{D} \prod_{v=1}^{V} p(w_{dv} \mid \theta_d, \beta_v) \prod_{k=1}^{K} \left[\prod_{v=1}^{V} p(\beta_{k,v} \mid s_\beta, r_\beta) \prod_{d=1}^{D} p(\theta_{dk} \mid s_\beta, r_\beta) \right] \\
&= \prod_{d=1}^{D} \prod_{v=1}^{V} \prod_{k=1}^{K} p(z_{dv,k} \mid \theta_{dk}, \beta_{v,k}) \cdot \\
&\quad \prod_{k=1}^{K} \left[\prod_{v=1}^{V} p(\beta_{k,v} \mid s_\beta, r_\beta) \prod_{d=1}^{D} p(\theta_{dk} \mid s_\beta, r_\beta) \right] \\
\log p(\beta_{k,v} \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z}) &\propto \sum_{d=1}^{D} \sum_{v=1}^{V} \sum_{k=1}^{K} \log p(z_{dv,k} \mid \theta_{dk}, \beta_{v,k}) + \\
&\quad \sum_{k=1}^{K} \left[\sum_{v=1}^{V} \log p(\beta_{k,v} \mid s_\beta, r_\beta) + \sum_{d=1}^{D} \log p(\theta_{dk} \mid s_\beta, r_\beta) \right]
\end{aligned}$$

$$
\begin{aligned}
&\propto \sum_{d=1}^{D} \log p(z_{dv,k} \mid \theta_{dk}, \beta_{v,k}) + \log p(\beta_{k,v} \mid s_\beta, r_\beta) \\
&= \sum_{d=1}^{D} [z_{dv,k} \log(\theta_{dk}) + z_{dv,k} \log(\beta_{v,k}) - \theta_{dk}\beta_{v,k} - \log(z_{dv,k}!)] + \\
&\quad s_\beta \log r_\beta - \log \Gamma(s_\beta) + (s_\beta - 1) \log \beta_{k,v} - r_\beta \beta_{k,v} \\
&\propto \sum_{d=1}^{D} [z_{dv,k} \log(\beta_{v,k}) - \theta_{dk}\beta_{v,k}] + (s_\beta - 1) \log \beta_{k,v} - r_\beta \beta_{k,v} \\
&= \log \beta_{k,v} \left[s_\beta - 1 + \sum_{d=1}^{D} z_{dv,k} \right] - \beta_{k,v} \left[r_\beta + \sum_{d=1}^{D} \theta_{dk} \right] \\
&\propto \log \text{Gamma} \left(s_\beta + \sum_{d=1}^{D} z_{dv,k}, r_\beta + \sum_{d=1}^{D} \theta_{dk} \right)
\end{aligned}
$$

This gives us:

$$
\beta_{k,v} \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z} \sim \text{Gamma} \left(s_\beta + \sum_{d=1}^{D} z_{dv,k}, r_\beta + \sum_{d=1}^{D} \theta_{dk} \right) \tag{B.4}
$$

Derivation of LDA/PF Hybrid Updates

$$
\begin{aligned}
p(\beta_k \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z}) &\propto p(\mathbf{w}, \boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{z}) \\
&= \prod_{k=1}^{K} p(\beta_k \mid \alpha_\beta) \prod_{d=1}^{D} \prod_{v=1}^{V} p(w_{dv} \mid \theta_d, \beta_v) \prod_{k=1}^{K} \prod_{d=1}^{D} p(\theta_{dk} \mid s_\beta, r_\beta) \\
&= \prod_{k=1}^{K} p(\beta_k \mid \alpha_\beta) \prod_{d=1}^{D} \prod_{v=1}^{V} \prod_{k=1}^{K} p(z_{dv,k} \mid \theta_{dk}, \beta_{v,k}) \prod_{k=1}^{K} \prod_{d=1}^{D} p(\theta_{dk} \mid s_\beta, r_\beta) \\
\log p(\beta_k \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z}) &\propto \sum_{k=1}^{K} \log p(\beta_k \mid \alpha_\beta) + \sum_{d=1}^{D} \sum_{v=1}^{V} \sum_{k=1}^{K} \log p(z_{dv,k} \mid \theta_{dk}, \beta_{v,k}) + \\
&\quad \sum_{k=1}^{K} \sum_{d=1}^{D} \log p(\theta_{dk} \mid s_\beta, r_\beta) \\
&\propto \log p(\beta_k \mid \alpha_\beta) + \sum_{d=1}^{D} \sum_{v=1}^{V} \log p(z_{dv,k} \mid \theta_{dk}, \beta_{v,k})
\end{aligned}
$$

$$
\begin{aligned}
&= \left[-\log \mathbf{B}(\alpha_\beta) + \sum_{v=1}^{V} (\alpha_\beta - 1) \log \beta_{k,v} \right] + \\
&\quad \sum_{d=1}^{D} \sum_{v=1}^{V} [z_{dv,k} \log \theta_{dk} + z_{dv,k} \log \beta_{k,v} - \theta_{dk} \beta_{k,v} - \log (z_{dv,k}!)] \\
&\propto \sum_{v=1}^{V} (\alpha_\beta - 1) \log \beta_{k,v} + \sum_{d=1}^{D} \sum_{v=1}^{V} [z_{dv,k} \log \beta_{k,v} - \theta_{dk} \beta_{k,v}] \\
&= \sum_{v=1}^{V} \log \beta_{k,v} \left[\alpha_\beta - 1 + \sum_{d=1}^{D} z_{dv,k} \right] - \beta_{k,v} \left[\sum_{d=1}^{D} \theta_{dk} \right] \\
&\propto \log \text{Gamma} \left(\alpha_\beta + \sum_{d=1}^{D} z_{dv,k}, \sum_{d=1}^{D} \theta_{dk} \right)
\end{aligned}
$$

This gives us:

$$
\beta'_{k,v} \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{z} \sim \text{Gamma} \left(\alpha_\beta + \sum_{d=1}^{D} z_{dv,k}, \sum_{d=1}^{D} \theta_{dk} \right) \tag{B.5}
$$

$$
\beta_{k,v} = \frac{\beta'_{k,v}}{\sum_{j=1}^{V} \beta'_{k,j}} \tag{B.6}
$$

Bibliography

Adams, R. P. and D. J. MacKay (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

Allan, J., R. Papka, and V. Lavrenko (1998). On-line new event detection and tracking. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 37–45.

Amatriain, X., P. Castells, A. de Vries, and C. Posse (2012). Workshop on recommendation utility evaluation: Beyond RMSE. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, pp. 351–352.

Andersen, R., C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz (2008). Trust-based recommendation systems: an axiomatic approach. In *Proceedings of the International World Wide Web Conference (WWW)*, pp. 199–208.

Arnold, B. C., E. Castillo, and J. M. Sarabia (1999). *Conditional specification of statistical models*. Springer.

Becker, H., M. Naaman, and L. Gravano (2010). Learning similarity metrics for event identification in social media. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 291–300.

Besag, J. (1975). Statistical analysis of non-lattice data. *The statistician*, 179–195.

Billiot, J.-M., J.-F. Coeurjolly, R. Drouilhet, et al. (2008). Maximum pseudolikelihood estimator for exponential family models of marked Gibbs point processes. *Electronic Journal of Statistics 2*, 234–264.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer New York.

Bishop, C. M. (1998). Latent variable models. In *Learning in graphical models*, pp. 371–403. Springer.

Blei, D., T. Griffiths, and M. Jordan (2010). The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM 57*(2), 1–30.

Blei, D. and J. Lafferty (2006). Dynamic topic models. In *Proceedings of the International Conference on Machine Learning (ICML)*, New York, NY, USA, pp. 113–120.

Blei, D. and J. Lafferty (2007). A correlated topic model of Science. *Annals of Applied Statistics 1*(1), 17–35.

Blei, D., A. Ng, and M. Jordan (2003, January). Latent Dirichlet allocation. *The Journal of Machine Learning Research 3*, 993–1022.

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

Blei, D. M. (2014). Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application 1*, 203–232.

Blei, D. M. and J. D. Lafferty (2009). Topic models. *Text mining: classification, clustering, and applications 10*(71), 34.

Brants, T., F. Chen, and A. Farahat (2003). A system for new event detection. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 330–337.

Canny, J. (2004). Gap: a factor model for discrete data. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 122–129.

Cao, N., J. Sun, Y.-R. Lin, D. Gotz, S. Liu, and H. Qu (2010). FacetAtlas: Multifaceted Visualization for Rich Text Corpora. *IEEE Transactions on Visualization and Computer Graphics 16*(6), 1172 – 1181.

Carlin, B. P. and N. G. Polson (1991). Inference for nonconjugate Bayesian models using the Gibbs sampler. *Canadian Journal of statistics 19*(4), 399–405.

Chakrabarti, D. and K. Punera (2011). Event summarization using tweets. *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM) 11*, 66–73.

Chang, J. and D. Blei (2009). Relational topic models for document networks. In *Artificial Intelligence and Statistics*.

Chang, J., J. Boyd-Graber, C. Wang, S. Gerrish, and D. Blei (2009). Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems (NIPS)*.

Chen, C.-h., W. K. Härdle, and A. Unwin (2007). *Handbook of data visualization*. Springer Science & Business Media.

Chen, Y., L. Wang, M. Dong, and J. Hua (2009). Exemplar-based visualization of large document corpus. *IEEE Transactions on Visualization and Computer Graphics 15*(6), 1161–1168.

Cleveland, W. S. (1993). *Visualizing data*. Hobart Press.

Cleveland, W. S. and R. McGill (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association 79*(387), 531–554.

Condry, N. (2016). Meaningful models: Utilizing conceptual structure to improve machine learning interpretability. *arXiv preprint arXiv:1607.00279*.

Crandall, D., D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri (2008). Feedback effects between similarity and social influence in online communities. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 160–168.

Cremonesi, P., Y. Koren, and R. Turrin (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, pp. 39–46.

Das, K., J. Schneider, and D. B. Neill (2008). Anomaly pattern detection in categorical datasets. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 169–176.

Das Sarma, A., A. Jain, and C. Yu (2011). Dynamic relationship and event discovery. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 207–216.

Diener, E. (2000). Subjective well-being: The science of happiness and a proposal for a national index. *American psychologist 55*(1), 34.

Dou, W., X. Wang, D. Skau, W. Ribarsky, and M. 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*, pp. 93–102. IEEE.

Du, N., L. Song, H. Woo, and H. Zha (2013). Uncover topic-sensitive information diffusion networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 229–237.

Fayyad, U. M., A. Wierse, and G. G. Grinstein (2002). *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann.

Fazeli, S., B. Loni, A. Bellogin, H. Drachsler, and P. Sloep (2014). Implicit vs. explicit trust in social matrix factorization. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, pp. 317–320.

Fung, G. P. C., J. X. Yu, P. S. Yu, and H. Lu (2005). Parameter free bursty events detection in text streams. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 181–192. VLDB Endowment.

Gao, W., P. Li, and K. Darwish (2012). Joint topic modeling for event summarization across news and social media streams. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pp. 1173–1182.

Gardener, M. J., J. Lutes, J. Lund, J. Hansen, D. Walker, E. Ringger, and K. Seppi (2010). The topic browser: An interactive tool for browsing topic models. In *Proceedings of the Workshop on Challenges of Data Visualization (in conjunction with NIPS)*.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2014). *Bayesian data analysis*, Volume 2. Chapman & Hall/CRC Boca Raton, FL, USA.

Gerrish, S. and D. Blei (2010). A language-based approach to measuring scholarly impact. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ghahramani, Z. and M. J. Beal (2001). Propagation algorithms for variational bayesian learning. *Advances in Neural Information Processing Systems (NIPS)*, 507–513.

Glynn, T. J. (1981). Psychological sense of community: Measurement and application. *Human Relations 34*(9), 789–818.

Golbeck, J. and J. Hendler (2006, January). FilmTrust: Movie recommendations using trust in web-based social networks. *TOIT 6*(4), 497–529.

Gopalan, P., L. Charlin, and D. M. Blei (2014a). Content-based recommendation with Poisson factorization. *Advances in Neural Information Processing Systems (NIPS)*.

Gopalan, P., J. M. Hofman, and D. M. Blei (2015). Scalable recommendation with hierarchical Poisson factorization. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 326–335.

Gopalan, P., F. J. R. Ruiz, R. Ranganath, and D. M. Blei (2014). Bayesian nonparametric Poisson factorization for recommendation systems. *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Gopalan, P. K., L. Charlin, and D. Blei (2014b). Content-based recommendations with Poisson factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (Eds.), *Advances in Neural Information Processing Systems (NIPS)*, pp. 3176–3184. Curran Associates, Inc.

Gretarsson, B., J. O'donovan, S. Bostandjiev, T. Höllerer, A. Asuncion, D. Newman, and P. Smyth (2012). Topicnets: Visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST) 3*(2), 23.

Grimmer, J. (2013). Comment: Evaluating model performance in fictitious prediction problems. *Journal of the American Statistical Association*.

Guille, A., H. Hacid, C. Favre, and D. A. Zighed (2013, July). Information diffusion in online social networks: A survey. *SIGMOD Record 42*(2), 17–28.

Guo, G., J. Zhang, D. Thalmann, and N. Yorke-Smith (2014). ETAF: An extended trust antecedents framework for trust prediction. In *Proceedings of the International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, pp. 540–547.

Guo, G., J. Zhang, and N. Yorke-Smith (2015). TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 123–129.

Guralnik, V. and J. Srivastava (1999). Event detection from time series data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 33–42.

Han, S., X. Liao, and L. Carin (2013). Integrated non-factorized variational inference. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2481–2489.

Havre, S., B. Hetzler, and L. Nowell (2000). Themeriver(tm): In search of trends, patterns, and relationships. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*.

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika 58*(1), 83–90.

Hearst, M. A. (2008). UIs for faceted navigation: Recent advances and remaining open problems. *Workshop on Human-Computer Interaction and Information Retrieval*.

Herlocker, J. L., J. A. Konstan, and J. Riedl (2000). Explaining collaborative filtering recommendations. In *Proceeding of the ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*, pp. 241–250.

Hoffman, M., D. Blei, and F. Bach (2010). On-line learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*.

Hoffman, M. D. and D. M. Blei (2015). Structured stochastic variational inference. In *Artificial Intelligence and Statistics*.

Hoffman, M. D., D. M. Blei, C. Wang, and J. Paisley (2013). Stochastic variational inference. *The Journal of Machine Learning Research 14*(1), 1303–1347.

Hu, Y., J. Boyd-Graber, B. Satinoff, and A. Smith (2014). Interactive topic modeling. *Machine learning 95*(3), 423–469.

Jackoway, A., H. Samet, and J. Sankaranarayanan (2011). Identification of live news events using twitter. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pp. 25–32. ACM.

Jamali, M. and M. Ester (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, pp. 135–142.

Johnstone, J. and E. Katz (1957, May). Youth and popular music: A study in the sociology of taste. *Journal of Sociology 62*(6), 563–568.

Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, and L. K. Saul (1999, November). An introduction to variational methods for graphical models. *Machine Learning 37*(2), 183–233.

Koller, D. and N. Friedman (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Koren, Y., R. Bell, and C. Volinsky (2009). Matrix factorization techniques for recommender systems. *IEEE Computer 42*, 30–37.

Kullback, S. (1997). *Information theory and statistics*. Courier Corporation.

Kumaran, G. and J. Allan (2004). Text classification and named entities for new event detection. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 297–304.

Lau, J. H., N. Collier, and T. Baldwin (2012). On-line trend analysis with topic models:\# twitter trends detection topic model online. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pp. 1519–1534.

Lee, B., G. Smith, G. G. Robertson, M. Czerwinski, and D. S. Tan (2009). FacetLens: exposing trends and relationships to support sensemaking within faceted datasets. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 1293–1302.

Lee, D. D. and H. S. Seung (2000). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 556–562.

Leskovec, J., A. Singh, and J. Kleinberg (2006). Patterns of influence in a recommendation network. In *Proceedings of the The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 380–389.

Li, Z., B. Wang, M. Li, and W.-Y. Ma (2005). A probabilistic model for retrospective news event detection. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 106–113.

Linderman, S. W. and R. P. Adams (2014). Discovering latent network structure in point process data. *arXiv preprint arXiv:1402.0914*.

Liu, X., R. Troncy, and B. Huet (2011). Using social media to identify events. In *Proceedings of the ACM SIGMM International Workshop on Social Media (WSM)*, pp. 3–8.

Loiacono, D., A. Lommatzsch, and R. Turrin (2014). An analysis of the 2014 recsys challenge. In *RecSysChallenge*, pp. 1.

Ma, H., I. King, and M. R. Lyu (2009). Learning to recommend with social trust ensemble. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 203–210.

Ma, H., H. Yang, M. R. Lyu, and I. King (2008). SoRec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pp. 931–940.

Ma, H., D. Zhou, C. Liu, M. R. Lyu, and I. King (2011). Recommender systems with social regularization. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 287–296.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Martin, A. D. and K. M. Quinn (2002). Dynamic ideal point estimation via Markov chain Monte Carlo for the US Supreme Court, 1953–1999. *Political Analysis 10*(2), 134–153.

Massa, P. and P. Avesani (2007). Trust-aware recommender systems. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, pp. 17–24.

Mathioudakis, M., N. Bansal, and N. Koudas (2010). Identifying, attributing and describing spatial bursts. *Proceedings of the International Conference on Very Large Data Bases (VLDB) 3*(1-2), 1091–1102.

Mimno, D. and D. Blei. (2011). Bayesian checking of topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mimno, D., H. M. Wallach, E. Talley, M. Leenders, and A. McCallum (2011). Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 262–272. Association for Computational Linguistics.

Neill, D. B., A. W. Moore, M. Sabhnani, and K. Daniel (2005). Detection of emerging space-time clusters. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 218–227.

Newman, D., A. Asuncion, C. Chemudugunta, V. Kumar, P. Smyth, and M. Steyvers (2006). Exploring large document collections using statistical topic models. In *KDD-2006 Demo Session*.

Newman, D., A. Asuncion, P. Smyth, and M. Welling (2007). Distributed inference for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*.

Newman, D., J. H. Lau, K. Grieser, and T. Baldwin (2010). Automatic evaluation of topic coherence. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic books.

Paisley, J., D. Blei, and M. I. Jordan (2014). Bayesian nonnegative matrix factorization with stochastic variational inference. *Handbook of Mixed Membership Models and Their Applications. Chapman and Hall/CRC*.

Paul, M. J. and M. Dredze (2012). A model for mining public health topics from twitter. *Health 11*, 16–6.

Peng, W., C. Perng, T. Li, and H. Wang (2007). Event summarization for system management. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1028–1032.

Purushotham, S., Y. Liu, and C.-C. J. Kuo (2012). Collaborative topic regression with social matrix factorization for recommendation systems. *CoRR abs/1206.4684*.

Quinn, K. M., B. L. Monroe, M. Colaresi, M. H. Crespin, and D. R. Radev (2010). How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science 54*(1), 209–228.

Ranganath, R., S. Gerrish, and D. M. Blei (2014). Black box variational inference. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 814–822.

Reuter, T. and P. Cimiano (2012). Event-based classification of social media streams. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, pp. 22. ACM.

Rosen-Zvi, M., T. Griffiths, M. Steyvers, and P. Smith (2004). The author-topic model for authors and documents. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 487–494.

Sakaki, T., M. Okazaki, and Y. Matsuo (2010). Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the International World Wide Web Conference (WWW)*, pp. 851–860.

Salakhutdinov, R. and A. Mnih (2007). Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1257–1264.

Sayyadi, H., M. Hurst, and A. Maykov (2009). Event detection and tracking in social streams. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*.

Schein, A., J. Paisley, D. M. Blei, and H. Wallach (2015). Bayesian Poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1045–1054.

Shang, S., P. Hui, S. R. Kulkarni, and P. W. Cuff (2011). Wisdom of the crowd: Incorporating social influence in recommendation models. In *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 835–840.

Shmueli, G. (2010). To explain or to predict? *Statistical science*, 289–310.

Singh, P., G. Singh, and A. Bhardwaj (2014). Ranking approach to recsys challenge. In *RecSysChallenge*, pp. 19.

Steyvers, M. and T. Griffiths (2006). Probabilistic topic models. In T. Landauer, D. McNamara, S. Dennis, and W. Kintsch (Eds.), *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum.

Strobelt, H., D. Oelke, C. Rohrdantz, A. Stoffel, D. A. Keim, and O. Deussen (2009). Document Cards: A top trumps visualization for documents. *IEEE Transactions on Visualization and Computer Graphics 15*(6), 1145.

Su, X. and T. M. Khoshgoftaar (2009, January). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 4.

Teh, Y., M. Jordan, M. Beal, and D. Blei (2007). Hierarchical Dirichlet processes. *Journal of the American Statistical Association 101*(476), 1566–1581.

Teh, Y., D. Newman, and M. Welling (2006). A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*.

Telea, A. C. (2014). *Data visualization: principles and practice*. CRC Press.

Thai, V. and S. Handschuh (2010). FacetLens: exposing trends and relationships to support sensemaking within faceted datasets. *Proceedings of the First International Workshop on Intelligent Visual Interfaces for Text Analysis*.

Tufte, E. R. (1991). Envisioning information. *Optometry & Vision Science 68*(4), 322–324.

Tufte, E. R. and P. Graves-Morris (1983). *The visual display of quantitative information*, Volume 2. Graphics press Cheshire, CT.

Tufte, E. R. and D. Robins (1997). *Visual explanations*. Graphics Cheshire.

Tukey, J. W. (1977). Exploratory data analysis.

Van Ham, F., M. Wattenberg, and F. B. Viégas (2009). Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics 15*(6), 1169–1176.

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

Volz, I. P. (2006). The impact of online music services on the demand for stars in the music industry. In *Proceedings of the International World Wide Web Conference (WWW)*, pp. 659–667.

Wainwright, M. J. and M. I. Jordan (2008, January). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning 1*(1-2), 1–305.

Wallach, H. M., I. Murray, R. Salakhutdinov, and D. Mimno (2009). Evaluation methods for topic models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1105–1112.

Wang, C. and D. M. Blei (2013). Variational inference in nonconjugate models. *The Journal of Machine Learning Research*.

Wang, X., C. Zhai, X. Hu, and R. Sproat (2007). Mining correlated bursty topic patterns from coordinated text streams. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 784–793. ACM.

Wattenberg, M. and F. B. Viégas (2008). The word tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics 14*(6), 1221–1228.

Weiss, G. M. and H. Hirsh (1998). Learning to predict rare events in event sequences. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 359–363.

Wilkinson, L. (2005). *The Grammar of Graphics* (Second ed.). Springer Science+Business Media, Inc.

Yang, B., Y. Lei, D. Liu, and J. Liu (2013). Social collaborative filtering by trust. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2747–2753.

Ye, M., X. Liu, and W.-C. Lee (2012). Exploring social influence for recommendation: A generative model approach. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 671–680.

Zhang, Y., J. Callan, and T. Minka (2002). Novelty and redundancy detection in adaptive filtering. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 81–88.

Zhao, Q., P. Mitra, and B. Chen (2007). Temporal and information flow based event detection from social text streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 7, pp. 1501–1506.

Zhao, T., J. McAuley, and I. King (2014). Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pp. 261–270.

Zhao, W. X., R. Chen, K. Fan, H. Yan, and X. 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*, pp. 43–47. Association for Computational Linguistics.