Topic Modeling: A Complete Introductory Guide

Jacob Su Wang SHREKWANG@UTEXAS.EDU
Department of Statistics and Data Science
Department of Linguistics
University of Texas
Austin, TX 78712, USA

Editor: Lynn Bender

Abstract

I present an in-detail introduction to *Topic Models* (TM), a family of probabilistic models for (mainly) document modeling. I introduce and motivate the model, and illustrate its applications in *Natural Language Processing* (NLP), with the particular focus on a thorough description and derivation of the common inference algorithms proposed for TMs. I also compare the algorithms, overviewing various evaluation methods, and provide practical suggestions. Finally I look at a few popular extensions of TM before concluding.

Keywords: Topic Modeling, Natural Language Processing, Bayesian Inference.

1. Introduction

TMs are a class of generative models which aims at capturing the latent distributions that underpin given observational data (e.g. word distribution in a collection of documents). In the context of NLP, TMs are particularly useful for discovering the statistical regularities hidden in textual data in supervised/semi-supervised/unsupervised settings. Although TM was originally developed to handle texts, as a general framework in machine learning, it has potential applications in modeling various types of data (e.g. bioinformatics and computer vision) (Blei, 2012).

The tutorial is organized as follows: I start by introducing the general ideas of TM (Section 1) and its applications in NLP (Section 2). I then present an in-detail description of the model in its original form (Blei, Ng, and Jordan, 2003; Steyvers and Griffiths, 2007) and compare the two most common inference algorithms. Finally I close with a brief overview of evaluation methods (Section 3), and move on to introducing a few popular extensions (Section 4). I summarize and conclude the tutorial in Section 5. For understanding the tutorial thoroughly, I assume the reader is at least familiar with basic calculus, linear algera and probability theory. No further background is needed beyond that.

General idea. To aid understanding, one may consider an imaginary process in which a human author creates a set of documents $\mathcal{D}$ word by word. For each word in a document $d \in \mathcal{D}$, she first selects a topic, then picks a word given the topic. This process is often termed a *generative process* or *generative story* (Roller and Schulte im Walde, 2013; Wang, Roller, and Erk, 2017). Taking the outcome of this process (i.e. $\mathcal{D}$) as the observational input to the TM, our objective is to model the human author: to infer the underlying parameters

with which $\mathcal{D}$ are generated, for the purposes of (i) estimating a soft topical classification for the documents in $\mathcal{D}$, and (ii) inferring topic distributions for new documents. The learned parameters, beside revealing the underlying statistical regularities in the texts, may also other aid other up/downstream tasks (cf. Section 2, 4).

2. Applications

Document classification. While showing strong performance in various document classification tasks (Blei, Ng, and Jordan, 2003; Steyvers and Griffiths, 2007; Hoffman, Blei, and Bach, 2010), TM in its original form makes three false but convenient assumptions (Blei, 2012) (i) exchangeability of words in documents (i.e. bag-of-words assumption), (ii) exchangeability of documents in a corpus (i.e. bag of documents assumption), and (iii) parametric topics (i.e. fixed-number-of-topics assumption). Extensions of the model take into account word dependencies in the same document (Wallach, 2006; Griffiths, Steyvers, Blei, and Tenenbaum, 2005), addressing (i); model bodies of documents as a time series (i.e. documents created in some chronical order) (Blei and Lafferty, 2006), addressing (ii); and allow TM to have unlimited number of topics (Teh, Jordan, Beal, and Blei, 2006; Blei, Griffiths, and Jordan, 2010), addressing (iii).

Other applications. Beyond document classification, TM has demonstrated its potential in distributed semantic representation (Dinu and Lapata, 2010) by garnering better correlation when extended as a multimodal model (in particular visual data) (Andrews et al., 2009; Roller and Schulte im Walde, 2013). More recently, it has also been shown to model word learning effectively in low data conditions (Wang et al., 2017). I do not cover the applications of TM in bioinformatics and computer vision, but defer the reader to references in Blei (2012).

Setting the main focus on NLP, I walk through a selected set of exciting new extensional architectures of TM in more details in Section 4.

3. Model

I start by a complete description of the TM in its original formulation (Section 3.1), with an exposition on the two major branches of learning algorithms (CGS in Section 3.2, VB in section 3.3) which are compared in Section 3.4. Finally I close the section with an overview of evaluation methods (Section 3.5).

3.1 General Description

In this section, I formalize TM and motivate the inference algorithms described in the following sections (Section 3.2,3).

Generative process. To begin, let us first formalize the generative story of TM and introduce some notations.

1. For each document $d \in \mathcal{D}$, draw a multinomial θ^d from a Dirichlet prior with parameters α, over topics $\mathcal{T}$

$$\theta \sim Dir(\alpha)$$

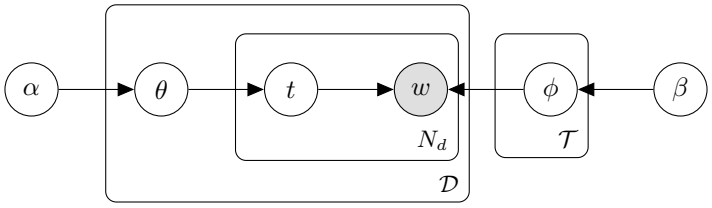

Figure 1: Plate diagram for Topic Model

2. For the ith word slot in d ($i \in N_d$, where N_d is the number of words in d),

   (a) Draw a topic t_i from the multinomial parameterized by θ^d, where

   $$t \sim Mult(\theta^d)$$

   (b) Draw a word w_i from a multinomial parameterized by ϕ^{t_i}, where

   $$w_i \sim Mult(\phi^{t_i})$$

   and $\phi \sim Dir(\beta)$.

The process is illustrated with a plate diagram[1] (Koller and Friedman, 2009) in Figure 1. For the reader's convenience in operationalization, I now elaborate on the details of the notational scheme.

- α: corpus-level parameters, a length-$|T|$ vector of Dirichlet parameters $\langle \alpha_1, \ldots, \alpha_T \rangle$, where and T is the set of all topics.
- β: corpus-level parameters, a $|T| \times |V|$ matrix where each row t is a vector of Dirichlet parameters $\langle \beta_1^t, \ldots, \beta_V^t \rangle$, where $|V|$ is the size of the vocabulary.
- θ: a length-T vector of document-level multinomial parameters $\langle \theta_1, \ldots, \theta_T \rangle$, where, for any $\theta^d, d \in \mathcal{D}$, $\sum_{t=1}^{T} \theta_t^d = 1$.
- ϕ: a length-V vector of corpus-level multinomial parameters $\langle \phi_1, \ldots, \phi_V \rangle$, where, for any topic $t \in T$, $\sum_{w=1}^{V} \phi_w^t = 1$.

Parameter estimation. As is stated in Section 1, our objective is to estimate the model parameters[2] that have generated the observation of some documents $\mathcal{D}$, i.e. maximizing the

1. In Blei et al. (2003), β plays the role of ϕ here, and directly links to w. However, it is a common practice (Steyvers and Griffiths, 2007) to also give word distributions a prior such that smoothing tuning on them is possible (analogously, α is the smoothing prior for topics). This will become important in comparing different training algorithms (Asuncion et al., 2008) (Section 3.4), therefore I follow the latter convention.
2. Note that, different algorithms target different parameters, which will become clear in following sections (Section 3.2,3).

following likelihood[3], which factorizes as follows:

$$\begin{aligned} p(\mathcal{D}|\alpha,\beta) &= \prod_{d=1}^{\mathcal{D}} p(d|\alpha,\beta) \\ &= \prod_{d=1}^{\mathcal{D}} \int\int p(\theta^d|\alpha)p(\phi|\beta)p(d|\theta^d,\phi)d\theta^d d\phi \\ &= \prod_{d=1}^{\mathcal{D}} \int\int p(\theta^d|\alpha)p(\phi|\beta)\prod_{i=1}^{N_d} p(t_i|\theta^d)p(w_i|\phi^{t_i})d\theta^d d\phi \end{aligned} \tag{3.1}$$

where N_d is the number of words in document d. Eq. 3.1, however, is shown to be intractable, due to the coupling between the latent variables θ and ϕ (Blei et al., 2003). Specifically, in the integration over $\{\theta, \phi\}$, the choice of the word distribution ϕ^t depends on the choice of the topic t, which further depends on the topic distribution θ^d.

Because of the intractibility in exact inference, approximation algorithms have been applied to estimating the posterior, most commonly *Markov Chain Monte Carlo* based *Collapsed Gibbs Sampling*[4] (CGS) (Steyvers and Griffiths, 2007), which estimates parameters by approximating the "true" distribution through a stationary random walk (Section 3.2) and optimization based *Variational Bayes* (VB) (Blei et al., 2003), which maximizes the likelihood of observed data by searching for the optimal parameters (Section 3.3). Finally note that the topics do not have intrinsic interpretation — they are simply labels that are exchangeable[5] (Steyvers and Griffiths, 2007, Section 4). I elaborate on the algorithms in the following subsections, and also provide supporting definitions, theorems and their proofs in the Appendix.

3.2 Collapsed Gibbs Sampling (CGS)

Overview. In CGS, we learn a topical assignment to a corpus $\mathcal{D}$ such that each word token in each document is associated with a topic token, where word tokens of the same type may have different topics, and a topic may be assigned to different word tokens. From this assignment, we can then collect (i) $p(w,t)$: the joint distribution of words and topics, and (ii) $p(t,d)$: the joint distribution of topics and documents, and thereby obtain an estimation for parameters $\{\theta, \phi\}$, because

$$\theta^d = p(t|d) = \frac{p(t,d)}{p(d)} \tag{3.2}$$

$$\phi^t = p(w|t) = \frac{p(w,t)}{p(t)} \tag{3.3}$$

3. The derivation should be easier to follow by referring to Figure 1.
4. In inference, the model integrate over hidden parameters, hence the term "collapsed".
5. Note this is only true in the "base" version of TMs which are described here. Topics can be grounded with supervised, directly interpretable labels (Ramage et al., 2009).

Algorithm 1 Gibbs Sampling (GENERAL)

1: **Initialize** (randomly) all variables t_j
2: **for** k iterations **do**
3: **for** each t_j **do**
4: Sample a topic value $\underline{t}$ from $p(t|\boldsymbol{t}_{-j}, \cdot)$
5: Reset $t_j = \underline{t}$
6: **end for**
7: **end for**
8: Estimate parameters from the approximating distribution $\hat{p}(t|\mathcal{D}; \alpha, \beta)$

In particular, note that in CGS the Dirichlet priors $\{\alpha, \beta\}$ are pre-set to be symmetric[6]. The likelihood (i.e. Eq. 3.1) now becomes

$$p(\mathcal{D}|\theta, \phi) = \prod_{d=1}^{\mathcal{D}} p(d|\theta^d, \phi) = \prod_{d=1}^{\mathcal{D}} \prod_{i=1}^{N_d} p(t_i|\theta^d) p(w_i|\phi^{t_i}) \tag{3.4}$$

and recall the idea is to estimate the posterior distribution $p(\theta, \phi|\mathcal{D})$ by approximating it with an empirical distribution obtained from a topical assignment on $\mathcal{D}$, where topics are generated by the following distribution

$$p(t_j|\mathcal{D}, \theta, \phi), \quad j \in N_d \times |\mathcal{D}| \tag{3.5}$$

As Eq. 3.5 is still woefully intractable[7] as Eq. 3.1,3.4, we apply *Gibbs Sampling* technique to find $\hat{p}(t_j|\mathcal{D}, \theta, \phi)$ to approximate it. The procedure is described in Algorithm 1[8].

Essentially, Gibbs sampling performs a random walk on the observed data, where the samples gradually accumulate to form an empirical distribution that approximates the real distribution from which the observations are generated[9]. In the initial stage of the process, the samples are of poor quality, therefore typically we run a number of passes on the data before estimating parameters. This is called a *burn-in* period. Typically the burn-in period lasts 500-1,000 iterations (Yao et al., 2009). Concisely, in the context of topic modeling, on the one hand words that tend to cooccur gradually form clusters under topics, and on the other, in any given document, a subset of topics gradually gain statistical dominance, drawing on the word statistics in the document. This will become clear soon.

6. i.e. $\alpha_t = \alpha, \forall t \in T$, and $\beta_{tw} = \beta, \forall t \in T, w \in V$. This is reasonable, because for a given corpus, unless we have strong belief that some topics/words should be favored over others in the corpus, they are assumed to have symmetric priors (Steyvers and Griffiths, 2007). They are, of course, learnable if needed (Blei et al., 2003, Section 5.3).
7. It is easy to see the variables $\{t_j \mid j \in N_d \times |\mathcal{D}|\}$ are in coupling relationship and thus covary in complicated ways.
8. In the algorithm, $p(t|\boldsymbol{t}_{-j}, \cdot)$ denotes the conditional probability of $t \in \mathcal{T}$ given the assignments of all the topic variables *except for* t_j. "$\cdot$" refers to all other known or observed information (e.g. word tokens and pre-set priors α, β).
9. The Gibbs sampling works because of the *Ergodic Theorem* for *Markov Chains*. For more details, cf. Appendix A and B.

Approximation by Gibbs. Based on the foregoing discussion, the core distribution in parameter estimation (elaborated from Eq. 3.5), concretely, is as follows:

$$\begin{aligned} p(t_i = j|\boldsymbol{t}_{-i}, w_i, d, \cdot) &= \frac{1}{Z} \cdot p(w_i|t_i)p(t_i|d) \\ &= \frac{1}{Z} \cdot \frac{C^{VT}_{w_i,j} + \beta}{\sum_{w=1}^{V} C^{VT}_{w,j} + V\beta} \frac{C^{DT}_{d,j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d,t} + T\alpha} \end{aligned} \tag{3.6}$$

where the normalizing constant $Z = \sum_T p(w_i|t_i)p(t_i|d)$. w_i, t_i are the ith word-topic pairs in d; $\boldsymbol{t}_{-i}$ is the set of all topic variables in d except for t_i; '$\cdot$' denotes all the known information (cf. footnote 3). C^{VT}, C^{DT} are $|V| \times |T|$ and $|D| \times |T|$ count matrices (e.g. $C^{VT}_{w_i,j}$ is the count of word tokens that are of the same word type as w_i (in d) that are assigned topic j). To handle out-of-vocabulary tokens, α, β in the numerators and $V\beta, T\alpha$ in the denominators are added for *Laplace Smoothing* (Jurafsky and Martin, 2008, Section 4.5.1), licensed by *Jeffrey's Conditioning* (Shafer, 1981).

To understand how Eq. 3.6 works in the burn-in period, consider the distributions $p(w|t)$ and $p(t|d)$. For $p(w|t)$, the words that are more frequently assigned to t in initialization will tend to cluster under the topic. On the other hand, for $p(t|d)$, topics that are more frequent in d in initialization tend to be assigned to words in the document. Consequently, for any single document d, statistically prominent words in d will fall under that subset of topics, and the words of the same type elsewhere (i.e. in other documents) will in turn more likely to cluster under the same subset of topics. Eventually, the overall topical assignments on $\mathcal{D}$ stabalizes, with cooccurring words fall under same topics, and each document has relatively higher probabilities for a (usually small)[10] subset of topics.

With Eq. 3.6, we train the model following the procedure described in Algorithm 2, where $\{\hat{\theta}, \hat{\phi}\}$ are estimated parameters of the approximating distribution for the parameters of the true distribution, i.e. $\{\theta, \phi\}$. $\{\hat{\theta}, \hat{\phi}\}$ define topic/word distributions on $\mathcal{D}$, but to infer distributions for new documents $\mathcal{D}_{new}$, they are a starting point for further sets of CGS runs: Yao et al. (2009) propose three alternatives (each has its strengths and weaknesses): (i) *Gibbs1*: Rerun CGS on $\mathcal{D} \cup \mathcal{D}_{new}$; (ii) *Gibbs2*: Rerun CGS while holding the topical assignments on $\mathcal{D}$ fixed; and (iii) *Gibbs3*: Run CGS on $\mathcal{D}_{new}$ independently.

3.3 Variational Bayes (VB)

Overview. VB is an optimization algorithm that attempts to maximize the likelihood of the observed data (i.e. Eq. 3.1).

$$\begin{aligned} \{\alpha^*, \beta^*\} &= \underset{\{\alpha,\beta\}}{\text{argmax}}\ p(\mathcal{D}|\alpha, \beta) \\ &= \underset{\{\alpha,\beta\}}{\text{argmax}} \int\int \sum_{\boldsymbol{t}} p(\mathcal{D}; \theta, \boldsymbol{t}, \phi|\alpha, \beta) d\theta d\phi \end{aligned} \tag{3.7}$$

$$\text{where} \quad p(\mathcal{D}; \theta, \boldsymbol{t}, \phi|\alpha, \beta) = p(\phi|\beta) \prod_{d=1}^{\mathcal{D}} p(\theta^d|\alpha) p(\boldsymbol{t}^d|\theta^d) p(\boldsymbol{w}^d|\phi^{\boldsymbol{t}^d})$$

10. The resulting distribution over topics in documents is subject to the setting of smoothing hyperparameter α. If $\alpha < 1$, then the probability mass tends to concentrate on a few topics, and if $\alpha > 1$, it is more likely to spread out over topics. Similarly for β in word distributions (Steyvers and Griffiths, 2007, Section 3).

Algorithm 2 Gibbs Sampling (specific)

Initialize (randomly) topic assignments to all words in $\mathcal{D}$
▷ Burn-In
for k iterations **do**
  for $d \in \mathcal{D}$ **do**
    for $w_i \in d$ **do**
      Sample topic t from Eq. 3.5
      Reassign the topic of w_i to t
    end for
  end for
end for
▷ Parameter Estimation
Estmiate $\hat{\theta}$, where $\hat{\theta}_t^d = p(t|d) = \frac{C_{d,t}^{DT} + \alpha}{\sum_{t'=1}^{T} C_{d,t'}^{DT} + T\alpha}$
Estimate $\hat{\phi}$, where $\hat{\phi}_w^t = p(w|t) = \frac{C_{w,t}^{VT} + \beta}{\sum_{w'=1}^{V} C_{w',j}^{VT} + V\beta}$

where $\boldsymbol{t}$ are the set of all topic assignments for $\mathcal{D}$, $\boldsymbol{t}^d$ are topic assignments for a document d, and $\boldsymbol{w}^d$ are the tokens in d. As we know, due to the coupling between θ and ϕ, the optimization problem is intractable (Section 3.1). To decouple the variables, we disentagle their dependence. Specifically, we assign parameters over $\{\theta, \boldsymbol{t}, \phi\}$ with their own *variational parameters* (i.e. variational priors) $\{\gamma, \zeta, \eta\}$, with the objective to find the setting of $\{\gamma, \zeta, \eta\}$ such that q is close to the distribution of interest (i.e. p in Eq. 3.7)

$$q(\theta, \boldsymbol{t}, \phi|\gamma, \zeta, \eta) = q(\phi|\eta) \prod_{d=1}^{\mathcal{D}} q(\theta^d|\gamma^d) q(\boldsymbol{t}^d|\zeta^d) \tag{3.8}$$

In plate diagram, this is illustrated in Figure 2. In addition, as a notational clarification[11] for the convenience of implementation:

- γ: a $|T|$-dimensional vector, updated for each document.
- ζ: a $|T| \times |V|$ matrix, updated at each token in a document.
- η: a $|T| \times |V|$ matrix, updated at each pass of a corpus.

The ELBO. In VB, instead of optimizing the objective parameters directly, we formulate an *Evidence LOwer Bound* (ELBO) (Hoffmann and Johnson, 2016) of log $p(\mathcal{D}|\alpha, \beta)$. It can be shown that maximizing the ELBO is (i) equivalent to minimizing the *Kullback-Leibler divergence* (KL-divergence) between p and q (Blei et al., 2003, Section 5.2), and (ii) corresponds to finding the optimal parameters for our MLE objective (i.e. Eq. 3.7) (Blei et al., 2003, Section 5.3). Concretely, the ELBO, notated $\mathcal{L}(\gamma, \zeta, \eta; \alpha, \beta)$, is derived[12] as

11. **NB**: The row-column indices are subscripted. Any exception will be specially noted.
12. For details on *Jensen's Inequality* and how it is applied to construct ELBO, see Appendix C.

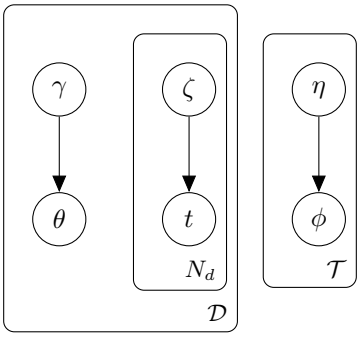

Figure 2: Variable Decoupling with Variational Bayes

follows[13]:

$$\begin{aligned}
\log p(\mathcal{D}|\alpha,\beta) &= \log \int\int \sum_{\boldsymbol{t}} p(\mathcal{D};\theta,\boldsymbol{t},\phi|\alpha,\beta)d\theta d\phi && \text{Eq. 3.7} \\
&= \log \int\int \sum_{\boldsymbol{t}} \frac{p(\mathcal{D};\theta,\boldsymbol{t},\phi|\alpha,\beta)q(\theta,\boldsymbol{t},\phi)}{q(\theta,\boldsymbol{t},\phi)}d\theta d\phi \\
&= \log \mathbb{E}_q\left[\frac{p(\mathcal{D};\theta,\boldsymbol{t},\phi|\alpha,\beta)}{q(\theta,\boldsymbol{t},\phi)}\right] \\
&\geq \mathbb{E}_q\left[\log \frac{p(\mathcal{D};\theta,\boldsymbol{t},\phi|\alpha,\beta)}{q(\theta,\boldsymbol{t},\phi)}\right] && \text{by Jensen's Inequality} \\
&= \mathbb{E}_q[\log p(\mathcal{D};\theta,\boldsymbol{t},\phi|\alpha,\beta)] - \mathbb{E}_q[\log q(\theta,\boldsymbol{t},\phi)] && \text{i.e. } -D(q||p) \\
&= \mathcal{L}(\gamma,\zeta,\eta;\alpha,\beta) && (3.9)
\end{aligned}$$

Clearly, maximizing $\mathcal{L}$ with respect to the variational prameters is equivalent to minimizing the KL-divergence between the variational distribution and the true distribution. Now we analyze the ELBO. For the convenience of derivation, $\mathcal{L}$ is factorized, from the penultimate

13. **NB**: In the following the variational parameters $\{\gamma,\zeta,\eta\}$ are omitted for readability.

formula in Eq. 3.9 (and Eq. 3.7,8), as follows:

$$
\begin{aligned}
&\mathcal{L}(\gamma, \zeta, \eta; \alpha, \beta) = \\
&\qquad \mathbb{E}_q[\log\, p(\phi|\beta)] && \text{(i)}\\
&+ \sum_{d=1}^{\mathcal{D}} \Big(\mathbb{E}_q[\log\, p(\theta^d|\alpha)] && \text{(ii)}\\
&+ \quad \mathbb{E}_q[\log\, p(\boldsymbol{t}^d|\theta^d)] && \text{(iii)}\\
&+ \quad \mathbb{E}_q[\log\, p(\boldsymbol{w}^d|\phi^{\boldsymbol{t}^d})] \Big) && \text{(iv)}\\
&- \quad \mathbb{E}_q[\log\, q(\phi)] && \text{(v)}\\
&- \sum_{d=1}^{\mathcal{D}} \Big(\mathbb{E}_q[\log\, q(\theta^d)] && \text{(vi)}\\
&- \quad \mathbb{E}_q[\log\, q(\boldsymbol{t}^d)] \Big) && \text{(vii)}
\end{aligned}
$$

Leveraging Dirichlet's *exponential form* and *digamma function* (see Appendix D), we further derive $\mathcal{L}$ as follows:

$$
\begin{aligned}
&\mathcal{L}(\gamma, \zeta, \eta; \alpha, \beta) = \\
&\qquad \sum_{t=1}^{T}\sum_{w=1}^{V}(\beta-1)(\Psi(\eta_w^t) - \Psi\big(\sum_{w'=1}^{V}\eta_{w'}^t\big)) + \log\Gamma\big(V\beta\big) - V\log\Gamma(\beta) && \text{(i)}\\
&+ \sum_{d=1}^{\mathcal{D}} \Big(\sum_{t=1}^{T}(\alpha-1)(\Psi(\gamma_t^d) - \Psi\big(\sum_{t'=1}^{T}\gamma_{t'}^d\big)) + \log\Gamma\big(T\alpha\big) - T\log\Gamma(\alpha) && \text{(ii)}\\
&+ \quad \sum_{t=1}^{T}\sum_{w=1}^{V} n_d^w \zeta_{tw}^d (\Psi(\gamma_t^d) - \Psi\big(\sum_{t'=1}^{T}\gamma_{t'}^d\big)) && \text{(iii)}\\
&+ \quad \sum_{t=1}^{T}\sum_{w=1}^{V} n_w^d \zeta_{tw}^d (\Psi(\eta_w^t) - \Psi\big(\sum_{w'=1}^{V}\eta_{w'}^t\big)) \Big) && \text{(iv)}\\
&- \quad \sum_{t=1}^{T}\sum_{w=1}^{V}(\eta_w^t - 1)(\Psi(\eta_w^t) - \Psi\big(\sum_{w'=1}^{V}\eta_{w'}^t\big)) + \log\Gamma\big(\sum_{w=1}^{V}\eta_w^t\big) - \sum_{w=1}^{V}\log\Gamma(\eta_w^t) && \text{(v)}\\
&- \sum_{d=1}^{\mathcal{D}} \Big(\sum_{t=1}^{T}(\gamma_t^d - 1)(\Psi(\gamma_t^d) - \Psi\big(\sum_{t'=1}^{T}\gamma_{t'}^d\big)) + \log\Gamma\big(\sum_{t=1}^{T}\gamma_t^d\big) - \sum_{t=1}^{T}\log\Gamma(\gamma_t^d) && \text{(vi)}\\
&- \quad \sum_{t=1}^{T}\sum_{w=1}^{V} n_w^d \zeta_{tw}^d \mathbb{E}_q\left[\log(\zeta_{tw}^d)\right] \Big) && \text{(vii)}
\end{aligned}
$$

As we update of the variational parameters one document at a time, we notate the ELBO on the single document-level with ℓ,

$$\mathcal{L}(\gamma, \zeta, \eta; \alpha, \beta) = \sum_{d=1}^{\mathcal{D}} \ell(\gamma, \zeta, \eta; \alpha, \beta) \tag{3.10}$$

Further simplifying ℓ by folding digamma functions, we have

$$\begin{aligned}
\ell(\gamma, \zeta, \eta; \alpha, \beta) = &\sum_{w=1}^{V} n_w^d \sum_{t=1}^{T} \zeta_{tw}^d (\mathbb{E}_q[\log(\theta_t^d)] + \mathbb{E}_q[\log \phi_w^t] - \log(\zeta_{tw}^d)) \\
&- \log\Gamma\big(\sum_{t'=1}^{T} \gamma_{t'}^d\big) + \sum_{t=1}^{T} (\alpha - 1 - \gamma_t^d + 1)\mathbb{E}_q[\log(\theta_t^d)] + \log\Gamma(\gamma_t^d) \\
&+ \log\Gamma(T\alpha) - T\log\Gamma(\alpha) \\
&+ \sum_{t=1}^{T}\sum_{w=1}^{V} (\beta - 1 - \eta_w^t + 1)\mathbb{E}_q[\log(\phi_w^t)] - \log\Gamma\big(\sum_{w'=1}^{V} \eta_{w'}^t\big) \\
&- \log\Gamma(\eta_w^t) + \log\Gamma(V\beta) - V\log\Gamma(\beta)
\end{aligned} \tag{3.11}$$

Variational EM. Having obtained the ELBO, we now turn to the optimization. In brief terms, the idea to iterate over an EM routine[14] (Dempster et al., 1977), where

- E-step: Holding $\{\alpha, \beta\}$ fixed, for each document, find

$$\{\gamma^*, \zeta^*, \eta^*\} = \underset{\{\gamma, \zeta, \eta\}}{\text{argmax}}\ \ell(\gamma, \zeta, \eta; \alpha, \beta) \tag{3.12}$$

- M-step: For each pass on $\mathcal{D}$, holding $\{\gamma^*, \zeta^*, \eta^*\}$ fixed, find

$$\{\alpha^*, \beta^*\} = \underset{\{\alpha, \beta\}}{\text{argmax}}\ \mathcal{L}(\gamma^*, \zeta^*, \eta^*; \alpha, \beta) \tag{3.13}$$

Differentiating ℓ with respect to the variational parameters, set the results to 0, and finally solve for the parameters, we obtain the following:

$$\zeta_{tw}^d = \frac{1}{Z}\exp(\mathbb{E}_q[\log(\theta_t^d)] + \mathbb{E}_q[\log \phi_w^t]) \tag{3.14}$$

$$\text{where } Z = \sum_{w=1}^{V} \exp(\mathbb{E}_q[\log(\theta_t^d)] + \mathbb{E}_q[\log \phi_w^t])$$

$$\gamma_t^d = \alpha + \sum_{w=1}^{V} n_w^d \zeta_{tw}^d \tag{3.15}$$

$$\eta_w^t = \beta + \sum_{d=1}^{\mathcal{D}} n_w^d \zeta_{tw}^d \tag{3.16}$$

14. EM-algorithm is guaranteed to converge at a local minimum or a saddle point (McLachlan and Krishnan, 2008).

Algorithm 3 Batch Variational Bayes

Initialize (randomly) η
while relative improvement in $\mathcal{L} > \epsilon$ **do**
▷ E-step
  for $d \in \mathcal{D}$ **do**
    Initialize $\gamma^d = 1$ (arbitrary constant)
    repeat
      set $\zeta_{tw} \propto \exp(\mathbb{E}_q[\log(\theta_t^d)] + \mathbb{E}_q[\log\phi_w^t])$
      set $\gamma_t^d = \alpha + \sum_{w=1}^{V} n_w^d \zeta_{tw}^d$
    until $\frac{1}{T}\sum_{t=1}^{T} |\Delta\gamma_t| < \epsilon$
  end for
▷ M-step
  compute $\hat{\eta_w^t} = \beta + \sum_{d=1}^{\mathcal{D}} n_w^d \zeta_{tw}^d$
end while

where $\mathbb{E}_q[\log(\theta_t^d)] = \Psi(\gamma_t^d) - \Psi\big(\sum_{t'=1}^{T}\gamma_{t'}^d\big)$, $\mathbb{E}_q[\log\phi_w^t] = \Psi(\eta_w^t) - \Psi\big(\sum_{w'=1}^{V}\eta_{w'}^t\big)$, which are convenient forms for implementation. For $\{\alpha, \beta\}$

$$\alpha' = \log\frac{\alpha}{\exp\left(\frac{\partial\mathcal{L}}{\partial\alpha} / \left(\frac{\partial^2\mathcal{L}}{\partial\alpha^2}\alpha + \frac{\partial\mathcal{L}}{\partial\alpha}\right)\right)} \tag{3.17}$$

$$\beta_{tw} = \frac{1}{Z}\sum_{d=1}^{\mathcal{D}}\sum_{w=1}^{V} n_{tw}^d \zeta_{tw}^d \tag{3.18}$$

$$\text{where } Z = \sum_{t=1}^{T}\sum_{d=1}^{\mathcal{D}}\sum_{w=1}^{V} n_{tw}^d \zeta_{tw}^d$$

In practice, the update for the parameters $\{\alpha, \beta\}$ is oftentimes avoided (i.e. fixed as symmetric Dirichlet parameters) in favor of the following simplified formulation, to save computational cost[15].

- E-step: Holding η constant, for each document, find

$$\{\gamma^*, \zeta^*\} = \underset{\{\gamma,\zeta,\}}{\text{argmax}}\ \ell(\gamma, \zeta, \eta) \tag{3.19}$$

- M-step: For each pass on $\mathcal{D}$, holding $\{\gamma^*, \zeta^*\}$ fixed, find

$$\eta^* = \underset{\eta}{\text{argmax}}\ \mathcal{L}(\gamma^*, \zeta^*, \eta) \tag{3.20}$$

To ensure the stability in updates, it is also common to process documents in batches. The batched-version of the algorithm is summarized in Algorithm 3.

15. Estimating α, in particular, involves computing Hessian matrix, which incurs at least additional linear time complexity (cf. (Blei et al., 2003, Appendix A.2)). As the empirical motivation for the additional cost is not strong (i.e. usually not affect performance), it is thus often ignored all together (Hoffman et al., 2010).

Algorithm 4 Online Variational Bayes

Define $\rho_s = (\tau_0 + t)^{-\kappa}$
Initialize (randomly) η
for $s = 0$ to ∞ **do**
▷ E-step
  Initialize $\gamma^d = 1$ (arbitrary constant) for incoming document d at step s
  repeat
    set $\zeta_{tw} \propto \exp(\mathbb{E}_q[\log(\theta_t^d)] + \mathbb{E}_q[\log\phi_w^t])$
    set $\gamma_t^d = \alpha + \sum_{w=1}^{V} n_w^d \zeta_{tw}^d$
  until $\frac{1}{T}\sum_{t=1}^{T} |\Delta\gamma_t| < \epsilon$
▷ M-step
  compute $\hat{\eta_w^t} = \beta + n_w^d \zeta_{tw}^d$
  set $\eta = (1 - \rho_s)\eta + \rho_s\hat{\eta}$
end for

Online Variational Bayes for LDA. We now look at a variant[16] of VB — *Online Variational Bayes* (Online VB) (Hoffman et al., 2010), which has been shown to be particularly effectively in dealing with large quantity of data, and learns in an online manner — the model updates for each incoming new document. To handle massive number of documents, Online VB only looks at each document in a data stream one time for parameter updates, then discard the data, and thereby avoids the necessity of store data locally. In addition, Online VB has a set "flow-control" hyperparameters are introduced in the M-step to tune the rate at which the model learns:

$$\eta = (1 - \rho_s)\eta + \rho_s\hat{\eta}$$
$$\text{where } \rho_s = (\tau_0 + t)^{-\kappa}$$

where s is the global step, $\kappa \in (0.5, 1]$ controls the rate at which old values are "forgotten", and $\tau_0 \geq 0$ slows down the early steps of the algorithm. Essentially, the algorithm is a version of stochastic coordinate ascent. Interestingly, despite of the "cutting-corners", Online VB performs just as well as the "full" version (Hoffman et al., 2010). The procedure is outlined in Algorithm 4.

3.4 Algorithm Comparison

Asuncion et al. (2008) conduct a comprehensive empirical comparison of learning algorithms for TM, and reach two main conclusions: (i) With optimal tuning of smoothing hyperparameters (i.e. Dirichlet priors), the difference between CGS and VB in performance[17] diminishes; (ii) Given (i), speed becomes the main concern as far as algorithm selection. Generally speaking, while CGS has the strength of asymptotic 0 bias and sample variance (see Appendix A and B), and does better empirically than VB, it is often much slower to converge than the latter (Asuncion et al., 2008; Wang et al., 2017), due in non-trivial part to its non-parallelizability (Zhai et al., 2012).

16. As we will see soon, VB usually performs less well than CGS empirically. Its advantage lies in the speed over the latter, which is not by a large margin in its original formulation (Blei et al., 2003).
17. Performance in terms of perplexity. See Section 3.5.

Teh et al. (2007) present a *collapsed Variation Bayes* (CVB) algorithm which has been shown, in a detailed algorithmic comparison with "base" CGS and VB discussed here, to perform as well as CGS while speedwise as fast as VB (Asuncion et al., 2008). In fact, boiling down to the algorithmic details, some variants of CVB compute exactly the same probability as CGS, but in addition deterministic and parallelizable (explains its speed).

VB-based algorithms, however, involve also non-trivial amount of additional implementational complexity. This is especially true if one intends to extend the TM to downstream tasks[18]. For instance, if the modeling objective is how to learn effectively from small amounts of data (Wang et al., 2017), then CGS provides a more convenient option.

Crucially, note that perplexity-based evaluation does not provide direct evidence that one algorithm option is better than the other even in intrinsic evaluation (Chang et al., 2009) in terms of alignment with human intuition (see Section 3.5). Similar pattern is also observed in extrinsic evaluation (Wang et al., 2017). In practice, model selection highly depends on empirical results in target tasks.

In general, in any given project involving TM, the following are the rules-of-thumb to follow in my humble opinion:

- If the objective of the project lies in some downstream task, always start with a variant of MCMC method (e.g. CGS). If the results suffices to show the main insight intended for the project, one may always explore VB-based algorithms later for software "weaponization".
- If in the project one must process a large quantity of documents (rule-of-thumb: more than 1 billion tokens), skip MCMC and opt directly for a base VB for prototyping, and explore more sophisticated variants of it later.

Note that the suggestions above assume the absence of an off-the-shelf implementation (e.g. `Gensim`, Řehůřek and Sojka (2010)) for the purpose of the project.

3.5 Evaluation Methods

How a topic model is evaluation often depends on its application. In general, the evaluation methods are classified along two dimensions:

- *Intrinsic* vs. *Extrinsic*: Whether the model is useful in and of itself, or is it useful for tackling another problem?
- *Automatic* vs. *Human-based*: Whether the model is evaluated in unsupervised, or supervised fashion?

For a better intuition, consider some examples[19]:

- Intrinsic + Automatic: Mimno et al. (2011) leverage cooccurrence information in the corpus to design a *Pointwise Mutual Information* (PMI) inspired metric called *topic coherence* for evaluating the quality of topic assignments.

18. A nice quote from David Blei encapsulate nicely this dilemma we face in algorithm selection: "Variational inference is that thing you implement while waiting for your Gibbs sampler to converge." (quote cited by Jason Eisner).
19. Note that the examples do not include all the other ideas and techniques proposed in cited works.

- Intrinsic + Human-based: Chang et al. (2009) formulate a novel *word/topic intrusion* method to evaluate topic models with human judgment.

- Extrinsic + Automatic: Wei and Croft (2006) apply topic models to informational retrieval tasks and evaluate the models based on the improvement the retrieval tasks.

- Extrinsic + Human-based: Wang et al. (2017) present a bimodal extension (i.e. words and word properties) of topic models in a word meaning task, and evaluate the model with correlation scores between model-predicted and human-judged properties for unknown words.

To stay focused on topic models per se, I elaborate only on intrinsic evaluation methods, and refer interested readers to the listed references here on extrince methods.

Intrinsic automatic methods. The most straightforward evaluation metric for a TM is the predictive probability for a held-out document set (Wallach et al., 2009), which is often used to further compute a *perplexity* (Foulds and Smyth, 2014), i.e. $P(\mathcal{D}_{test}|\mathcal{D}_{train})^{-1/N_{test}}$.

$$P(\mathcal{D}_{test}|\mathcal{D}_{train}) = \int p(\mathcal{D}_{test}|\omega)p(\omega|\mathcal{D}_{train})d\omega \tag{3.21}$$

where ω is a shorthand for the relevant parameters of the topic model. Despite its clarity and interpretability, the method has been shown to not always consistent with human judgment on the quality of topic assignment (Chang et al., 2009). In addition, such method gets mathematically very complex in its various extensions (Wallach et al., 2009).

As an alternative, Mimno et al. (2011) propose a much simpler metric (i.e. topic coherence), and empirically tested its effectiveness by a human judgment based validation.

$$C(t, V^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(v_m^{(t)}, v_l^{(t)} + 1)}{D(v_l^{(t)})} \tag{3.22}$$

where $V^{(t)} = (v_1^{(t)}, \ldots, v_M^{(t)})$ is a list of the M most probable words in topic t, $D(v)$ is the *document frequency* of word type v (i.e. the number of documents with at least 1 token of type v), and $D(v, v')$ is *co-document frequency* of word types v and v' (i.e. the number of documents containing 1 or more tokens of type v and at least 1 token of type v'). Stated informally, if two words under topic t cooccur frequently, and they are not the words which are prevalent in a large number of documents in a corpus (i.e. stop words), then they contribute more to the coherence score.

Intrinsic human-based methods. Chang et al. (2009) define two human evaluation tasks. After having trained a topic model and collected related statistics, human participants are presented with the following tasks:

- *Word intrusion*: For a given topic t, the participant is presented with 6 randomly ordered words, among which 5 are from the most probable words in t, and 1 is a randomly sampled out-of-place "intruder". The task is to identify the intruder that do not belong with the rest of the words.

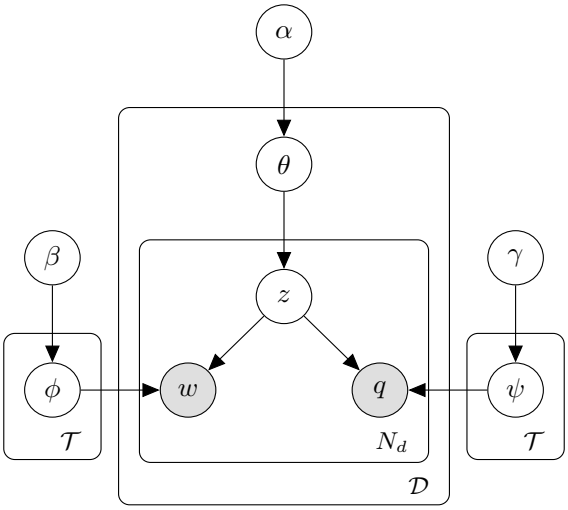

Figure 3: Bimodal Topic Model (bi-TM)

- *Topic intrusion*: For a given document d, the participant observes 6 randomly ordered topics, in which 5 are top-probable topics predicted by the model, and 1 out-of-place "intruder". The task is the same as above.

It is worth mentioning that, besides validating (Chang et al., 2009)'s finding of the inefficacy of held-out predictive probability based metrics (Wallach et al., 2009; Foulds and Smyth, 2014), Mimno et al. (2011) in addition demonstrate their topic coherence score correlates very well with human judgment.

4. Extensions

This section explores a selected set of extensions to the basic TMs described in Section 3. Section 4.1 features a "multichannel" version of TM, where the model takes input of multiple types (e.g. textual and visual) and executes inference in parallel. Section 4.2 introduces TMs with supervision (each document is pair with a "correct answer"). Section 4.3 looks at dynamic TMs, which model the change in the "word composition" of topics in a Gaussian time series. Finally, I close the section with a *Neural Network* based topic model (section 4.4).

4.1 Multimodal Topic Models

Andrews et al. (2009) present a *bimodal* generalized topic model (bi-TM), which takes *distributional data* (i.e. texts) and *experiential data* (i.e. textually coded visual, tactile, etc. sensory information) in parallel, and apply the model to various tasks, including semantic inference and word similarity. The model shows strong performance (in terms of correlation with human judgment) on these tasks, providing initial evidence for the viability of the multimodal formulation. Roller and Schulte im Walde (2013) further extend bi-TM in a *multimodal Latent Dirichlet Allocation* (mLDA) in similar tasks. mLDA includes in addition

bi-TM multi-shot	`clothing`, `made_of_material`, `has_sleeves`, `different_colours`, `worn_by_women`
bi-TM one-shot	`clothing`, `is_long`, `made_of_material`, `different_colours`, `has_sleeves`

Table 1: Top 5 properties of *gown* predicted by bi-TM. Top entries: multi-shot. Bottom entry: one-shot, context *undo-dobj*.

low-level image features (which are extracted automatically) through various techniques, showing excellent correlation with crowdsourced human judgment.

More recently, Wang et al. (2017) adopt the bimodal formulation to model the fast-mapping in humans' learning of unknown concepts (Carey and Bartlett, 1978), demonstrating the model exploits statistical patterns (i.e. cooccurrence of context terms[20]) in parallel channels efficiently. In the following I briefly describe this variant of multimodal TM, the architecture of which is illustrated with Figure 3.

The main task for the bi-TM is to learn lexical semantics of concepts[21] from their distributional contexts (in some corpora), then perform inference on an unknown concept. For instance, in the following sentence, the task of the bi-TM is to infer the meaning of the unknown concept *wampimuk*, for which a human would likely to give reasonable guesses (e.g. a small animal of some sort) after observing the single instance:

> We found a cute, hairy *wampimuk* sleeping behind the tree[22].

The model learns from two corpora: A distributional corpus (i.e. texts) and a *feature norm* (McRae et al., 2005) corpus. Feature norms are definitional properties collected from human participants (e.g. the concept `cat` has property `an_animal, feline, has_4_legs`, etc.). For each training concept, a 2-channeled *pseudo-document* is constructed to represent its lexical semantics — A document that contains pairs of a context item $w \in V$ (in distributional data) and a property $q \in Q$ (in feature norm data), meaning that w has been observed to occur with an instance of c that had q. The generative story is as follows. For each known concept c, draw a multinomial θ over topics. For each topic t, draw a multinomial ϕ over context items $w \in V$, and a multinomial ψ over properties $q \in Q$. To generate an entry for c's pseudo-document, draw a topic $t \sim Mult(\theta)$. Then, from t, simultaneously draw a context item from ϕ and a property from ψ. The model uses a bimodal extension of CGS for inference[23].

The model produces impressive[24] property predictions for unknown concepts, in terms of correlation with human judgment. Table 1 illustrates top properties predicted for the concept `gown`. In *multi-shot* condition, the model gets to observe all the contexts of `gown` in

20. A context term is a *predicate-role*. For instance, `feed-dobj` is a context term for `cat`, where the latter is an argument filler for the verbal predicate (i.e. `feed`) in the former.
21. This can be taken as a subset word types, specifically concrete nouns (e.g. `cat, house, cup`).
22. Example from Lazaridou et al. (2014)
23. VB runs several factors faster than CGS, but produces poor performance. This provides a good example where equivalent performance in perplexity does not translate into that in downstream tasks.
24. The quality of the results is not excellent in absolute terms, but impressive considering the meager amount of data it learns from — McRae et al. (2005) collect definitional properties for only 541 nouns.

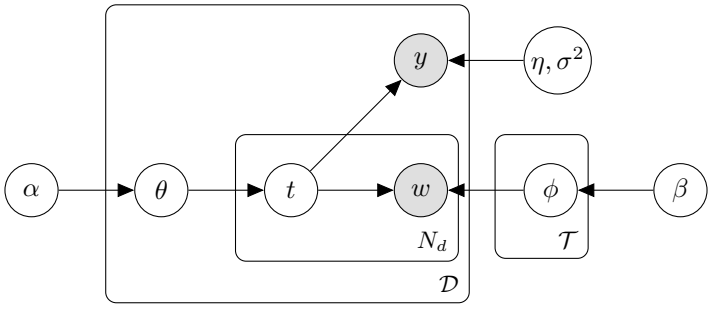

Figure 4: Supervised Topic Model

the distributional corpus before making prediction, whereas in *one-shot* condition[25], only one context is observed.

4.2 Supervised Topic Models

While prototypically presented as an unsupervised dimension-reduction model, TM can easily be extended to include supervision (sLDA). Blei and McAuliffe (2007) present a simple formulation (architecture illustrated in Figure 4) where a Gaussian linear regression component is incorporated to model per-document responses, mirroring the PCA-PLS[26] unsupervised-supervised pair in dimension-reduction techniques.

The generative story for sLDA is briefly described as follows:

1. For each document $d \in \mathcal{D}$, draw a distribution θ^d over topics: $\theta \sim Dir(\alpha)$
2. For the ith word in d,
   (a) Draw a topic t_i: $t \sim Mult(\theta^d)$
   (b) Draw a word w_i: $w_i \sim Mult(\phi^{t_i})$, where $\phi \sim Dir(\beta)$.
3. For the response of d, draw $y \sim \mathcal{N}(\eta^T \bar{t}, \sigma^2)$, where $\bar{t} = \frac{1}{N_d} \sum_{i=1}^{N_d} t_i$.

The inference algorithm is a modified VB, where the updates for the variational parameters take into account the parameters of the regressor[27]. Replacing the Gaussian linearity with a *Generalized Linear Model* (GLM), sLDA is also shown to be compatible with the discrete/categorical response type. It is particularly worth noting that, while having been shown to capture topical (i.e. genre) differentiators well in document clustering, TMs are flexible enough to be "re-oriented" to focus on non-topical information (e.g. `excellent` vs. `terrible` in movie rating prediction, which is independent of topics).

Blei and McAuliffe (2007) experiment on movie rating and web-page popularity prediction tasks and show sLDA's advantage over state-of-the-art regularized linear regression models. More recently, improving on sLDA, Zhu et al. (2012) integrate max-margin methods

25. Not all contexts are equally informative. For instance, for `gown`, the context `undo-dobj` "gives away" the concept's properties better than `like-dobj` does.
26. *Principal Component Analysis* (PCA); *Partial Least Square* regression (PLS).
27. As it is difficult to describe the modification in brief terms, I refer the reader to the original work.

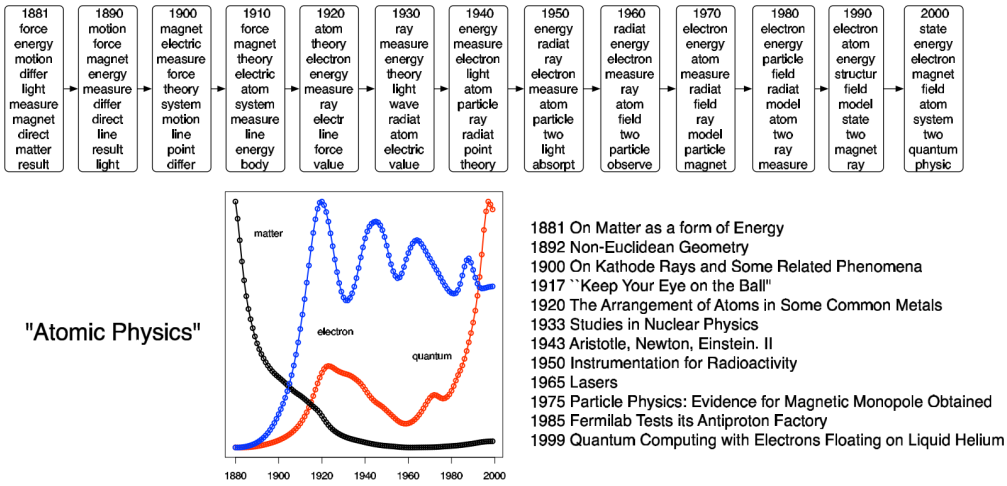

Figure 5: Change in Topic Composition for `Atomic Physics`

in a *Maximum Entropy Discrimination LDA* (MedLDA), and demonstrate the potential of supervised TMs in classification tasks. Rubin et al. (2012) show that supervised TMs perform competitively against conventional discriminative classification models such as *Support Vector Machines* (SVM), and maintain strong performance even with large number of class labels where SVM usually fails. It must be noted that, as far as my knowledge, a comprehensive comparison between neural network based discriminative classifiers (which usually hold state-of-the-art titles in the tasks (Lai et al., 2015; Shrestha et al., 2017)) and supervised TMs is still lacking for a good empirical basis to promote the latter to a top contender in the area.

4.3 Dynamic Topic Models

Blei and Lafferty (2006) propose a *Dynamic Topic Model* (DTM) architecture which treats a collection of documents as a time series, motivated by the time-sensitivity in the word composition of topics over time. For instance, the topic `atomic physics`, as we understand it in 2000, differs substantially from it is in late 19th century (Figure 5)[28]. DTMs attempts to model observations of this type properly. One potential application of DTM could be popular perception of brands over time, for instance. *Microsoft* and *Google*, for instance, may be associated with non-trivially different sentiments now as they were in the early 2000. Understanding the underlying cause for the shift might shed light on the dynamics behind public opinions in relation to managerial strategies of corporates[29].

In DTM, $\{\alpha, \beta\}$, i.e. the parameters governing the distribution of topics and words (given topics), are modeled as log Gaussian[30] state space distributions, where, at time s, $\{\alpha_t, \beta_t\}$

28. Source: Blei and Lafferty (2006), Figure 4.
29. For more applications of DTM in various areas, see Hall et al. (2008); Lee et al. (2016).
30. log Gaussian ensures $\alpha_s, \beta_s > 0, \forall s$.

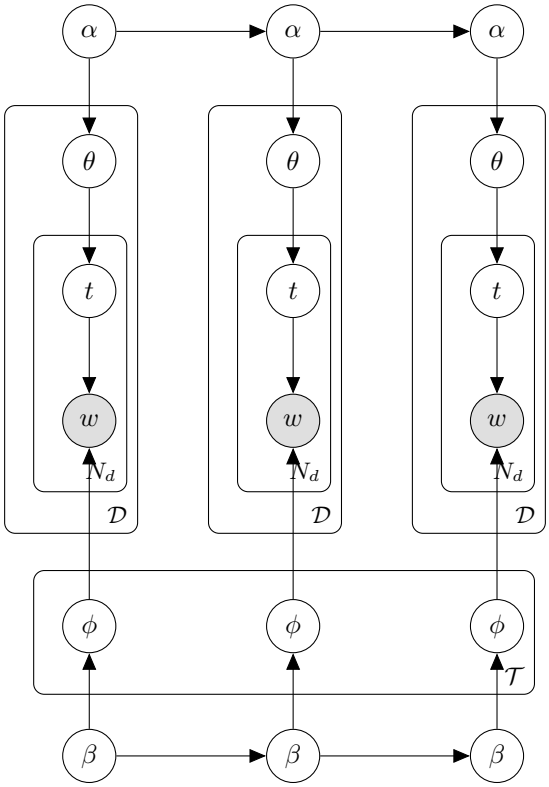

Figure 6: Supervised Topic Model

depends on their respective previous state as follows:

$$\alpha_s \sim \log\mathcal{N}(\alpha_{s-1}, \delta^2 I) \tag{4.1}$$

$$\beta_s \sim \log\mathcal{N}(\beta_{s-1}, \sigma^2 I) \tag{4.2}$$

The generative story for DTM summarizes[31] as follows:

1. Initialize symmetric Diriclet priors $\alpha, \beta = C$ (arbitrary constant[32].)
2. For each subcorpus $\mathcal{D}_s$ (i.e. at time slice s), draw
   (a) $\alpha_s \sim \log\mathcal{N}(\alpha_{s-1}, \delta^2 I)$
   (b) $\beta_s \sim \log\mathcal{N}(\beta_{s-1}, \sigma^2 I)$
3. For each document $d \in \mathcal{D}$,
   (a) Draw $\eta \sim \log\mathcal{N}(\alpha_s, a^2 I)$
   (b) Get a distribution θ^d over topics, where $\theta = \pi(\eta)$

31. The outline expands the description in the original work (Blei and Lafferty, 2006, Section 2) for implementational convenience.
32. Note this is in fact a tunable hyperparameter as per specific modeling purposes (cf. footnote 10)

Class 1(alt.atheism)			Class 16(soc.religion.christian)			Class 20(talk.religion.misc)		
LDA(4)	DocNADE(31)	NTM(47)	LDA(66)	DocNADE(27)	NTM(59)	LDA(5)	DocNADE(8)	NTM(72)
belief	moral	atheism	jesus	jesus	god	lds	church	lds church
god	atheism	creation	god	god	jesus	istanbul	catholic	pope
truth	keith	human being	christ	sin	catholic faith	georgia	orthodox	protestant
reason	religion	true nature	s	lord	christ	ermeni	holy	nestorius
atheist	system	science	lord	heaven	saint	york	tradition	religious belief
christian	islam	sacred scripture	sin	church	bible	church	doctrine	catholic church
bible	belief	theism	bible	life	homosexuality	ankara	movement	religious fanaticism
religion	muslim	proof	heaven	hell	heaven	##	spirit	theology

Figure 7: Top Items in Sample Topics from Models

4. For the ith word in d,
   (a) Draw a topic t_i: $t \sim Mult(\theta^d)$
   (b) Draw a word w_i: $w_i \sim Mult(\phi^{t_i})$, where $\phi = \pi(\psi)$, $\psi \sim Dir(\beta_s)$.

where π maps samples from log Gaussian to multinomial distributions:

$$\pi(x) = \frac{\exp(x)}{\sum_{w=1}^{V} \exp(x)} \tag{4.3}$$

The plate diagram for the process is illustrated in Figure 6. In addition, comparing the DTM with static TM, Blei and Lafferty (2006) demonstrate experimentally that the former indeed captures the Markov-chain style composition change much better (by perplexity).

4.4 Neural Topic Models

Wan et al. (2015) propose an integration of neural networks and TMs in a hybrid *NN-LDA* which fuses together a *Deep Belief Network* (DBN) (Hinton et al., 2006) and an sLDA (Blei and McAuliffe, 2007). The NN-LDA extracts "visual words" from images, in the form of SIFT features (Lowe, 2004), for pre-training in its DBN component, then feed the output to the sLDA component for an image/scene classification task. The model outperform both stand-alone sLDA and neural network classifiers, together with a set of SVM-based discriminative classifiers.

A more interesting recent study has been Cao et al. (2015), who abandon the "traditional" LDA framework entirely, and present a neural network simulation of TM — *Neural Topic Model* (NTM), where the distributions over topics and words in LDA are replaced with layers of neurons. They motivate the overhaul by LDA's weakness in capturing n-gram statistics in texts (due to sparsity), which are sometimes shown to be more accurate characterization for topics (as a distribution over vocabulary items). N-grams, while ineffective in LDA due to sparsity, can be learned as embeddings with NN through `Word2Vec` (Mikolov et al., 2013) (or similar) routines by tapping into large quantity of open domain free texts. Where n-gram representations are lacking, one may take instead compositions from unigram representations through pairwise addition (Huang et al., 2012). Taking advantage of n-gram statistics, NTM produces superb topical clusters. Figure 7 (Table 2 in Cao et al. (2015)) compares sample outputs from LDA, DocNADE[33] and NTM.

33. A well-tuned neural autoregressive topic model (Larochelle and Lauly, 2012).

Further, NTM does away with complex inference algorithms which are known to cause difficulty in implementation for mathematically less sophisticated machine learning researchers (cf. Section 3.4, and footnote 18) — the architecture can be easily put together, leveraging convenient off-the-shelf automatic differentiation programs[34]. More crucially, preliminary experiments of Cao et al. (2015) show that it outperforms various strong TM competitors, including sLDA and DocNADE, in a multiclass classification task, showing strong promise of the model.

I now describe NTM in more detail. Treating TM from the perspective of NN, Cao et al. (2015) observes that the conditional probability

$$p(w|d) = \sum_{t=1}^{T} p(w|t_i)p(t_i|d) = \sum_{t=1}^{T} \phi_w^t \theta_t^d \tag{4.4}$$

can be modeled with a conventional matrix operation

$$\begin{aligned} p(w|d) &= (\phi(w))^\intercal \cdot \theta(d) \\ &\text{where } \phi(w) = \langle p(w|t_1), \ldots, p(w|t_T)\rangle^\intercal, \theta(d) = \langle p(t_1|d), \ldots, p(t_T|d)\rangle^\intercal \end{aligned} \tag{4.5}$$

The core components in LDA, the word-topic and topic-document joint distributions, therefore, are encapsulated in $\phi(w), w \in V$ and $\theta(d), d \in \mathcal{D}$. As the first step, the NTM computes a matching score for each pair of n-gram and document. This computation proceeds in the following steps:

1. Embedding lookup for n-grams. Let $g = \{w_1, ..., w_n\}, w \in \mathbb{R}^{V\times 1}, \forall n$ be an n-gram entry if g is not in the vocabulary of pre-trained embeddings $\mathcal{V}$, otherwise $g \in \mathbb{R}^{V\times 1}$. All inputs are one-hot coded. Let $W_e \in \mathbb{R}^{V\times h}$ be the embedding lookup matrix.

$$f_e(g) = \begin{cases} W_e^\intercal \cdot g & \text{if } g \in \mathcal{V} \\ \sum_{w\in g} W_e^\intercal \cdot w & \text{otherwise} \end{cases} \tag{4.6}$$

   where h is the dimension of embeddings. W_e is not trainable (i.e. held constant)[35].

2. Map n-gram embeddings to distributions over topics. Let $W_g \in \mathbb{R}^{h\times T}$ be a weight matrix, $b_g \in \mathbb{R}^{T\times 1}$ the bias.

$$f_g(f_e) = \sigma(W_g^\intercal \cdot f_e + b_g) \tag{4.7}$$

   For each n-gram g, this produces a vector in $\mathbb{R}^{T\times 1}$.

3. Embed documents. Let $d \in \mathbb{R}^D$ be a one-hot vector, $W_d \in \mathbb{R}^{D\times T}$ the document embedding matrix, $b_d \in \mathbb{R}^{T\times 1}$ the bias.

$$f_d(d) = \text{softmax}(W_d^\intercal \cdot d + b_d) \tag{4.8}$$

   For each document d, this produces a vector in $\mathbb{R}^{T\times 1}$.

34. e.g. `Tensorflow, Theano, Caffe`, etc.

35. Cao et al. (2015) argues usually the target corpora in document modeling tasks are substantially smaller than the corpora on which pre-trained embeddings are obtained (e.g. 100 billion words for `Word2Vec` embeddings), therefore it is safer to held the embeddings constant.

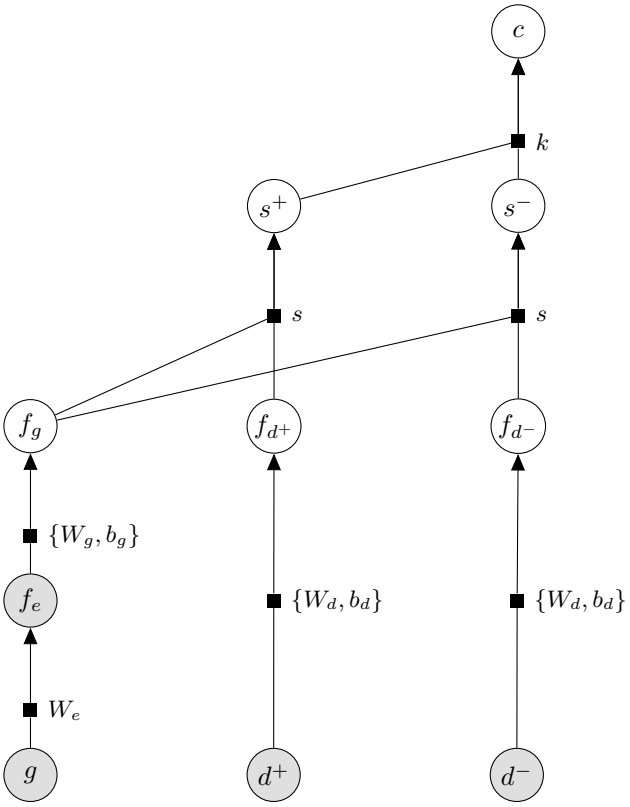

Figure 8: Neural Topic Model (Unsupervised)

4. Compute matching scores for n-gram and document pairs (g, d).

$$s(g, d) = f_g(g)^\intercal \cdot f_d(d) \tag{4.9}$$

The score $s(g, d)$ is a scalar.

Next, a *pairwise ranking* (Collobert et al., 2011) training routine[36] is applied to pairs (g, d^+), where g appears in d^+. For g in each pair, a *negative document* d^- is randomly sampled from documents where g does not appear. From this we obtain pair (g, d^-). A hinge-loss based cost function for (g, d^+, g^-) is formulated as follows:

$$c(g, d^+, d^-) = \max(0, k - s(g, d^+) + s(g, d^-)) \tag{4.10}$$

where k is the margin constant. It is clear the cost will be low if the score for the "positive pair" $s(g, d^+)$ is higher than $s(g, d^-) + k$, then the cost is 0, otherwise it is non-zero. The entire architecture of NTM is outlined in Figure 8. Because the n-gram/word embedding W_e is held constant, we only update W_d and W_g. Specifically, for each (g, d^+) and (g, d^-) pair

$$W' = W - \alpha \cdot \frac{\partial c}{\partial W} + \lambda \cdot W^\intercal W \tag{4.11}$$

36. This reminds us of the *negative sampling* training in `Word2Vec` (Mikolov et al., 2013), which follows similar idea.

where α, λ are learning rate and regularization constants.

The unsupervised NTM easily extends to a supervised version (sNTM) by attaching a classification function `Clf` to f_d, which can be treated as a vectorial representation of a document (see Cao et al. (2015) for more detail).

$$\texttt{Clf}(d) = f(W^\intercal f_d + b) \tag{4.12}$$

where W, b are the parameters of the classifier.

5. Conclusion

In this introduction, I started with a general sketch of the core ideas of TMs, and a broad overview of their applications, before diving into a complete description of the architecture of TMs in its original formulation. I then compared the two common training algorithms (i.e. CGS and VB) with suggestions for algorithm selection, and accounted for various evaluation methods. Finally I sketched in some detail four interesting and promising extension of TMs: Multimodal TMs, supervised TMs, dynamic TMs and neural TMs. The tutorial guide does not do justice to the plethora of new architectures built in the framework of topic models since its nascence in the early 2000s, however I sincerely hope it could serve as a nice entry point for fellow machine learning researchers in their further exploration.

Acknowledgments

This research was supported by the NSF grant IIS 1523637 and by the DARPA DEFT program under AFRL grant FA8750-13-2-0026. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the view of DARPA, DoD, or the US government. Most sincerely appreciate to the helpful comments from Dr. Stephen Roller and my advisor Dr. Katrin Erk in the mathematical derivations of learning algorithms described. Thanks also to my supervisor at OJO Labs, Inc. Dr. Joshua Levy, and the conference organizer of NLP Day Mr. Lynn Bender for their encouragement.

Appendix

A. Monte Carlo Integration

For any distributional function f, *Monte Carlo Integration* shows that the empirical distribution $\hat{f}$ built from samples is a discrete approximation to the true distribution (i.e. f), and the approximation becomes exact in the limit of sample size (i.e. $\hat{f} \to f$, as $N \to \infty$).

Theorem 1. Monte Carlo Integration. *Let f be an unknown distribution from which we can draw samples. Let x be a continuous variable of f. Then the empirical distribution $\hat{f}$, which is formed by M samples from f, is an unbiased estimator for f (i.e. $\hat{f} \to f$, as $M \to \infty$).*

$$\hat{f}(x) = \frac{1}{M} \sum_{i=1}^{M} f(x_i) \tag{Ap.1}$$

where $x_1, \ldots, x_M$ are assumed to be i.i.d. (i.e. independent and identically distributed).

Proof. First I show $\mathbb{E}[\hat{f}] = f$:

$$\mathbb{E}[\hat{f}(x)] = \mathbb{E}\left[\frac{1}{M} \sum_{i=1}^{M} f(x_i)\right] = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[f(x_i)] = f(x) \tag{Ap.2}$$

I then show $Var[\hat{f}(x)] \to 0$ as $M \to \infty$:

$$Var[\hat{f}(x)] = Var\left[\frac{1}{M} \sum_{i=1}^{M} f(x_i)\right] = \frac{1}{M^2} \sum_{i=1}^{M} Var[f(x_i)] \leq \frac{1}{M^2} \mathbb{E}[f(x)^2] \tag{Ap.3}$$

As $f(x)^2 \leq 1$, $\mathbb{E}[f(x)^2] \leq 1$. Therefore by *Comparison Test* (Stewart, 2012, Section 11.4), $Var[\hat{f}(x)] \to 0$ as $M \to \infty$. □

B. Ergodic Theorem

In the random walk in Gibbs sampling, samples are not independent of each other (i.e. specifically, they are a *Markov Chain* in this case). To qualify for using Monte Carlo Integration, however, as we have seen they need to be i.i.d.. We leverage the *Ergodic Theorem*[37] to construct a *stationary process* such that the samples in random walk can be taken as i.i.d. (i.e. a *pseudo-i.i.d* sequence).

Definition 1. Markov Chain. A sequence $x_1, ..., x_M$ of random variates (of a conditional distribution f) is called *Markov*, if for any m,

$$f(x_m | x_{m-1}, \ldots, x_1) = f(x_m | x_{m-1}) \tag{Ap.4}$$

i.e., if f of x_m assuming $x_{m-1}, \ldots, x_1$ equals f of x_m assuming only the immediately previous variate x_{m-1} (Papoulis, 1984).

37. The proof of Theorem 2., however, is beyond the scope of this tutorial. I refer interested readers to the following lecture notes:
`http://www.math.uchicago.edu/~may/VIGRE/VIGRE2007/REUPapers/FINALFULL/Casarotto.pdf`

Theorem 2. Ergodic Theorem for Markov Chains. *Let $\boldsymbol{x} = x_1, ..., x_M$ be an irreducible, time-homogeneous, and discrete Markov Chain. Let f be an aperiodic distributional function. If $\boldsymbol{x}$ is drawn from a density ξ, and ξ is stationary, i.e.*

$$f(x_{m+1}) = \int \xi(x_{m+1}|x_m) f(x_m) dx_m \tag{Ap.5}$$

Then,

$$\frac{1}{M} \sum_{i=1}^{M} f(x_i) \to \mathbb{E}[f(x)], \quad \text{as } M \to \infty \tag{Ap.6}$$

Given above, the following Monte Carlo approximation is therefore still valid:

$$\hat{f}(x) = \frac{1}{M} \sum_{i=1}^{M} f(x_i) \tag{Ap.7}$$

Without digressing further afield, I now explain the new concepts introduced in Theorem 2 in informal terms. (i) *irreducibility*: the probability of transitioning from any state to any other state is greater than 0 (i.e. the random walk can reach any point in the true distribution at a probability); (ii) *time-homogeneity*: the transition on the stationary density is not dependent on time (the transitional process is not related to any temporal sequence); (iii) *aperiodicity*: the pattern of transition is not following any periodic pattern (e.g. if I generate natural numbers by the rule of even numbers, I will never reach odd numbers) Essentially, the conditions guarantee that the random walk approximates the true distribution asymptotically.

C. Jensen's Inequality and ELBO

Before introducing Jensen's Inequality, we need to define *convexity/concavity.*

Definition 2. Convexity and Concavity. Let $f(x)$ be a real valued function defined on the interval $I = [a, b]$. f is said to be *convex*, if $\forall x_1, x_2 \in I$, and $0 \leq \lambda 1$,

$$f(\lambda x_1 + (1 - \lambda) x_2) \geq \lambda f(x_1) + (1 - \lambda) f(x_2) \tag{Ap.8}$$

A function is said to be *strictly convex* if the inequality is strict for $x_1 \neq x_2$. Further, f is *concave*, if $-f$ is convex.

Now we are ready for the theorem:

Theorem 3. Jensen's Inequality. *Let $f(x)$ be a convex function defined on an interval I. Then, if $x_1, \ldots, x_M \in I$, and $\lambda_1, \ldots, \lambda_M \geq 0$ with $\sum_{i=1}^{M} \lambda_i = 1$, then*

$$f\left(\sum_{i=1}^{M} \lambda_i x_i\right) \leq \sum_{i=1}^{M} \lambda_i f(x_i) \tag{Ap.9}$$

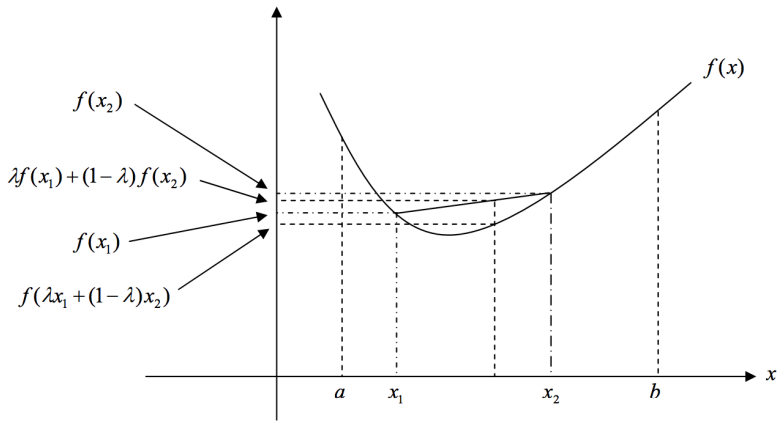

Figure 9: Jensen's Inequality in 2D

Proof. We prove by induction. For $M = 1$, the theorem is trivially true. The case of $M = 2$ is true by the the definition of convexity. Suppose the theorem is true for $M = k - 1$, and further let[38] $\lambda_i' = \frac{\lambda_i}{1-\lambda_k}$ for $i = 1, \ldots, k-1$, then

$$\begin{aligned}
\sum_{i=1}^{k} \lambda_i f(x_i) &= \sum_{i=1}^{k-1} \lambda_i f(x_i) + \lambda_k f(x_k) \\
&= (1-\lambda_k) \sum_{i=1}^{k-1} \lambda_i' f(x_i) + \lambda_k f(x_k) \\
&\geq (1-\lambda_k) f\left(\sum_{i=1}^{k-1} \lambda_i' x_i \right) + \lambda_k f(x_k) && \text{by inductive hypothesis} \\
&\geq f\left((1-\lambda_k) \sum_{i=1}^{k-1} \lambda_i' x_i + \lambda_k x_k \right) && \text{by M=2 case} \\
&= f\left(\sum_{i=1}^{k-1} \lambda_i x_i + \lambda_k x_k \right) = f\left(\sum_{i=1}^{k} \lambda_i x_i \right) && \text{(Ap.10)}
\end{aligned}$$

□

Figure 9 gives a good visualization of convexity and Jensen's Inequality in 2D space.

To validate the application of Jensen's Inequality in ELBO (i.e. Eq. 3.9), we introduce one more theorem (but skip the proof) to show that the function involved therein (i.e. logarithm) is concave.

Theorem 4. Test for Convexity/Concavity. *If $\frac{d^2}{dx^2} f(x)$ exists on $I = [a, b]$, and $\frac{d^2}{dx^2} f(x) \geq 0$ on the interval. Then, $f(x)$ is convex on I. $f(x)$ is concave if $\frac{d^2}{dx^2} f(x) \leq 0$ (other conditions being equal).*

38. So that we have $\sum_{i=1}^{k-1} \lambda_i' = 1$ in using the inductive hypothesis.

Since $\frac{d^2}{dx^2}\log(x) = -\frac{1}{x^2} \leq 0$, the function is concave, verifying Eq. 3.9. For more details on the subject, cf. Dempster et al. (1977).

D. Exponential Family

In the expansion of the ELBO formula (Section 3.3), we observe many terms involving Dirichlet distribution, which is a member of a broader class of distribution — *Exponential Family*. As we will see, it is crucial that we take advantage of Dirichlet's membership as an exponential in simplifying the ELBO for operationalizing a learning algorithm for VB. We begin by a definition of the exponential family[39]:

Definition 3. Exponential Family. A probability distribution $p(\boldsymbol{x}|\rho)$ for $\boldsymbol{x} = \{x_1, ..., x_M\}$ and $\boldsymbol{\rho} \in \mathbb{R}^k$ is said to be in the *exponential family*, if it is of the form

$$p(\boldsymbol{x}|\boldsymbol{\rho}) = h(\boldsymbol{x})\exp[\eta(\boldsymbol{\rho})^T\phi(\boldsymbol{x}) - A(\eta(\boldsymbol{\rho}))] \tag{Ap.11}$$

where η is a function that maps parameters $\boldsymbol{\rho}$ to the *natural parameters* $\boldsymbol{\eta} = \eta(\boldsymbol{\rho})$, $\phi(\boldsymbol{x})$ is the *sufficient statistic*, A is the *log partition function*, and $h(\boldsymbol{x})$ is the *scaling constant*.

If $p(\boldsymbol{x}|\boldsymbol{\rho})$ is a Dirichlet, it writes in exponential form as:

$$p(\boldsymbol{x}|\boldsymbol{\rho}) = \exp\Big(\big(\sum_{i=1}^{k}(\rho_i - 1)\log\, x_i\big) + \log\,\Gamma\big(\sum_{i=1}^{k}\rho_i\big) - \sum_{i=1}^{k}\log\,\Gamma(\rho_i)\Big) \tag{Ap.12}$$

where the natural parameters $\boldsymbol{\eta} = \eta(\boldsymbol{\rho}) = \boldsymbol{\rho} - 1$, the sufficient statistic $\phi(\boldsymbol{x}) = \log(\boldsymbol{x})$, the log partition function $A = \sum_{i=1}^{k}\log\,\Gamma(\rho_i) - \log\,\Gamma(\sum_{i=1}^{k}\rho_i)$, and the scaling constant is $h(\boldsymbol{x}) = 1$. In particular note that, the ith part of A is $A_i = \log\,\Gamma(\rho_i) - \log\,\Gamma(\sum_{j=1}^{k}\rho_j)$.

In the simplification of the ELBO, we observe the frequent appearance of the expectation of the form $\mathbb{E}[\log(x)|\rho]$. We now show it can be reduced to a function of the Dirichlet parameter ρ. For this we need the following important result:

Theorem 5. Property of Log Partition Function. *Let distribution $p(\boldsymbol{x}|\boldsymbol{\rho})$ be in the exponential family. Then, the first derivative of its log partition function A with respect to the natural parameter is the expectation of its sufficient statistic. That is,*

$$\frac{dA}{d\boldsymbol{\rho}} = \mathbb{E}_p[\phi(\boldsymbol{x})] \tag{Ap.13}$$

39. Adapted from Murphy (2012), Section 9.2.

Proof. $A = \log \int h(\boldsymbol{x})\exp(\eta(\boldsymbol{\rho})^T\phi(\boldsymbol{x}))d\boldsymbol{x}$, therefore

$$\begin{aligned}
\frac{dA}{d\boldsymbol{\rho}} &= \frac{d}{d\boldsymbol{\rho}}\left(\log \int h(\boldsymbol{x})\exp(\eta(\boldsymbol{\rho})^T\phi(\boldsymbol{x}))d\boldsymbol{x}\right) \\
&= \frac{\frac{d}{d\boldsymbol{\rho}}\int h(\boldsymbol{x})\exp(\eta(\boldsymbol{\rho})^T\phi(\boldsymbol{x}))d\boldsymbol{x}}{\int h(\boldsymbol{x})\exp(\eta(\boldsymbol{\rho})^T\phi(\boldsymbol{x}))d\boldsymbol{x}} \\
&= \frac{\int h(\boldsymbol{x})\exp(\eta(\boldsymbol{\rho})^T\phi(\boldsymbol{x}))\phi(\boldsymbol{x})d\boldsymbol{x}}{\exp(A)} \\
&= \int \underbrace{h(\boldsymbol{x})\exp(\eta(\boldsymbol{\rho})^T\phi(\boldsymbol{x}) - A)}_{\text{exponential form}}\phi(\boldsymbol{x})d\boldsymbol{x} \\
&= \int p(\boldsymbol{x})\phi(\boldsymbol{x})d\boldsymbol{x} = \mathbb{E}_p[\phi(\boldsymbol{x})]
\end{aligned}$$

□

Given Theorem 5, we have the following result:

$$\mathbb{E}_p[\log(x_i|\rho)] = \Psi(\rho_i) - \Psi\left(\sum_{j=1}^{k}\rho_j\right) \tag{Ap.14}$$

where Ψ is the *digamma function*: $\Psi(\rho) = \frac{d}{d\rho}\log\Gamma(\rho) = \frac{\Gamma'(\rho)}{\Gamma(\rho)}$.

We now expand the (i)-(vii) terms in $\mathcal{L}$ in Section 3.3:

(i) $\mathbb{E}_q[\log\, p(\phi|\beta)]$

$$\begin{aligned}
\mathbb{E}_q[\log\, p(\phi|\beta)] &= \mathbb{E}_q\left[\log(\exp\Big(\big(\sum_{t=1}^{T}\sum_{w=1}^{V}(\beta-1)\log(\phi_w^t)\big) + \log\Gamma\big(\sum_{w=1}^{V}\beta\big) - \sum_{w=1}^{V}\log\Gamma(\beta)\Big))\right] \\
&= \mathbb{E}_q\left[\big(\sum_{t=1}^{T}\sum_{w=1}^{V}(\beta-1)\log(\phi_w^t)\big) + \log\Gamma\big(\sum_{w=1}^{V}\beta\big) - \sum_{w=1}^{V}\log\Gamma(\beta)\right] \\
&= \sum_{t=1}^{T}\sum_{w=1}^{V}(\beta-1)\mathbb{E}_q\left[\log(\phi_w^t|\eta_w^t)\right] + \log\Gamma\big(V\beta\big) - V\log\Gamma(\beta) \\
&= \sum_{t=1}^{T}\sum_{w=1}^{V}(\beta-1)(\Psi(\eta_w^t) - \Psi\big(\sum_{w'=1}^{V}\eta_{w'}^t\big)) + \log\Gamma\big(V\beta\big) - V\log\Gamma(\beta)
\end{aligned} \tag{Ap.15}$$

(ii) $\mathbb{E}_q[\log\, p(\theta^d|\alpha)]$

$$\begin{aligned}\mathbb{E}_q[\log\, p(\theta^d|\alpha)] &= \mathbb{E}_q\left[\log(\exp\Big(\big(\sum_{t=1}^{T}(\alpha-1)\log(\theta_t^d)\big)+\log\Gamma\big(\sum_{t=1}^{T}\alpha\big)-\sum_{t=1}^{T}\log\Gamma(\alpha)\Big))\right]\\ &= \mathbb{E}_q\left[\big(\sum_{t=1}^{T}(\alpha-1)\log(\theta_t^d)\big)+\log\Gamma\big(\sum_{t=1}^{T}\alpha\big)-\sum_{t=1}^{T}\log\Gamma(\alpha)\right]\\ &= \sum_{t=1}^{T}(\alpha-1)\mathbb{E}_q\left[\log(\theta_t^d|\gamma_t^d)\right]+\log\Gamma\big(T\alpha\big)-T\log\Gamma(\alpha)\\ &= \sum_{t=1}^{T}(\alpha-1)(\Psi(\gamma_t^d)-\Psi\big(\sum_{t'=1}^{T}\gamma_{t'}^d\big))+\log\Gamma\big(T\alpha\big)-T\log\Gamma(\alpha) \quad \text{(Ap.16)}\end{aligned}$$

(iii) $\mathbb{E}_q[\log\, p(\boldsymbol{t}^d|\theta^d)]$

$$\begin{aligned}\mathbb{E}_q[\log\, p(\boldsymbol{t}^d|\theta^d)] &= \mathbb{E}_q\left[\log\prod_{i=1}^{N_d}p(t_i|\theta^d)\right]=\mathbb{E}_q\left[\sum_{i=1}^{N_d}\log p(t_i|\theta^d)\right]\\ &= \mathbb{E}_q\left[\sum_{i=1}^{N_d}\log\prod_{t=1}^{T}(\theta_t^d)^{\zeta_{ti}^d}\right]=\mathbb{E}_q\left[\sum_{i=1}^{N_d}\sum_{t=1}^{T}\zeta_{ti}^d\log(\theta_t^d)\right]\\ &= \mathbb{E}_q\left[\sum_{t=1}^{T}\sum_{w=1}^{V}n_d^w\zeta_{tw}^d\log(\theta_t^d)\right]=\sum_{t=1}^{T}\sum_{w=1}^{V}n_d^w\zeta_{tw}^d\mathbb{E}_q\left[\log(\theta_t^d)\right]\\ &= \sum_{t=1}^{T}\sum_{w=1}^{V}n_d^w\zeta_{tw}^d(\Psi(\gamma_t^d)-\Psi\big(\sum_{t'=1}^{T}\gamma_{t'}^d\big)) \quad \text{(Ap.17)}\end{aligned}$$

NB: ζ_{ti}^d is temporarily use to notate "the (t,w) cell in ζ for w_i of type w."

(iv) $\mathbb{E}_q[\log\, p(\boldsymbol{w}^d|\phi^{\boldsymbol{t}^d})]$

$$\begin{aligned}\mathbb{E}_q[\log\, p(\boldsymbol{w}^d|\phi^{\boldsymbol{t}^d})] &= \mathbb{E}_q\left[\log\prod_{i=1}^{N_d}p(w_i|\phi^{t_i})\right]=\mathbb{E}_q\left[\sum_{i=1}^{N_d}\log p(w_i|\phi^{t_i})\right]\\ &= \mathbb{E}_q\left[\sum_{i=1}^{N_d}\log\prod_{t=1}^{T}\prod_{w=1}^{V}((\phi_w^t)^{\zeta_{ti}^d})^{n_{w_i}^d}\right]=\mathbb{E}_q\left[\sum_{i=1}^{N_d}\sum_{t=1}^{T}\sum_{w=1}^{V}n_{w_i}^d\zeta_{ti}^d\log(\phi_w^t)\right]\\ &= \mathbb{E}_q\left[\sum_{t=1}^{T}\sum_{w=1}^{V}n_w^d\zeta_{tw}^d\log(\phi_w^t)\right]=\sum_{t=1}^{T}\sum_{w=1}^{V}n_w^d\zeta_{tw}^d\mathbb{E}_q\left[\log(\phi_w^t)\right]\\ &= \sum_{t=1}^{T}\sum_{w=1}^{V}n_w^d\zeta_{tw}^d(\Psi(\eta_w^t)-\Psi\big(\sum_{w'=1}^{V}\eta_{w'}^t\big)) \quad \text{(Ap.18)}\end{aligned}$$

NB: ζ_{ti}^d as in (iii). $n_{w_i}^d$ temporarily for "# words that are of the type of the token w_i".

(v) $\mathbb{E}_q[\log\, q(\phi)]$

$$\begin{aligned}
\mathbb{E}_q[\log\, q(\phi)] &= \mathbb{E}_q[\log\, q(\phi|\eta)] \\
&= \mathbb{E}_q\left[\log(\exp\Big(\big(\sum_{t=1}^{T}\sum_{w=1}^{V}(\eta_w^t-1)\log(\phi_w^t)\big)+\log\Gamma\big(\sum_{w=1}^{V}\eta_w^t\big)-\sum_{w=1}^{V}\log\Gamma(\eta_w^t)\Big))\right] \\
&= \mathbb{E}_q\left[\big(\sum_{t=1}^{T}\sum_{w=1}^{V}(\eta_w^t-1)\log(\phi_w^t)\big)+\log\Gamma\big(\sum_{w=1}^{V}\eta_w^t\big)-\sum_{w=1}^{V}\log\Gamma(\eta_w^t)\right] \\
&= \sum_{t=1}^{T}\sum_{w=1}^{V}(\eta_w^t-1)\mathbb{E}_q\left[\log(\phi_w^t|\eta_w^t)\right]+\log\Gamma\big(\sum_{w=1}^{V}\eta_w^t\big)-\sum_{w=1}^{V}\log\Gamma(\eta_w^t) \\
&= \sum_{t=1}^{T}\sum_{w=1}^{V}(\eta_w^t-1)(\Psi(\eta_w^t)-\Psi\big(\sum_{w'=1}^{V}\eta_{w'}^t\big))+\log\Gamma\big(\sum_{w=1}^{V}\eta_w^t\big)-\sum_{w=1}^{V}\log\Gamma(\eta_w^t)
\end{aligned} \tag{Ap.19}$$

(vi) $\mathbb{E}_q[\log\, q(\theta^d)]$

$$\begin{aligned}
\mathbb{E}_q[\log\, q(\theta^d)] &= \mathbb{E}_q[\log\, q(\theta^d|\gamma^d)] \\
&= \mathbb{E}_q\left[\log(\exp\Big(\big(\sum_{t=1}^{T}(\gamma_t^d-1)\log(\theta_t^d)\big)+\log\Gamma\big(\sum_{t=1}^{T}\gamma_t^d\big)-\sum_{t=1}^{T}\log\Gamma(\gamma_t^d)\Big))\right] \\
&= \mathbb{E}_q\left[\big(\sum_{t=1}^{T}(\gamma_t^d-1)\log(\theta_t^d)\big)+\log\Gamma\big(\sum_{t=1}^{T}\gamma_t^d\big)-\sum_{t=1}^{T}\log\Gamma(\gamma_t^d)\right] \\
&= \sum_{t=1}^{T}(\gamma_t^d-1)\mathbb{E}_q\left[\log(\theta_t^d|\gamma_t^d)\right]+\log\Gamma\big(\sum_{t=1}^{T}\gamma_t^d\big)-\sum_{t=1}^{T}\log\Gamma(\gamma_t^d) \\
&= \sum_{t=1}^{T}(\gamma_t^d-1)(\Psi(\gamma_t^d)-\Psi\big(\sum_{t'=1}^{T}\gamma_{t'}^d\big))+\log\Gamma\big(\sum_{t=1}^{T}\gamma_t^d\big)-\sum_{t=1}^{T}\log\Gamma(\gamma_t^d)
\end{aligned} \tag{Ap.20}$$

(vii) $\mathbb{E}_q[\log\, q(\boldsymbol{t}^d)]$

$$\begin{aligned}
\mathbb{E}_q[\log\, q(\boldsymbol{t}^d)] = \mathbb{E}_q[\log\, q(\boldsymbol{t}^d|\zeta^d)] &= \mathbb{E}_q\left[\log\prod_{i=1}^{N_d}q(t_i|\zeta^d)\right] \\
&= \mathbb{E}_q\left[\sum_{i=1}^{N_d}\log q(t_i|\zeta^d)\right] = \mathbb{E}_q\left[\sum_{i=1}^{N_d}\log\prod_{t=1}^{T}(\zeta_{ti}^d)^{\zeta_{ti}^d}\right] \\
&= \mathbb{E}_q\left[\sum_{i=1}^{N_d}\sum_{t=1}^{T}\zeta_{ti}^d\log(\zeta_{ti}^d)\right] = \mathbb{E}_q\left[\sum_{t=1}^{T}\sum_{w=1}^{V}n_w^d\zeta_{tw}^d\log(\zeta_{tw}^d)\right] \\
&= \sum_{t=1}^{T}\sum_{w=1}^{V}n_w^d\zeta_{tw}^d\mathbb{E}_q\left[\log(\zeta_{tw}^d)\right]
\end{aligned} \tag{Ap.21}$$

References

Mark Andrews, Gabriella Vigliocco, and David Vinson. Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, 116(3):463–498, 2009.

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topics models. In *NIPS*, 2008.

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

David Blei and John Lafferty. Dynamic topic models. In *Proceedings of ICML*, 2006.

David Blei and Jon McAuliffe. Supervised topic models. In *NIPS*, 2007.

David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.

David Blei, Tom Griffiths, and Michael Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *ACM*, 57 (2):1–30, 2010.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. A novel neural topic model and its supervised extension. In *Proceedings of AAAI*, 2015.

Susan Carey and Elsa Bartlett. Acquiring a single new word. *Papers and Reports on Child Language Development*, 15:17–29, 1978.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of NIPS*, 2009.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

Arthur P. Dempster, Nan M. Laird, and Ronald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (1): 1–38, 1977.

Georgiana Dinu and Mirella Lapata. Measuring distributional similarity in context. In *Proceedings of EMNLP*, Cambridge, MA, 2010.

James Foulds and Padhraic Smyth. Annealing paths for the evaluation of topic models. In *UAI*, 2014.

Tom Griffiths, Mark Steyvers, David Blei, and Joshua Tenenbaum. Integrating topics and syntax. In *Proceedings of NIPS*, 2005.

David Hall, Daniel Jurafsky, and Christopher Manning. Studying the history of ideas using topic models. In *Proceedings of EMNLP*, 2008.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18 (7):1527–1554, 2006.

Matthew Hoffman, David Blei, and Francis Bach. Online learning for Latent Dirichlet Allocation. In *Proceedings of NIPS*, 2010.

Matthew Hoffmann and Matthew Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *NIPS*, 2016.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*, 2012.

Daniel Jurafsky and James Martin. *Speech and Language Processing*. Prentice Hall, 2008.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, Cambridge, MA, 2009.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of AAAI*, 2015.

Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Proceedings of NIPS*, 2012.

Angeliki Lazaridou, Elia Bruni, and Marco Baroni. Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of ACL*, 2014.

Mikyung Lee, Zhichao Liu, Ruili Huang, and Weida Tong. Application of dynamic topic models to toxicogenomeics data. In *Proceedings of MCBIOS*, 2016.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 (2):91–110, 2004.

Geoffrey L. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and extensions*. John Wiley & Sons, Hoboken, NJ, 2008.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559, 2005.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL `http://arxiv.org/abs/1301.3781`.

David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of EMNLP*, 2011.

Kevin Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes (2ed.)*. New York: McGraw-Hill, 1984.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of EMNLP*, 2009.

Stephen Roller and Sabine Schulte im Walde. A multimodal lda model integrating textual, cognitive and visual modalities. In *Proceedings of EMNLP*, 2013.

Timothy Rubin, America Chambers, Padraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88:157–208, 2012.

Glenn Shafer. Jeffrey's rule of conditioning. *Philosophy of Science*, 48(3), 1981.

Prasha Shrestha, Sebastian Sierra, Fabio González, Paolo Rosso, Manuel Montes y Gómez, and Thamar Solorio. Convolutional nueral networks for authorship attribution of short texts. *EACL 2017*, 2:669–674, 2017.

James Stewart. *Single Variable Calculus: Early Transcendentals (7ed.)*. Brooks/Cole, Cengage Learning, 2012.

Mark Steyvers and Tom Griffiths. Probabilistic topic models. *In T. Landauer, D.S. McNamara, S. Dennis, and W. Kintsch, eds., Handbook of Latent Semantic Analysis*, 2007.

Yee Whye Teh, Michael Jordan, Matthew Beal, and David Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101 (476):1566–1581, 2006.

Yee Whye Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for Latent Dirichlet Allocation. In *Proceedings of NIPS*, 2007.

Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. *LREC*, 2010.

Hanna Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of ICML*, 2006.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of ICML*, 2009.

Li Wan, Leo Zhu, and Rob Fergus. A hybrid neural network latent topic model. In *Proceedings of AISTATS*, 2015.

Su Wang, Stephen Roller, and Katrin Erk. Distributional model on a diet: One-shot word learning from text only. *CoRR, arXiv:1704.04550*, 2017.

Xing Wei and Brue Croft. LDA-based document models for ad-hoc retrieval. In *Proceedings of SIGIR*, 2006.

Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *SIGKDD*, pages 937–946. ACM, 2009.

Ke Zhai, Jordan Boyd-Graber, Nima Asai, and Mohamad Alkhouja. Mr. LDA: a flexible large scale topic modeling package using variational inference in MapReduce. In *Proceedings of WWW*, 2012.

Jun Zhu, Amr Ahmed, and Eric P. Xing. MedLDA: a general framework of maximum margin supervised topic models. *Journal of Machine Learning Research*, 13:2237–2278, 2012.