

# HOMework 5 LDA AND VAE<sup>1</sup>

10-708 PROBABILISTIC GRAPHICAL MODELS (SPRING 2021)

<http://708.mlcourse.org>

OUT: 3/10/20

DUE: 3/24/20 at 11:59 PM

TAs: Elan, Raunaq, Yiwen

## START HERE: Instructions

**Summary** In Section A, you will derive the update steps of mean field variation inference and VAE. B will guide you through the implementation of a structured SVM and you will compare its performance with a linear SVM.

- **Collaboration policy:** The purpose of student collaboration is to facilitate learning, not to circumvent it. Studying the material in groups is strongly encouraged. It is also allowed to seek help from other students in understanding the material needed to solve a particular homework problem, provided no written notes (including code) are shared, or are taken at that time, and provided learning is facilitated, not circumvented. The actual solution must be done by each student alone. The presence or absence of any form of help or collaboration, whether given or received, must be explicitly stated and disclosed in full by all involved. See the Academic Integrity Section on the course site for more information: <http://www.cs.cmu.edu/~mgormley/courses/10708/about.html#7-academic-integrity-policies>
- **Late Submission Policy:** See the late submission policy here: <http://www.cs.cmu.edu/~mgormley/courses/10708/about.html#6-general-policies>
- **Submitting your work to Gradescope:** We use Gradescope to collect PDF submissions of open-ended questions on the homework (e.g. mathematical derivations, plots, short answers). The course staff will manually grade your submission, and you'll receive feedback explaining your final marks. You will also submit your code for programming questions on the homework to Gradescope (<https://www.gradescope.com/courses/228238>). We will manually grade your code for completeness.
- **Style:** For Section A,  $\LaTeX$  is required for solution. Please make sure you correctly select page for each question on Gradescope and do not change the size of the solution box. Failure to comply to the required style with result in deduction of style points.
- For **multiple choice** or **select all that apply questions**, shade in the box or circle in the template document corresponding to the correct answer(s) for each of the questions. For  $\LaTeX$  users, replace `\choice` with `\CorrectChoice` to obtain a shaded box/circle, and don't change anything else.

---

<sup>1</sup>Compiled on Friday 9<sup>th</sup> April, 2021 at 21:04

## A Written Questions[65 pts]

Answer the following questions in the template provided. Then upload your solutions to Gradescope. You have to use  $\LaTeX$ . Failure to follow the style will result in a penalty.

### A.1 Variational Inference[40 pts]

1. In this problem, we are going to work with approximate posterior inference via variational inference for a given topic model.

The standard Latent Dirichlet Allocation model only models the word co-occurrences, without considering temporal information, i.e. the time when a document is generated. However, a large number of subjects in documents change dramatically over time. It is important to interpret the topics in the context of the timestamps of the documents. To address how topics occur and shift over time, Topics on Time (TOT) model was proposed, by explicitly modeling of time jointly with word co-occurrence patterns [Wang and McCallum, 2006]. The model is shown in Figure A.1.

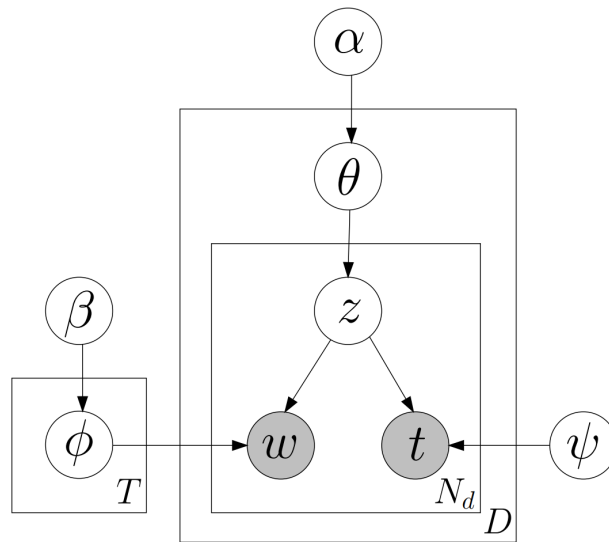


Figure A.1: TOT Model

In the model, there are  $D$  documents. Each document  $d$  contains  $N_d$  words  $w_{d1}, w_{d2}, \dots, w_{dN_d}$ . Each word  $w_{di}$  has a timestamp  $t_{di} \in (0, 1)$ , indicating when the document is generated in a relative time scale  $(0, 1)$ . All words in the same document have the same timestamp. There are  $K$  topics (also  $T = K$  topics for the notation in the paper and Figure A.1) in the document corpora. Each topic follows a multinomial distribution  $\phi$  over the  $V$  words in the vocabulary. Each document follows a multinomial distribution  $\theta$  over the  $K$  topics. The prior distribution for  $\phi$  and  $\theta$  are Dirichlet distributions with parameters  $\beta$  and  $\alpha$  respectively. For each topic  $k$ , the temporal occurrence follows a Beta distribution  $Beta(\psi_{k1}, \psi_{k2})$ , where  $\psi_k = (\psi_{k1}, \psi_{k2})$  and we use  $\psi \in \mathbb{R}_+^{K \times 2}$  to denote  $\psi_k$  for all topics. Each word  $w_{di}$  and its timestamp  $t_{di}$  are assumed to be generated from a topic, with a topic label  $z_{di} \in \{1, \dots, K\}$ .

The generative process of this model is described as follows.

1. Draw  $K$  multinomials  $\phi_k$  from a Dirichlet prior  $\beta$ , one for each topic  $k$ .
2. For each document  $d$ ,
  - Draw a multinomial  $\theta_d$  from a Dirichlet prior  $\alpha$ ;
  - For each word  $w_{di}$  in document  $d$ .
    - (a) Sample a topic  $z_{di}$  from multinomial  $\theta_d$ ;
    - (b) Sample a word  $w_{di}$  from multinomial  $\phi_{z_{di}}$ ;
    - (c) Sample a timestamp  $t_{di}$  from Beta  $\psi_{z_{di}}$ .

We use variational EM to approximate the posterior of latent variables and learn model parameters. To do this, a mean field variational distribution needs to be defined, which is parameterized by some parameters called variational parameters. The variational EM algorithm iteratively performs two steps: 1) in the E step, variational parameters are updated; 2) in the M step, model parameters are optimized. Same as in the paper, we consider  $\alpha$  and  $\beta$  are predefined fixed hyperparameters with no need to update. Therefore, in M step, only the other model parameters are optimized. The pseudo-code for the proposed algorithm is shown in Algorithm 1.

---

**Algorithm 1** Pseudo-code of variational EM algorithm for TOT model
 

---

- 1: **Input:** Observations, Topic number  $K$ , MaxIter, and other optional parameters
  - 2: **Output:** Posterior distributions for latent variables, optimized model parameters
  - 3: Initialize parameters;
  - 4: Compute and record ELBO with initial parameters
  - 5: **for**  $k \leftarrow 1$  to  $MaxIter$  **do**
  - 6:   Update variational variables ▷ Stage 1: E-Step
  - 7:   Update  $\psi$  with projected Newton method ▷ Stage 2: M-Step
  - 8:   Compute and record ELBO
  - 9: **end for**
- 

In the TOT model,  $\theta, \mathbf{z}, \phi$  are latent variables and  $\psi$  is the model parameter to be learned. As a start, we use mean-field variational inference and the variational distribution has the form:

$$q(\theta, \phi, \mathbf{z} | \gamma, \lambda, \pi) = \prod_{k=1}^K q(\phi_k | \lambda_k) \prod_{d=1}^D [q(\theta_d | \gamma_d) \prod_{n=1}^{N_d} q(z_{dn} | \pi_{dn})], \quad (\text{A.1})$$

where  $\gamma, \lambda, \pi$  are variational parameters that need to be updated in E-step.

Next, we write out the joint distribution of latent and observed variables:

$$p(\mathbf{x}, \mathbf{t}, \phi, \theta, \mathbf{z} | \alpha, \beta, \psi) = \prod_{k=1}^K p(\phi_k | \beta) \prod_{d=1}^D [p(\theta_d | \alpha) \prod_{n=1}^{N_d} p(z_{dn} | \theta_d) p(x_{dn} | z_{dn}, \phi) p(t_{dn} | z_{dn}, \psi)] \quad (\text{A.2})$$

Given Eq. A.1 and A.2, we can write out ELBO with:

$$\text{ELBO} = \mathbb{E}_{q(\theta, \phi, \mathbf{z})} [\log p(\mathbf{x}, \mathbf{t}, \phi, \theta, \mathbf{z}) - \log q(\theta, \phi, \mathbf{z})]. \quad (\text{A.3})$$

Variational EM basically maximizes ELBO w.r.t. variational parameters and model parameters in E- and M-step respectively.

**A.1.1 Update variational parameters**

Derive the update equations of variational parameters, and also specify their distributions. Here you can directly use the conclusion below for the derivation.

$$q_j^* \propto \exp\{\mathbb{E}_{q_{-j}}[\log p(\mathbf{x}, \mathbf{t}, \boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{z})]\},$$

where  $\mathbb{E}_{q_{-j}}$  denotes expectation over all latent variables excluding variable  $j$ .

(5 pts) Derive  $q(\boldsymbol{\theta})$ . Specify the distribution and parameters.

(5 pts) Derive  $q(\boldsymbol{\phi})$ . Specify the distribution and parameters.

- (5 pts) Derive  $q(\mathbf{z})$ . Specify the distribution and parameters. Use  $\Psi(\cdot)$  for digamma function and  $B$  for Beta function.

### A.1.2 Update model parameters

Derive the update equations of model parameters, as mentioned before, there is no need to update  $\alpha$  and  $\beta$ . For the updating rule of  $\psi$ , please be careful that  $\psi$  should be constrained as positive. Hint: For a problem with positive solution ( $x > 0$ ), a projected Newton method could be applied:

$$y = (x - H^{-1}g)_+$$

$$x^+ = x + \lambda(y - x)$$

where  $x$  is the current variable,  $y$  is the projected update,  $g$  and  $H$  are gradient and Hessian matrix respectively,  $\lambda$  is the step size and  $x^+$  is the updated variable.  $(\cdot)_+$  is defined as  $s_+ := \max(0, s)$ .

- (4 pts) Write out the terms in ELBO that contains  $\psi_k$ .

(5 pts) Derive the first and second order derivative of  $\text{ELBO}(\psi_k)$  with respect to  $\psi_k$ .

(1 pt) Derive the projected Newton method to update  $\psi_k$ .

### A.1.3 Detailed variational lower bound

Based on the variational distributions, expand Eq. A.3 to obtain detailed variational lower bound. The result should be as specific as possible, that is, it can be directly used in the implementation. To compute detailed variational lower bound, We next expand the terms in Eq. (A.3) one by one.

(2 pts) Write out  $\mathbb{E}_q[\sum_d \log p(\theta_d|\alpha)]$ .

(2 pts) Write out  $\mathbb{E}_q \left[ \sum_k \log p(\phi_k | \beta) \right]$ .

(2 pts) Derive  $\sum_{d,n} \mathbb{E}_q \left[ \log p(z_{dn} | \theta_d) \right]$ .

(2 pts) Derive  $\sum_{d,n} \mathbb{E}_q \left[ \log p(w_{dn} | z_{dn}, \phi) \right]$ .

(2 pts) Write out  $\sum_{d,n} \mathbb{E}_q \left[ \log p(t_{dn} | z_{dn}, \psi) \right]$ .

(2 pts) Write out  $\sum_d \mathbb{E}_q[\log q(\theta_d)]$ .

(2 pts) Write out  $\sum_{d,n} \mathbb{E}_q[\log q(z_{dn})]$ .

(2 pts) Write out  $\sum_k \mathbb{E}_q[\log q(\phi_k)]$ .



(4 pts) Write out the ELBO.

**Hint:** the problem is designed based on the paper [[Wang and McCallum, 2006](#)]. In the paper, Gibbs sampling was used for posterior inference, and here we are working with variational inference. You may gain better understanding of the model and get some ideas of how to solve the problem by reading the paper.

## A.2 Variational Autoencoders [25 pts]

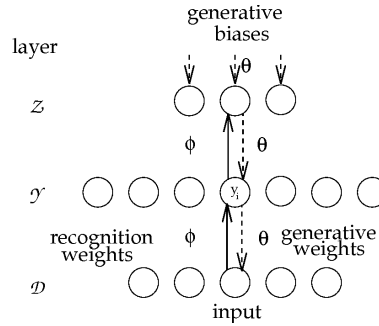


Figure A.2: A **Helmholtz machine** [Hinton et al. \[1995\]](#), [Dayan \[2000\]](#) contains two networks: (1) bottom-up “recognition” connections  $\phi$  that convert the input data into representations in successive hidden layers, and (2) top-down “generative” connections  $\theta$  that reconstruct the data from representation in one layer from the representation in the layer above.

The Helmholtz machine (Figure A.2) is an architecture that can find hidden structure in data by learning a generative model of the data. Helmholtz machines are usually trained using unsupervised learning algorithms such as the classical **Wake-Sleep** algorithm [Hinton et al. \[1995\]](#) or the modern **Auto-Encoding Variational Bayes (AEVB)** [Kingma and Welling \[2013\]](#), also known as variational autoencoder.

In this problem, you will (re-)derive the AEVB algorithm. The sections are organized as follows:

- (5 pts) Section A.2.1: Derivation of the **evidence lower bound objective (ELBO)**, which lowerbounds the data log-likelihood  $\log p_\theta(\mathbf{x})$ .
- (15 pts) Section A.2.2: Derivation of **AEVB**, which optimizes a stochastic estimate of ELBO.
- (5 pts) Section A.2.3: Derivation of an **alternate lower bound**  $\mathcal{L}_k(\mathbf{x})$  for the data log-likelihood, which will be used to evaluate trained models in the next section.

For all parts, assume the following:

- Assume a Gaussian prior on  $\mathbf{z} \sim N(0, I)$ , and let  $\mathbf{x}$  be binary vectors. In other words,  $p_\theta(\mathbf{x} | \mathbf{z})$  can be modeled with a sigmoid belief net, so the likelihood is of the form  $p_\theta(\mathbf{x} | \mathbf{z}) = \text{Bernoulli}(f_\theta(\mathbf{z}))$ . Actually, the data points  $\mathbf{x}$  in MNIST take values in  $[0, 1]$  rather than  $\{0, 1\}$ , but the loss term  $\mathbb{E}_q[p_\theta(\mathbf{x} | \mathbf{z})]$  still uses sigmoid cross-entropy loss, which is a common practice [Doersch \[2016\]](#).
- The variational distribution  $q_\phi$  is parameterized by a Gaussian, i.e.,  $q_\phi(\mathbf{z} | \mathbf{x}) = N(\mathbf{z}; \mu_\phi(\mathbf{x}), \Sigma_\phi(\mathbf{x}))$ .

### A.2.1 Evidence Lower Bound Objective (ELBO)

Suppose we want to learn a directed latent variable model (Figure A.3) that is able represent a complex distribution  $p(\mathbf{x})$  over the data in the following form:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x} | \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \quad (\text{A.4})$$

Suppose we want to approximate the posterior distribution  $p_\theta(\mathbf{z} | \mathbf{x})$  using some variational distribution  $q_\phi(\mathbf{z} | \mathbf{x})$ . A tractable way to learn this model is to optimize the **evidence lower bound objective**

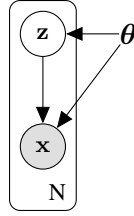


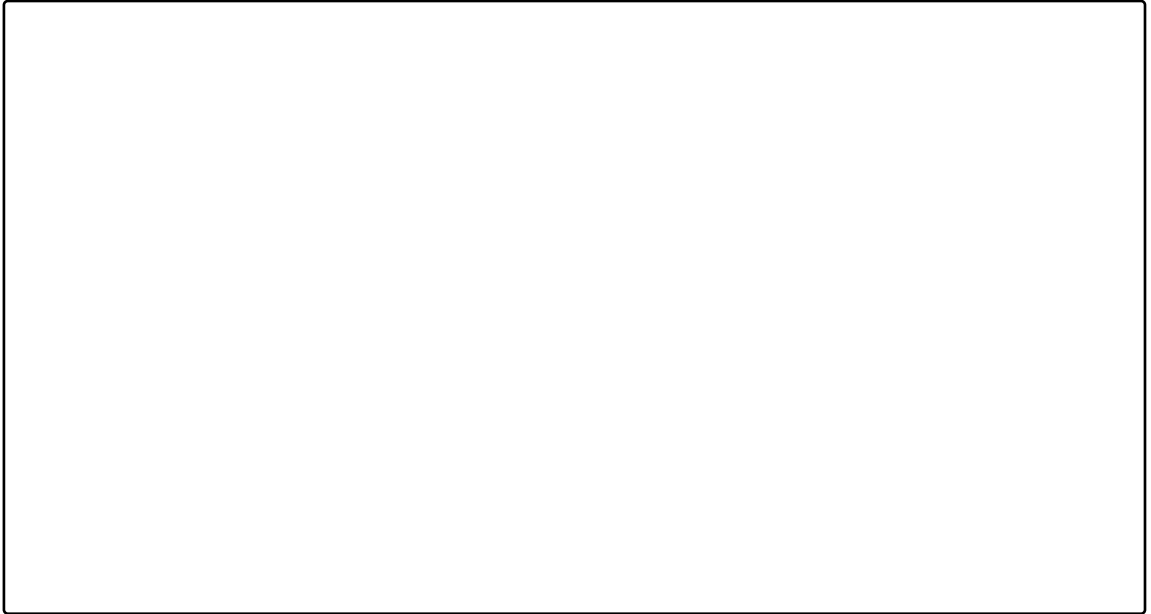
Figure A.3: The latent variable model in Problem A.2.1.

(ELBO), also known as the variational lower bound, defined as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{x}) &= \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z} | \mathbf{x})} [\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z} | \mathbf{x})] \\ &= \int_{\mathbf{z}} q_{\phi}(\mathbf{z} | \mathbf{x}) \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z} | \mathbf{x})} d\mathbf{z} .\end{aligned}$$

(5 pts) For a single data point  $\mathbf{x}^{(i)}$ , prove that

$$\log p_{\theta}(\mathbf{x}^{(i)}) \geq \mathcal{L}(\mathbf{x}^{(i)}) .$$



The above result shows that, for iid data points  $\mathbf{x} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ ,

$$\log p_{\theta}(\mathbf{x}) \stackrel{\text{iid}}{=} \sum_{i=1}^N \log p_{\theta}(\mathbf{x}^{(i)}) \geq \sum_{i=1}^N \mathcal{L}(\mathbf{x}^{(i)}) = \mathcal{L}(\mathbf{x})$$

which gives the ELBO  $\mathcal{L}(\mathbf{x})$  on the data log-likelihood  $\log p_{\theta}(\mathbf{x})$ .

### A.2.2 Autoencoding Variational Bayes (AEVB)

In this section, you will derive the optimization procedure for Auto-Encoding Variational Bayes (AEVB). Unlike Wake-Sleep, AEVB avoids the two-stage optimization procedure and instead optimizes a stochastic estimate of ELBO directly w.r.t. to parameters  $\theta$  of the generative model (generation network) and parameters  $\phi$  of the variational distribution (recognition network).

(5 pts) For a given data point  $\mathbf{x}^{(i)}$ , show that ELBO can be rewritten as

$$\mathcal{L}(\mathbf{x}^{(i)}) = -D_{KL} \left( q_\phi(\mathbf{z} \mid \mathbf{x}^{(i)}) \parallel p(\mathbf{z}) \right) + \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x}^{(i)})} [\log p_\theta(\mathbf{x}^{(i)} \mid \mathbf{z})]. \quad (\text{A.5})$$



Equation (A.5) gives a stochastic estimator for ELBO:

$$\tilde{\mathcal{L}}(\mathbf{x}^{(i)}) = -D_{KL} \left( q_\phi(\mathbf{z} \mid \mathbf{x}^{(i)}) \parallel p(\mathbf{z}) \right) + \frac{1}{L} \sum_{l=1}^L [\log p_\theta(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i,l)})] \quad (\text{A.6})$$

where  $\{\mathbf{z}^{(i,l)}\}_{l=1}^L$  are sampled from  $q_\phi(\mathbf{z} \mid \mathbf{x}^{(i)})$ . The AEVB algorithm optimizes this stochastic estimate of ELBO using a Monte Carlo gradient estimate.

In order to optimize the AEVB objective in Eq. (A.6) efficiently, we use a **reparameterization trick** to rewrite  $\mathbb{E}_{q_\phi(\mathbf{z} \mid \mathbf{x})}[\cdot]$  such that the Monte Carlo estimate of the expectation is differentiable w.r.t.  $\phi$ . More specifically, we reparameterize the latent variable

$$\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x}^{(i)}) = N(\mathbf{z} \mid \mu_\phi(\mathbf{x}^{(i)}), \Sigma_\phi^2(\mathbf{x}^{(i)}))$$

as a deterministic function of the input  $\mathbf{x}^{(i)}$  and an auxiliary noise variable  $\epsilon$ :

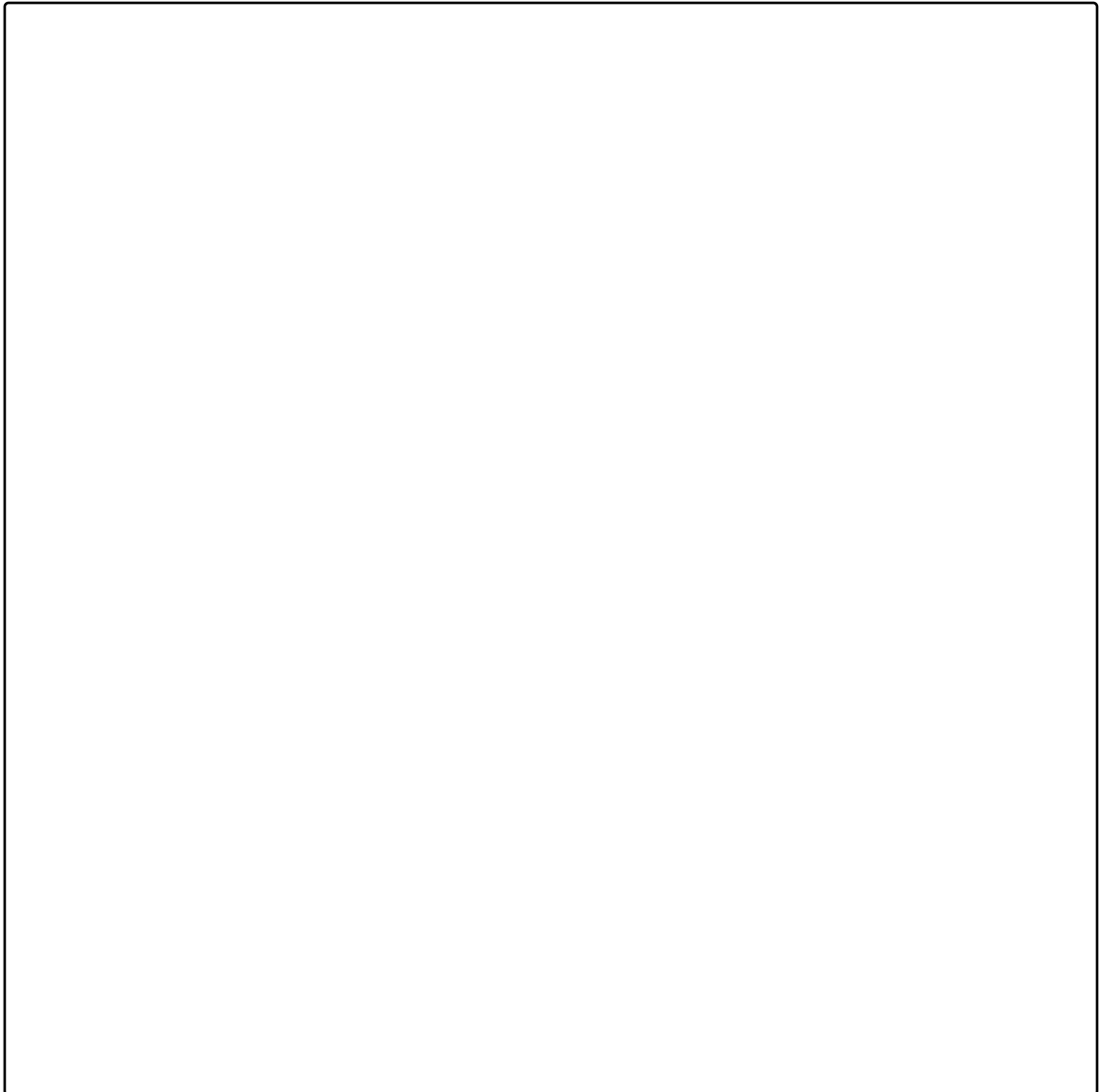
$$\mathbf{z} = \mu_\phi(\mathbf{x}^{(i)}) + \Sigma_\phi(\mathbf{x}^{(i)}) \odot \epsilon \quad \epsilon \sim N(0, I) \quad (\text{A.7})$$

where  $\odot$  signifies an element-wise product, and  $\Sigma_\phi(\mathbf{x}^{(i)})$  is a diagonal matrix.

(5 pts) Using this reparameterization, show that the AEVB objective in Eq. (A.6) can be rewritten as

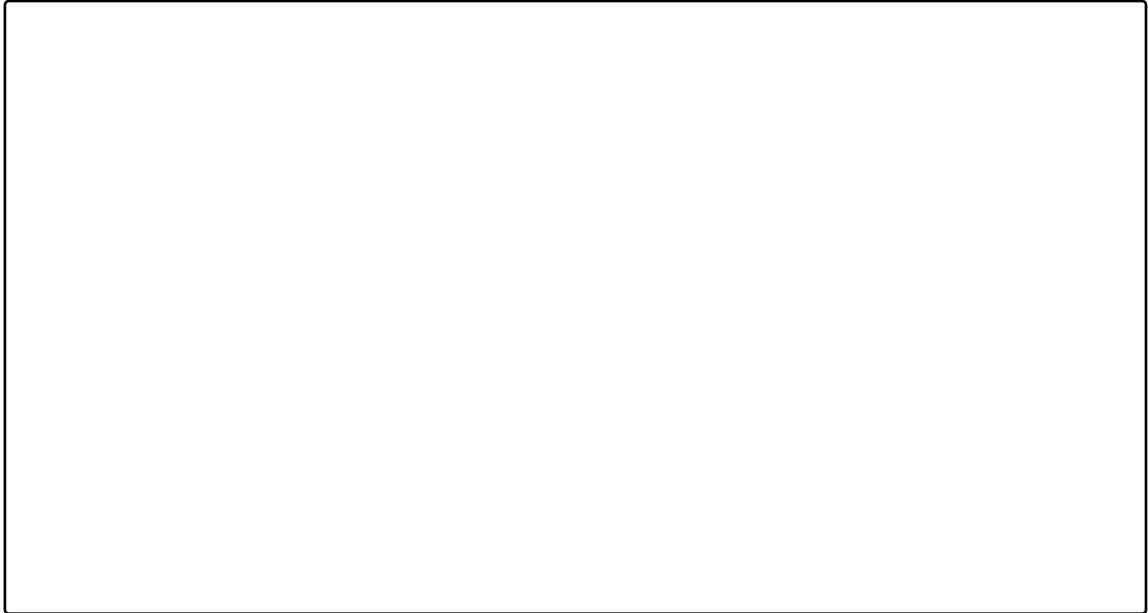
$$\tilde{\mathcal{L}}(\mathbf{x}^{(i)}) = \frac{1}{2} \sum_{j=1}^J \left( 1 + \log(\Sigma_{(i),j}^2) - \mu_{(i),j}^2 - \Sigma_{(i),j}^2 \right) + \frac{1}{L} \sum_{l=1}^L \log p_\theta(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i,l)}). \quad (\text{A.8})$$

where  $\mu_{(i)} := \mu_\phi(\mathbf{x}^{(i)})$  and  $\Sigma_{(i)} := \Sigma_\phi(\mathbf{x}^{(i)})$ .



The AEVB optimization procedure works as follows:

1. For each  $l \in [L]$ , draw  $\epsilon^{(l)} \sim N(0, I)$ , and compute  $\mathbf{z}^{(i,l)} := \mu_{(i)} + \Sigma_{(i)} \odot \epsilon^{(l)}$ .
  2. Optimize the AEVB objective in Eq. (A.8) w.r.t.  $\mu$ ,  $\Sigma$ , and  $\theta$ .
- (5 pt) Derive the gradients of the AEVB objective in Eq. (A.8) w.r.t.  $\mu_{(i),j}$ ,  $\Sigma_{(i),j}$ , and  $\theta$ . (For the gradient w.r.t.  $\theta$ , you can leave the answer in terms of  $p_{\theta}(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i,l)})$ .)

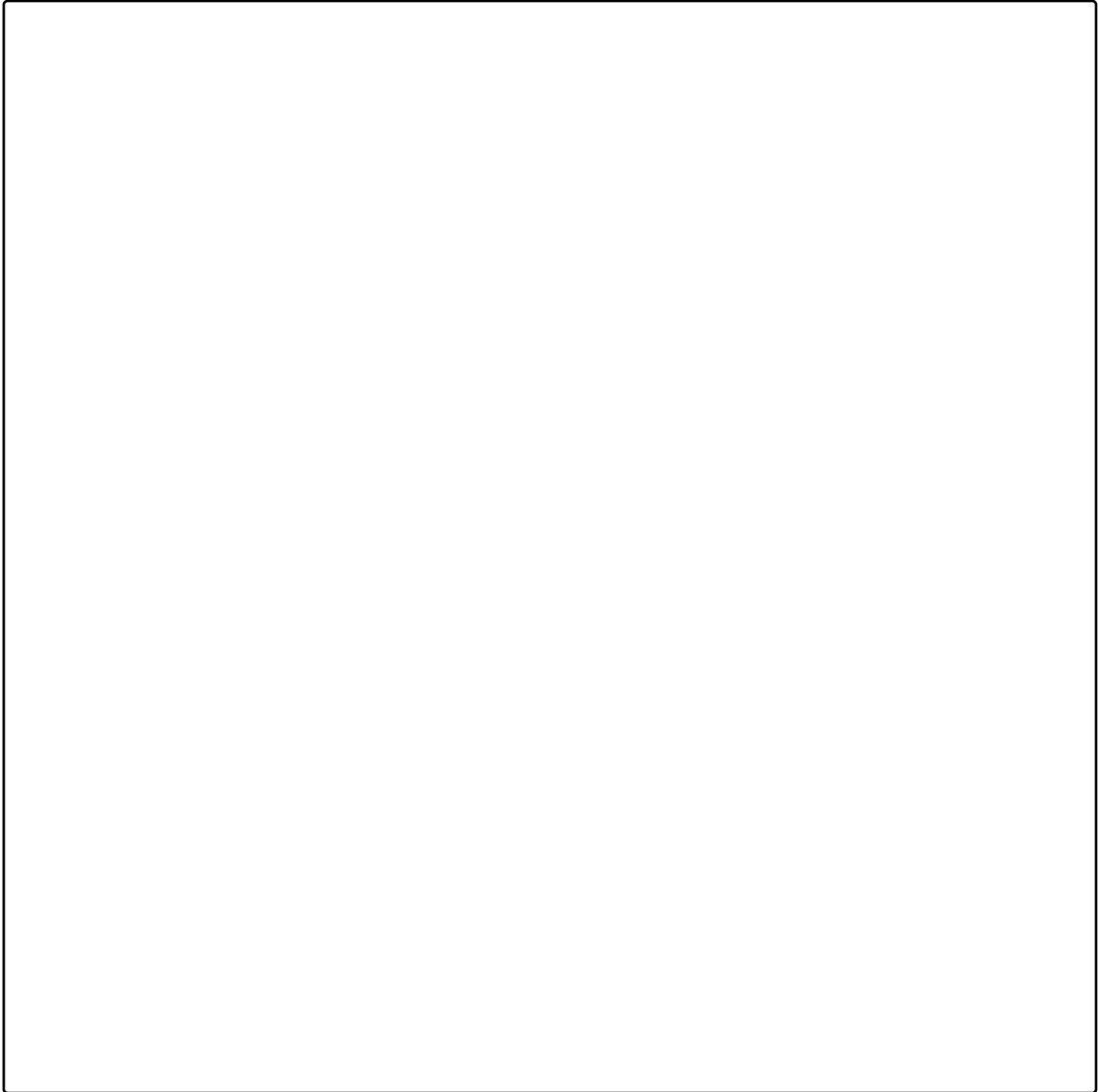


### A.2.3 An Alternate Lower Bound on the Log-Likelihood

To compare trained models, we could simply look at the values of the lower bound. However, the bound could be loose and hence the numbers could be misleading. Here, we derive and prove a tighter approximation of the lower bound on the marginal likelihood, defined as follows:

$$\mathcal{L}_k(\mathbf{x}) = \mathbb{E}_{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(k)} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{1}{k} \sum_{i=1}^k \frac{p_\theta(\mathbf{x}, \mathbf{z}^{(i)})}{q_\phi(\mathbf{z}^{(i)} | \mathbf{x})} \right]. \quad (\text{A.9})$$

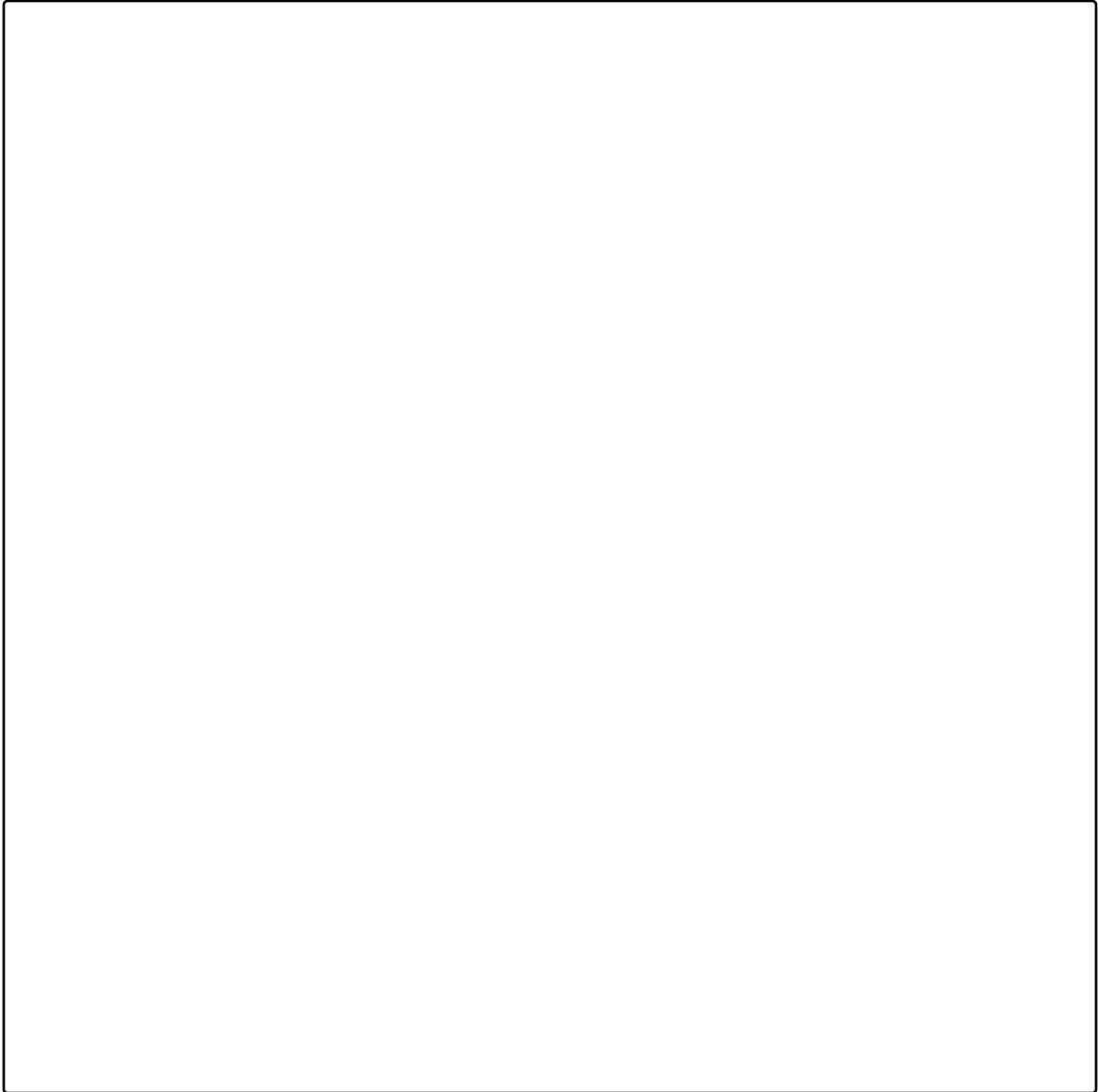
(3 pts) Prove that  $\log p(\mathbf{x}) \geq \mathcal{L}_k(\mathbf{x})$  for any  $k \in \mathbb{N}$ . (*Hint: Use Jensen's inequality.*)



(2 pts) Prove that  $\mathcal{L}_{k+1}(\mathbf{x}) \geq \mathcal{L}_k(\mathbf{x})$  for any  $k \in \mathbb{N}$ . You can use the following lemma without proof:

*Lemma:* Let  $I_k \subset [k+1] := \{1, \dots, k+1\}$  with  $|I_k| = k$  be a uniformly distributed subset of distinct indices from  $[k+1]$ . Then for any sequence of numbers  $a_1, \dots, a_{k+1}$ ,

$$\mathbb{E}_{I_k} \left[ \frac{\sum_{i \in I_k} a_i}{k} \right] = \frac{\sum_{i=1}^{k+1} a_i}{k+1} \quad (\text{A.10})$$



The above two results show that

$$\log p(\mathbf{x}) \geq \mathcal{L}_{k+1}(\mathbf{x}) \geq \mathcal{L}_k(\mathbf{x}).$$

However, the above inequalities do not guarantee  $\mathcal{L}_k(\mathbf{x}) \rightarrow \log p(\mathbf{x})$  when  $k \rightarrow \infty$ . (The proof is left as an exercise to the reader.)



## B Programming [35 pts]

For the programming portion of this homework, we’re going to focus less on complex software implementations and more on being able to see the amazing results that are possible with variational inference. Using a corpus of articles published in the New York Times, we will be learning the parameters of an LDA model (the original one, from 2003). Then, we’ll ask you to do some exploration with the model you’ve learned.

Since we’ve already derived the updates for the TOT model, we won’t be asking you to do any more math for this section. Instead, we will give an overview of the derivation of each of the updates, and you will just be asked to implement these steps in an efficient manner to perform inference. Then you’ll have the opportunity to see how this algorithm uncovers meaningful latent structure in the data!

### B.1 Vanilla LDA

The standard Latent Dirichlet Allocation model is a simpler version of the model you considered in the written section. The generative model is shown in plate notation in [B.1](#), where  $M$  is the number of documents and  $N$  is the number of words per document—for simplicity and ease of computation, we assume that each document has the same number of words, and in fact you will be writing code to ensure that this is the case.

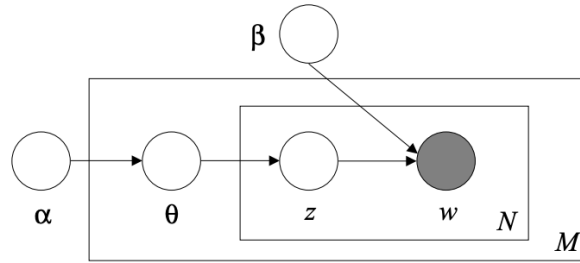


Figure B.1: LDA Generative Model

There are two major differences between this model and the one in [figure A.1](#). First, observe that we do not model the time occurrence of each word. Second, this simpler model does not assume a prior over topic multinomials (that is, each topic’s unique distribution over word probabilities). Instead, we assume that there is a fixed number of topics  $k$  and that each topic’s distribution over words is fixed but unknown. Our task is thus to estimate these parameters, as well as the latent parameters representing each document’s distribution over topics and each word’s latent topic. We will do so with a classical algorithm known as *Expectation-Maximization*, or the EM algorithm.

#### B.1.1 Expectation Step

Given a document, our first objective is to compute the posterior distribution on the latent variables:

$$p(\theta, z \mid w, \alpha, \beta) = \frac{p(\theta, z, w \mid \alpha, \beta)}{p(w \mid \alpha, \beta)}.$$

As usual, this is intractable due to the need to marginalize out the latents in the denominator of this expression. Instead, we turn to variational inference!

[Figure B.2](#) depicts our chosen variational distribution which we will use to approximate the intractable posterior. Observe that we have dropped the prior  $\alpha$  over topic distributions which is shared by all documents; instead, our approximation assumes that each document’s topic distribution  $\theta$  is drawn as a function of the variational parameter  $\gamma$ . Likewise, each word has its own variational distribution over topics.

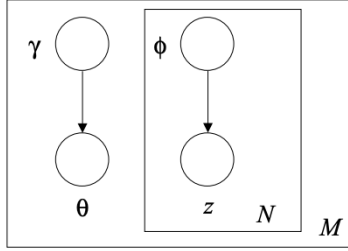


Figure B.2: Graphical Representation of the Variational Distribution

With the variational parameters defined, we can write the variational distribution for a single document as follows:

$$q(\theta, z \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_{n=1}^N q(z_n \mid \phi_n). \quad (\text{B.1})$$

Recall that standard variational inference (that is, not *amortized* as in a VAE) uses separate variables for each observation. So, we will be optimizing the variational parameters separately for each document.

Our goal is to optimize the variational parameters so as to minimize the KL Divergence from our variational distribution to the true posterior. That is,

$$(\gamma^*, \phi^*) = \arg \min_{\gamma, \phi} D(q(\theta, z \mid \gamma, \phi) \parallel p(\theta, z \mid w, \alpha, \beta)). \quad (\text{B.2})$$

We will minimize this objective by repeatedly solving for a fixed point. Taking the derivative of the KL Divergence, setting it equal to 0, and solving gives us a set of updates which ensure that our parameter estimates will converge to the optimum. As promised, we will not get too detailed in defining these updates, but the derivations and the intuitions behind them can be found in the original LDA paper [Blei et al. \[2003\]](#).

$$\phi_{ni} \propto \beta_{iw_n} \exp\{\mathbb{E}_q[\log \theta_i \mid \gamma]\}, \quad (\text{B.3})$$

$$\gamma_i = \alpha_i + \sum_{n=1}^N \phi_{ni}. \quad (\text{B.4})$$

Here,  $i$  indexes the topic and  $n$  indexes the word.  $w_n$  is the vocabulary index of the  $n^{\text{th}}$  word. Finally, the expectation term in the above update can be evaluated as

$$\mathbb{E}_q[\log \theta_i \mid \gamma] = \Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right), \quad (\text{B.5})$$

where  $\Psi$  is the derivative of the log of the Gamma function, also known as the *digamma function*. Observe that the second term in this expectation can be ignored, because we are only solving for the updated  $\phi$  up to proportionality (it will be normalized after each iteration to sum to 1 for each word).

These updates, repeated until convergence, give us the *Expectation* step of our expectation-maximization algorithm. For a fixed (assumed known)  $\alpha, \beta$ , we are optimizing the parameters to ensure our variational distribution gives as close an approximation to the true posterior as possible.

When implementing this algorithm, you should initialize the multinomial parameters  $\phi_{ni} = \frac{1}{k}$  for all  $n, i$ , and set  $\gamma_i = \alpha_i + N/k$ .

### B.1.2 Maximization Step

The above algorithm is only one half of our expectation-maximization procedure. Recall that this objective optimizes the variational parameters for a fixed  $\alpha, \beta$ . Once we have learned these parameters, our next step is to find the choice of  $\alpha, \beta$  which maximizes the resulting lower bound on the log-likelihood of the observed data (hence, the *Maximization* step). We will then cycle between these two procedures until we arrive at a complete solution.

Once again, we will skip the details and give you the precise update:

$$\beta_{ij} \propto \sum_{d=1}^M \sum_{n=1}^N \phi_{dni} w_{dn}^j. \quad (\text{B.6})$$

To implement this update efficiently, it may be helpful to look into the function `numpy.einsum`. In the above expression  $w_{dn}^j$  is an indicator variable which is 1 if and only if the  $n^{\text{th}}$  word of the  $d^{\text{th}}$  document is the  $j^{\text{th}}$  vocabulary word (recall that  $\beta$  is a matrix such that  $\beta_i$  parameterizes a multinomial distribution over words for topic  $i$ ). Additionally, the update to  $\alpha$  is given as

$$\alpha^+ = \alpha + \frac{g - c}{h}, \quad (\text{B.7})$$

where  $c = \frac{\sum_{j=1}^k g_j / h_j}{z^{-1} + \sum_{j=1}^k h_j^{-1}}$ . The  $g$  and  $h$  in these expressions are  $k$ -dimensional vectors which give the gradient, and a particular vector which shows up in the Hessian, of the log-likelihood with respect to  $\alpha$ , respectively. They can be computed as

$$g_i = M \left( \Psi \left( \sum_{j=1}^k \alpha_j \right) - \Psi(\alpha_i) \right) + \sum_{d=1}^M \left( \Psi(\gamma_{di}) - \Psi \left( \sum_{j=1}^k \gamma_{dj} \right) \right), \quad (\text{B.8})$$

$$h_i = M \Psi'(\alpha_i), \quad (\text{B.9})$$

$$z = -\Psi' \left( \sum_{j=1}^k \alpha_j \right) \quad (\text{B.10})$$

Note the use of  $\Psi'$ , not  $\Psi$ ! This is the derivative of the digamma function, or the second derivative of the log Gamma function. It is known as the *polygamma function* of order 1.

### B.1.3 Implementation notes

- For the first E-step, you can initialize each  $\alpha_i = 0.1$ , each  $\beta_{ij} \sim \mathcal{U}(0, 1)$  and normalize each row of  $\beta$ .
- To check convergence for the E-step, you can use the following criterion:

$$\frac{1}{2 \times \text{num\_docs}} \left( \|\phi^{(t)} - \phi^{(t-1)}\| + \|\gamma^{(t)} - \gamma^{(t-1)}\| \right) \leq 10^{-2}$$

- To check convergence of  $\alpha^+$  for the M-step, you can use a tolerance of  $\|\alpha^+ - \alpha\| \leq 10^{-4}$

That's it! After doing the two updates in the M step, you should return to the E step and again iterate until convergence. Cycling between these two steps gives the EM algorithm for learning an LDA model. In the next section, we will discuss the specifics of the dataset and what exactly you will be doing for this programming assignment.

## B.2 LDA on New York Times Articles

In this assignment, we will be using the above variational EM algorithm to learn the parameters of an LDA model for a corpus of articles by the New York Times. You'll also get to do this for an article of your choosing!

The handout includes two files, `nyt_vocab.txt` and `nyt_data.txt`. The former is a simple list of vocabulary words. The latter is a collection of articles coming from the New York Times which has already been partially formatted for you. Each document is on a separate line, encoded as key:value pairs—the key is the index of the vocabulary word as it appears in the vocab file (0-indexed, of course) and the value is the number of times that word appears in the document. Since the LDA model doesn't account for word order, this encodes all the information you need but in an easier format.

Unfortunately, we do not have the original source documents from which these counts were created. As a result, we won't be able to directly tie our learned topics to specific articles. To partially fix this, we want *you* to pick an article/document of your choice to add it to the corpus. Be sure to pick something which is available online **and save the url**. You can choose any body of text you like, but note the following:

- It should be sufficiently long that the number of vocabulary words which occur are somewhere around 100-200 (could be more).
- If you choose something wildly different from what might be covered in the Times, your model may have difficulty picking the right topic(s). We don't think this is likely to be a problem because of the size of the corpus, but just keep this in mind.
- If you like, you can do multiple articles! This will not take any additional work and it will mean you get some more cool results.

(2 pts) Report the title and url of the article you chose and give a brief description of what the article is about.

Now that you've chosen an article, you should copy the raw text into a file. Then, write a script to count the number of occurrences of each vocabulary word and encode the document in the same format as the provided corpus. Finally, append your formatted article to `nyt_data.txt`.

(10 pts) Now for the algorithm! Using the steps described in the previous section, **implement the variational EM algorithm** to learn the parameters for an LDA model of the data. A few specifics:

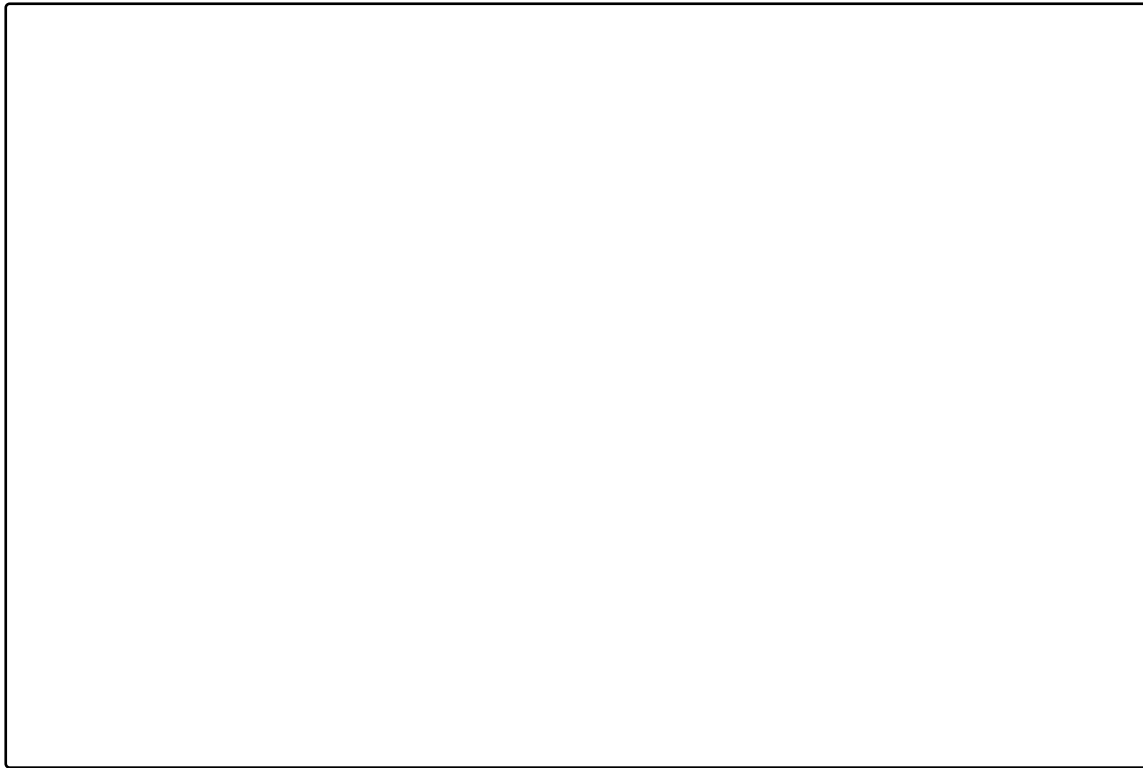
- Recall that we assume a fixed document length  $N$ —for this assignment, we will set  $N = 200$ . First, you should throw away all documents with fewer than 150 words. Then, for documents with *fewer* than  $N$  words, sample  $N$  words from the document uniformly with replacement, and for documents with *more* than  $N$  words, sample exactly  $N$  words *without* replacement.
- We also assume a fixed number of topics  $k$ . We choose to set  $k = 25$ ; this provides a nice balance such that there won't be too many combined topics, but also we won't have “leftover”

topics that don't correlate with anything.

We highly recommend you vectorize functions where applicable and think about which calculations can be evaluated as a matrix product—and keep in mind `numpy.einsum`. **Be sure to save your final parameter estimates!**

Once you've run this algorithm to convergence, you should have three sets of parameters which interest us. The first is  $\alpha$ , the Dirichlet prior over document topics. The second is  $\beta$ , which parameterizes each topic's multinomial distribution over words. The last one is  $\theta$ , the inferred distribution over topics for each document; this one you won't have solved for directly, but it will be the MAP estimate for your optimized variational parameters—i.e., the mode of  $q(\theta \mid \gamma)$ .

- (10 pts) Pick 5 random topics, and for each one, report the 10 words with the highest likelihood. Based on these words, can you identify the focus of each topic? What are they?



- (6 pts) Now you get to see the payoff of choosing your own article! Having inferred the distribution over topics  $\theta$  for your specific article, find the two or three topics with the highest likelihood under  $\theta$ , and report the top 10 words for these topics. What might these topics represent? Does this match your description of the article which you gave earlier?

- (7 pts) Using the inferred  $\theta$  for your article, generate a new “document” consisting of 30 words according to the LDA model: for each word, draw a topic  $t$  according to  $\theta$ , and then draw a word according to the multinomial  $\beta_t$ . Paste the generated document below. Can you identify any topics in the “document”? Do they match the themes of your chosen article?

### B.3 Collaboration Policy

After you have completed all other components of this assignment, report your answers to the collaboration policy questions detailed in the Academic Integrity Policies for this course.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details including names of people who helped you and the exact nature of help you received.

2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details including names of people you helped and the exact nature of help you offered.

3. Did you find or come across code that implements any part of this assignment? If so, include full details including the source of the code and how you used it in the assignment.

### References

- D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- P. Dayan. Helmholtz machines and wake-sleep learning. *Handbook of Brain Theory and Neural Network*. MIT Press, Cambridge, MA, 44(0), 2000.
- C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The “wake-sleep” algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. 2006.