HOMEWORK 3
VARIATIONAL INFERENCE[1]

10-708 PROBABILISTIC GRAPHICAL MODELS (SPRING 2022)
https://andrejristeski.github.io/10708-22/

OUT: Mar 14th
DUE: Mar 28th at 11:59 PM
TAs: Che-Ping, Kai-Ling

- **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: https://andrejristeski.github.io/10708-22/#:~:text=Academic%20Integrity%20Policies
- **Late Submission Policy:** See the late submission policy here: https://andrejristeski.github.io/10708-22/#:~:text=Grace%20Day/Late%20Homework%20Policy
- **Submitting your work to Gradescope:** We use Gradescope (https://www.gradescope.com/courses/349316/assignments) 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 personalized feedback explaining your final marks. The homework template must be used and can be completed in Latex or by hand. Handwritten submissions must be legible otherwise we will not be able to give credit to your solutions. No changes should be made to the template, boxes and choices **MUST** remain the same size and in the same locations between the template and your completed submission, the document has 24 pages so your submission must contain no more and no less than 24 pages.
- **Programming Code:** You will also submit your code for the programming questions on the homework to Gradescope, specifically the 'Homework 3 Programming' submission slot. All code written must be submitted in order for you to get any credit for the written components of the programming section.
- 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.
- **Bonus Question:** Question 5 in this homework is a bonus question which you can complete if you would like. The maximum score that you can get on this homework is 100 points, so the bonus question can only make up for points lost in other questions. For example if you get 98 points in the homework and 5 points in the bonus question your final score for the homework will be 100, you will not get the additional 3 points.

[1]Compiled on Tuesday 15th March, 2022 at 21:53

A Written Questions [80 pts] (+20 bonus pts)

1. (Non-concavity of the mean-field approximation) In this exercise, we will see that the objective that results from a mean-field relaxation can yield a non-concave maximization problem. In general, such objectives will have many local maxima and stationary points, thus will not be easy to optimize.

   Consider a simple two variables Ising model with parameter $\theta = [\theta_1, \theta_2, \theta_{12}] \in \mathbb{R}^3$:

   $$p(\mathbf{x}) \propto \exp\left(\theta_1 \mathbf{x}_1 + \theta_1 \mathbf{x}_1 + \theta_{12} \mathbf{x}_1 \mathbf{x}_2\right), \text{ for all } \mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2] \in \{-1, 1\}^2.$$

   Consider the mean-field approximation:

   $$q^* = \underset{q \in \mathcal{Q}}{\operatorname{argmax}}\, H(q) + \mathbb{E}_{\mathbf{x} \sim q}[E_\theta(x)], \tag{A.1}$$

   where $E_\theta(\mathbf{x}) = \theta_1 \mathbf{x}_1 + \theta_2 \mathbf{x}_2 + \theta_{12} \mathbf{x}_1 \mathbf{x}_2$, H is the Shannon entropy, and $\mathcal{Q}$ consists of product distributions over $\{1, -1\}^2$, i.e. its density function can be factorized as $q(\mathbf{x}) = q_1(\mathbf{x}_1) q_2(\mathbf{x}_2)$.

   (a) (10 points) Let $\mu_1 = \mathbb{E}_q[\mathbf{x}_1]$ and $\mu_2 = \mathbb{E}_q[\mathbf{x}_2]$. Show that the objective function (Eqn.(A.1)) can be written as

   $$\max_{\mu_1, \mu_2 \in \mathbb{R}} F_\theta(\mu_1, \mu_2), \text{ with } F_\theta(\mu_1, \mu_2) = \mu_1 \theta_1 + \mu_2 \theta_2 + \mu_1 \mu_2 \theta_{12} + H(q_1) + H(q_2),$$

   where $H(q_i) = -\frac{1}{2}(1 + \mu_i) \log \frac{1}{2}(1 + \mu_i) - \frac{1}{2}(1 - \mu_i) \log \frac{1}{2}(1 - \mu_i)$.

(b) (10 points) Assume that $\theta_1 = \theta_2 = 0$. Prove that there exists some $\theta_{12} \in \mathbb{R}$ such that the objective $F_\theta(\mu_1, \mu_2)$ is not a concave function.

Hint: prove that $F(t, -t)$ is not a concave function for $-1 < t < 1$ and for some θ_{12}.

2. As discussed in lecture, we know that for undirected graphical models, the local polytope constitutes an "outer approximation" of the marginal polytope. We showed the two polytopes are the same when the graph doesn't contain cycles (i.e. is a tree), but for an arbitrary undirected graph G, a valid set of local marginals $\{q_C(\mathbf{x}_C)\}$ isn't always globally valid. That is, for an arbitrary set of local marginals $\{q_C(\mathbf{x}_C)\}$, we cannot always find a distribution $\tilde{q}$ over $\mathbf{X}$, such that for every clique C, $\widetilde{q}_C(\mathbf{x}_C) = q_C(\mathbf{x}_C)$.

   In this question, we're going to see a natural family of additional inequalities that are also satisfied by valid marginals.

   (a) (12 points) Consider a pairwise undirected graphical model with vertices V and edges E. We're going to further assume that every variable $\mathbf{X}_i$ only takes values $\{-1, 1\}$ for simplicity.

   Consider any cycle $C \subseteq E$, that is a set of edges $\{(v_1, v_2), (v_2, v_3), \ldots, (v_{m-1}, v_m), (v_m, v_1)\}$, and any $F \subseteq C$, s.t. $|F|$ is odd. Show that if $\{q_e\}_e$ are valid marginals, that is there is a distribution $\tilde{q}$, s.t. for every edge in the graph, $\widetilde{q}_e(\mathbf{x}_e) = q_e(\mathbf{x}_e)$, the following inequality holds for $\{q_e\}$:

$$\sum_{(\mathbf{X}_i, \mathbf{X}_j) \in C \setminus F} (q_{ij}(\mathbf{x}_i = 1, \mathbf{x}_j = -1) + q_{ij}(\mathbf{x}_i = -1, \mathbf{x}_j = 1)) + \sum_{(\mathbf{X}_i, \mathbf{X}_j) \in F} (q_{ij}(\mathbf{x}_i = 1, \mathbf{x}_j = 1) + q_{ij}(\mathbf{x}_i = -1, \mathbf{x}_j = -1)) \geq 1,$$

   Hint: as the assignment for each variable can be either 1 or -1, think about how many times the assignment can change when we traverse the cycle in the order $(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m, \mathbf{X}_1)$. *The argument made in lecture would also help here.*

(b) (8 points) Unfortunately, even these cycle constraints are not *all* the constraints in the marginal polytope. You'll now see an example where the cycle polytope strictly contains the marginal polytope.

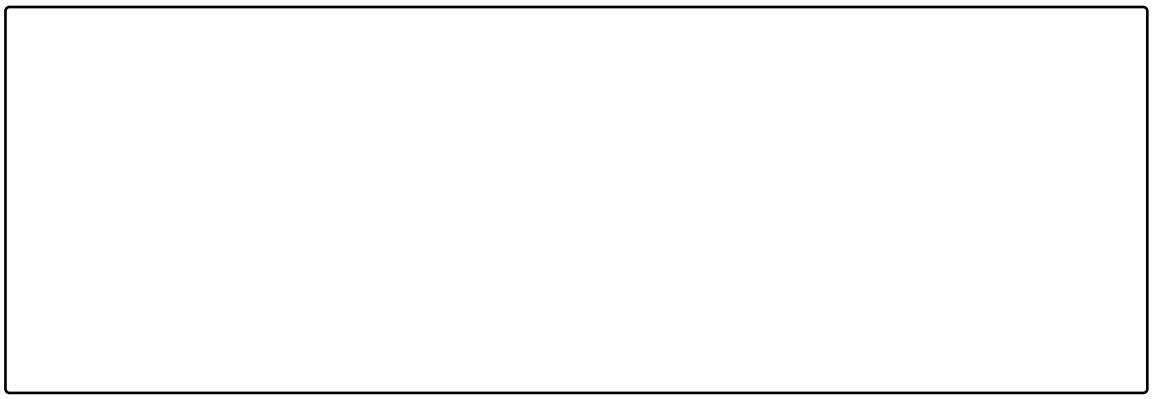
Consider a fully connected graph with 5 variables $\mathbf{X}_1, \ldots, \mathbf{X}_5$, and let the pairwise marginals be

$$q_{ij}(\mathbf{X}_i = 1, \mathbf{X}_j = 1) = q_{ij}(\mathbf{X}_i = -1, \mathbf{X}_j = -1) = \frac{1}{6}$$
$$q_{ij}(\mathbf{X}_i = -1, \mathbf{X}_j = 1) = q_{ij}(\mathbf{X}_i = 1, \mathbf{X}_j = -1) = \frac{1}{3}$$

for all pairs of $\mathbf{X}_i$ and $\mathbf{X}_j$. Show that

1. All the cycle inequalities are satisfied, and
2. there is no distribution $\tilde{q}$ such that for all $\mathbf{x}_i$ and $\mathbf{x}_j$, $\widetilde{q}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = q_{ij}(\mathbf{x}_i, \mathbf{x}_j)$.

Hint: to show that the distribution $\tilde{q}$ doesn't exist, consider maximizing the objective function $\sum_{i,j} \delta(\mathbf{x}_i \neq \mathbf{x}_j)$, compare the solution for valid marginals, and the expected value under q.



3. (Convergence and Non-convergence of the loopy belief propagation algorithm) In this question, we will see loopy belief propagation (LBP) algorithm can sometimes converge, and sometimes not. (Of course, even if it does converge, there's no guarantee it'll converge to the correct marginals.)

   Consider a pairwise undirected graphical model (UGM) defined on a graph $G = (V, E)$, where V denotes a set of random variables with domain $\mathcal{X}$ and E denotes a set of edges. The UGM can be factorized as:

$$p(\mathbf{X} = \mathbf{x}) = \frac{1}{Z_G} \prod_{(i,j)\in E} \Phi_{i,j}(\mathbf{x}_i, \mathbf{x}_j),$$

   where Z_G is the partition function of this UGM. Next, the LBP algorithm for the pairwise UGM can be characterized as following: given number of iterations T, initial messages $m_{j\to i,0}(x)$, $m_{i\to j,0}(x)$ for all $(i, j) \in E$ and for all $x \in \mathcal{X}$.

   procedure LBP(G, T, initial messages):
   - **for** $t \in (1 \dots T)$ **do**
     - **for** $(i, j) \in E$ **do**
       - **for** $x_i \in \mathcal{X}$ **do**
         - $m_{j\to i,t}(x_i) \leftarrow \sum_{x_j\in\mathcal{X}} \Phi_{i,j}(x_i, x_j) \prod_{k\in N(j)\setminus i} m_{k\to j,t-1}(x_j)$.
       - Normalize $m_{j\to i,t}(x)$ for all $x \in \mathcal{X}$ so that $\sum_{x\in\mathcal{X}} m_{j\to i,t}(x) = 1$.
       - **for** $x_j \in \mathcal{X}$ **do**
         - $m_{i\to j,t}(x_j) \leftarrow \sum_{x_i\in\mathcal{X}} \Phi_{i,j}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i,t-1}(x_i)$.
       - Normalize $m_{i\to j,t}(x)$ for all $x \in \mathcal{X}$ so that $\sum_{x\in\mathcal{X}} m_{i\to j,t}(x) = 1$.
     - Return $m_{j\to i,T}$ and $m_{i\to j,T}$ for all $(i, j) \in E$

   Ideally, we set the number of iterations T large enough so that all messages converge to some fixed point. After they converge, we can calculate the marginals using the following formula:

$$\hat{P}_T(X_i = x_i) \propto \prod_{k\in N(i)} m_{k\to i,T}(x_i).$$

   Now assume that we aim to use LBP to calculate the marginal distribution. Consider an undirected graphical model $G = (V, E)$, where the node set $V = \{1, 2, \cdots, n\}$ represents n binary random variables $X_1, \cdots, X_n \in \{0, 1\}$ and the edge set is $E = \{(1, 2), \cdots, (n-1, n), (n, 1)\}$. Therefore, these variables form a single cycle consisting of edges between i and $i + 1$ for all $1 \le i \le n - 1$ and between 1 and n. The potential functions for each edge is

$$\Phi_{i,j}(X_i = x_i, X_j = x_j) = \begin{cases} 1 & \text{, if } x_i = x_j. \\ \epsilon & \text{, otherwise.} \end{cases} \text{ for all } (i, j) \in E.$$

(a) (4 points) **Numerical Answer:** Assume that $0 \leq \epsilon < 1$. What is the true marginal probabilities $P(X_1 = 0)$ and $P(X_1 = 1)$?

(b) (8 points) Assume that $0 < \epsilon < 1$ and $[m_{j\to i,0}(0), m_{j\to i,0}(1)] = [\pi_0, 1-\pi_0]$ with $0 \leq \pi_0 \leq 1$ for all $(i,j) \in E$ and $(j,i) \in E$. Prove that $\hat{P}_T(X_1 = x_1)$ converges to the true marginal probability, i.e.

$$\lim_{T\to\infty} \hat{P}_T(X_1 = x_1) = P(X_1 = x_1).$$

Hint: derive the following recursive formulas for messages: show there exist positive constants $C_1, C_2 < 1$ *such that*

$$m_{i\to j,t}(x_i) - C_1 = C_2\left(m_{i\to j,t-1}(x_i) - C_1\right), \text{ for } x_i \in \{0,1\}, \text{ and for all } (i,j) \in E, (j,i) \in E.$$

(c) (8 points) Assume that $\epsilon = 0$ and initial messages are positive and normalized, i.e.

$$m_{j\to i,0}(0) + m_{j\to i,0}(1) = 1 \text{ for all } (i,j),(j,i) \in E.$$

Prove that if the initial messages for all edges are not identical, i.e. there exist $(i_1, j_1), (i_2, j_2) \in E$ satisfying $m_{j_1\to i_1,0}(0) \neq m_{j_2\to i_2,0}(0)$ and $m_{i_1\to j_1,0}(0) \neq m_{i_2\to j_2,0}(0)$, the messages do not converge.

4. (Bayesian Linear Regression) In this question, we're going to see a Bayesian version of linear regression, and how we can estimate the posterior distribution of $\mathbf{w}$ (which is analogous to finding the optimal weights in normal linear regression) by optimizing a mean-field approximation.

   For a set of features $\mathbf{X} \in \mathbb{R}^{N \times M}$, responses $\mathbf{y} \in \mathbb{R}^{N \times 1}$ and the weights $\mathbf{w} \in \mathbb{R}^{M \times 1}$, we assume they follow the distributions:

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) &= \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i \mid \mathbf{X}_i \mathbf{w}, \lambda^{-1}), \\ p(\mathbf{w} \mid \boldsymbol{\tau}) &= \mathcal{N}(\mathbf{w} \mid 0, \boldsymbol{\tau}^{-1}\mathbf{I}), \\ p(\boldsymbol{\tau}) &= \text{Gamma}(\boldsymbol{\tau} \mid \alpha_0, \beta_0), \end{aligned}$$

   where $p(\mathbf{w} \mid \boldsymbol{\tau})$ is the prior distribution of $\mathbf{w}$. Also assume that λ, α_0 and β_0 are given and fixed.
   As shown in A.1, the joint distribution of $\mathbf{y}$, $\mathbf{w}$ and $\boldsymbol{\tau}$ over the features $\mathbf{X}$ is given by the factorization:

$$p(\mathbf{y}, \mathbf{w}, \boldsymbol{\tau} \mid \mathbf{X}) = p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) p(\mathbf{w} \mid \boldsymbol{\tau}) p(\boldsymbol{\tau}).$$

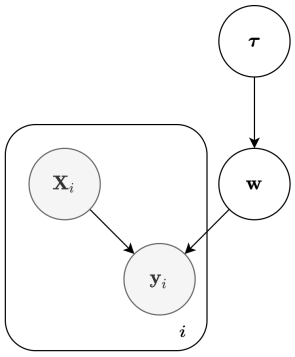


Figure A.1: Bayesian Linear Regression

   In order to perform inference on new data, we need to approximate the posterior distribution $p(\mathbf{w}, \boldsymbol{\tau} \mid \mathbf{y}, \mathbf{X})$. We will use in this problem a mean-field approximation to do so: that is, we will find a distribution q^*, which follows the factorization

$$q^*(\mathbf{w}, \boldsymbol{\tau}) = q^*(\mathbf{w}) q^*(\boldsymbol{\tau}),$$

   and maximizes the variational formulation:

$$q^*(\mathbf{w}, \boldsymbol{\tau}) = \text{argmax}_{q \in \mathcal{Q}} [H(q(\mathbf{w}, \boldsymbol{\tau})) + \mathbb{E}_q[p(\mathbf{y}, \mathbf{w}, \boldsymbol{\tau} \mid \mathbf{X})]].$$

   We will further assume that $q(\mathbf{w})$ is a Gaussian distribution, i.e. $q(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mu, \Sigma)$ and $q(\boldsymbol{\tau})$ is a Gamma distribution, i.e. $q(\boldsymbol{\tau}) = \text{Gamma}(\tau | \alpha, \beta)$.

   Suppose we are using block CAVI (coordinate ascent variational inference) to find the optimal parameters for the Gaussian and Gamma distribution, updating the "block" of coordinates $\mathbf{w}$ and τ alternatingly:

$$\begin{aligned} q(\mathbf{w}) &\propto \exp\left(\mathbb{E}_{\boldsymbol{\tau}}[\ln p(\mathbf{y}, \mathbf{w}, \boldsymbol{\tau} \mid \mathbf{X})\right) \\ q(\boldsymbol{\tau}) &\propto \exp\left(\mathbb{E}_{\mathbf{w}}[\ln p(\mathbf{y}, \mathbf{w}, \boldsymbol{\tau} \mid \mathbf{X})\right). \end{aligned}$$

(a) (8 points) Derive the update formula for the parameters of $q(\mathbf{w})$.

(b) (8 points) Derive the update formula for the parameters of $q(\boldsymbol{\tau})$.

(c) (4 points) After convergence, we can perform inference on new features $\mathbf{x}'$ by evaluating the predictive distribution:

$$\int_{\mathbf{w}} p(\mathbf{y}' \mid \mathbf{x}', \mathbf{w}) q^*(\mathbf{w}) d\mathbf{w}.$$

Derive the exact expression for the predictive distribution, assuming $q^*(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mu^*, \Sigma^*)$.

5. **(Bonus question)** In class, we saw that when the graph is a tree, the outer relaxation approach to approximating the partition function is exact. In this question, we'll see that a similar approach works when the graph has bounded treewidth.

   For any undirected graph G, we can define the notion of a *treewidth* — which often governs the efficiency of various graph algorithms. Towards this, a *tree decomposition* of $G = (V, E)$ is a tree T, with nodes $W_1, W_2 \ldots, W_n$, s.t. each V_i is a subset of the nodes of G, with the following additional properties:

   - $\cup_i W_i = V$, that is the union of all sets W_i is the set of all nodes of G.
   - For every edge $(u, v) \in E$, there is at least one set W_i that contains both u, v.
   - If two sets W_i, W_j contain node v, then all W_k on the unique path between W_i, W_j in the tree T also contain v. Equivalently, if W_k is on the path from W_i to W_j, $W_i \cap W_j \subseteq W_k$.

   An example of such a decomposition can be seen on Figure A.2.

   This decomposition is not unique — e.g. having a single set W with all the nodes in it is a valid tree decomposition. The *width* of a particular tree decomposition is the size of the largest set W_i in this decomposition minus one. The *treewidth* of a graph is the smallest width of any valid tree decomposition.

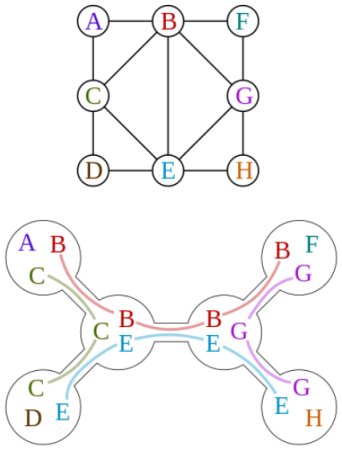


Figure A.2: Tree decomposition

Now, consider a pairwise graphical model $p(\mathbf{x}) \propto \exp\left(\sum_{(i,j)\in E(G)} \phi_{i,j}(\mathbf{x}_i, \mathbf{x}_j)\right)$ for a graph G with treewidth bounded by k, with a corresponding tree decomposition T. For simplicity, let the domain of each variable be $\{\pm 1\}$, i.e. let $\mathbf{x} \in \{\pm 1\}^{|V|}$.

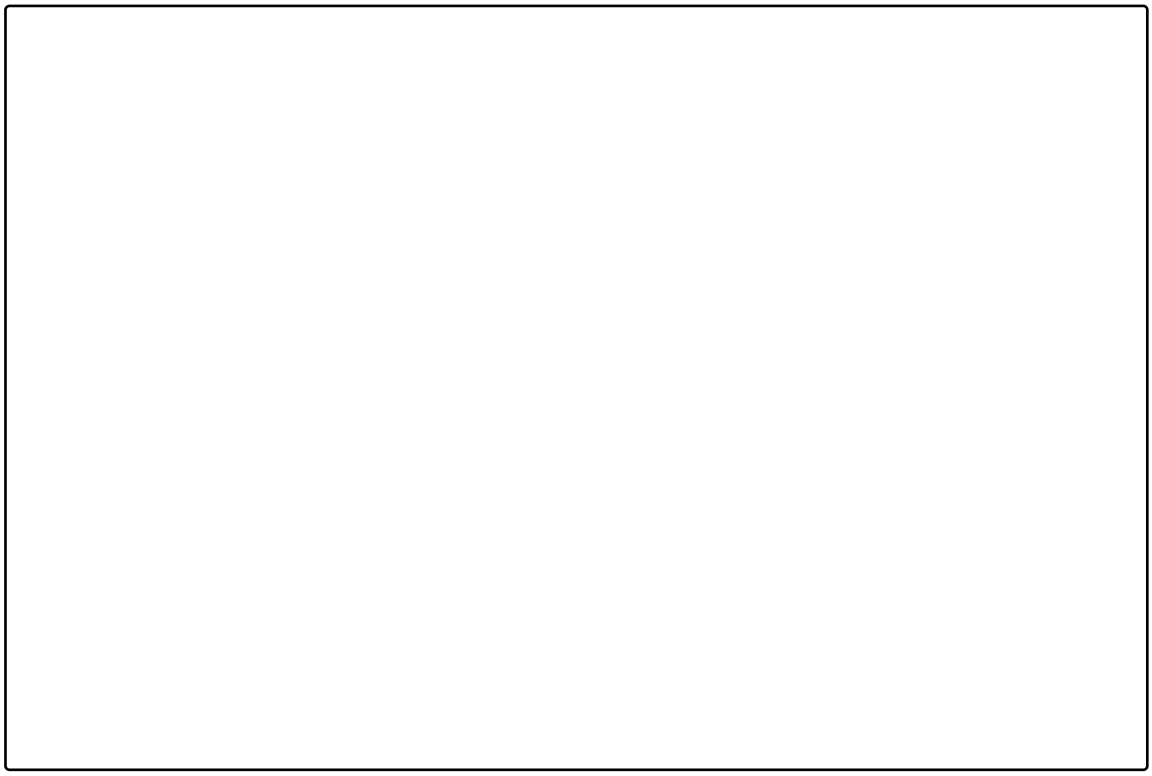
(a) (8 points) Root the tree T at W_1 and assume $W_1, W_2, \ldots, W_n$ are topologically sorted according to this rooting. Consider any distribution $\tilde{q}$ over $|V(G)|$ variables. Show that the Shannon entropy of $\tilde{q}$ satisfies

$$H(\tilde{q}) \leq H(\tilde{q}(W_1)) + \sum_{i=2}^{n} H(\tilde{q}(W_i | W_{\text{parent}(W_i)}))$$

Moreover, show that if $\tilde{q}$ corresponds to a distribution describable by a pairwise graphical model

with a tree decomposition T,

$$H(\tilde{q}) = H(\tilde{q}(W_1)) + \sum_{i=2}^{n} H(\tilde{q}(W_i|W_{\text{parent}(W_i)}))$$



(b) (12 points) Consider the local polytope L_k consisting of all valid marginals over subset of nodes of up to $2k$ variables. That is for every $S \subset \{1, 2, \ldots, |V|\}$, s.t. $|S| \leq 2k$ and every $x_S \in \{\pm 1\}^{|S|}$ we have a variable $q_S(\mathbf{x}_S)$, s.t. these variables satisfy the constraints:

- $\forall S, x_{\mathbf{S}}$: $q_S(\mathbf{x}_S) \geq 0$. (Non-negativity.)
- $\forall S$: $\sum_{\mathbf{x_S}} q_S(\mathbf{x}_S) = 1$. (Marginals are valid distributions.)
- $\forall S, S', \mathbf{x}_S$, s.t. $:S' \cap S = \emptyset; |S \cup S'| \leq 2k$: $\sum_{\mathbf{x}_{S'}} q_{S\cup S'}(\mathbf{x}_{S\cup S'}) = q_S(\mathbf{x}_S)$. (Marginals are consistent.)

Moreover, for set of local marginals $\{q_S(\mathbf{x}_S)\} \in L_k$, define, as in part (a),

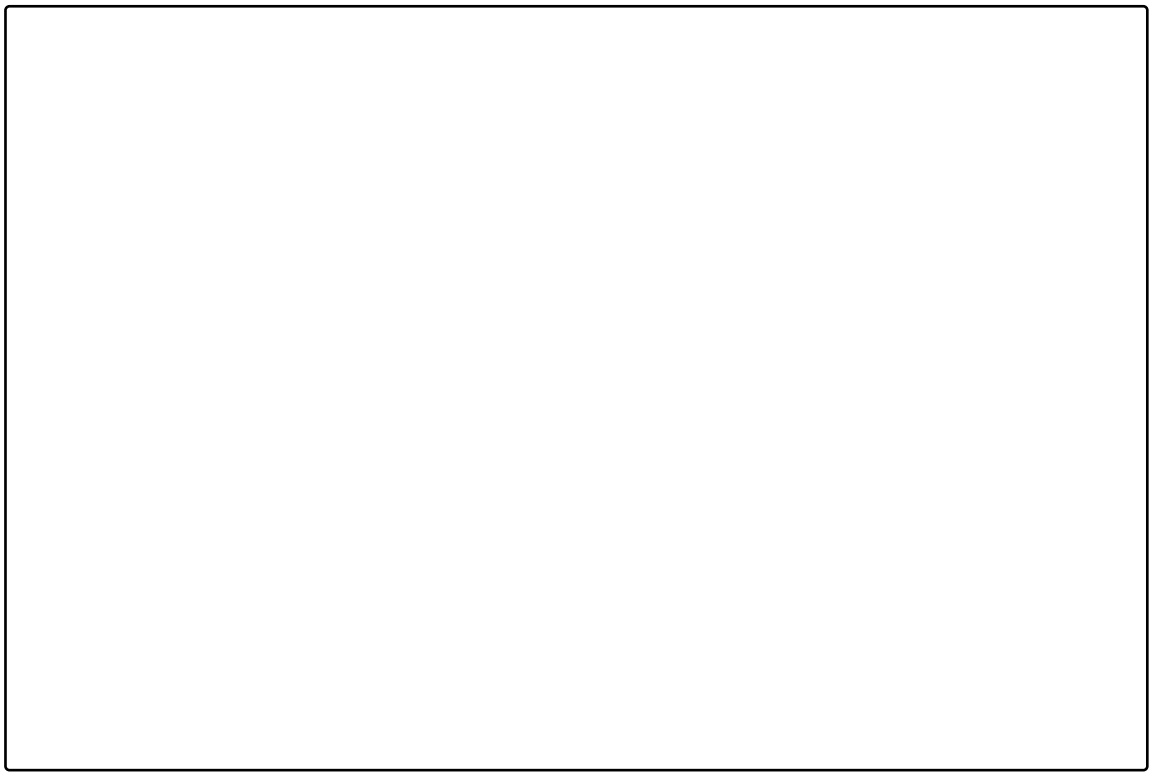
$$H_{\text{tw}}(q) := H(q(W_1)) + H\sum_{i=2}^{n} H(q(W_i|W_{\text{parent}(W_i)}))$$

Prove that

$$\log Z = \max_{q_S(\mathbf{x}_S)\in L_k} \left\{ \sum_{(i,j)\in E(G)} \mathbb{E}_{q_{i,j}} \phi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) + H_{\text{tw}}(q) \right\}$$

where Z is the partition function of p. In other words, prove that the local relaxation in which we include the local constraints for sets of size up to $2k$ is exact.
Hint: Imitate the proof from class. First prove that the RHS is a relaxation, that is the RHS is bigger

than or equal to $\log Z$*. Then, prove that given any* $q_S(\mathbf{x}_S) \in L_k$*, one can produce a distribution* q*, s.t. the value of the objective is exactly preserved. For this, follow the proof from class for trees.*



B Programming [20 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. Then, we'll ask you to do some exploration with the model you've learned.

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.

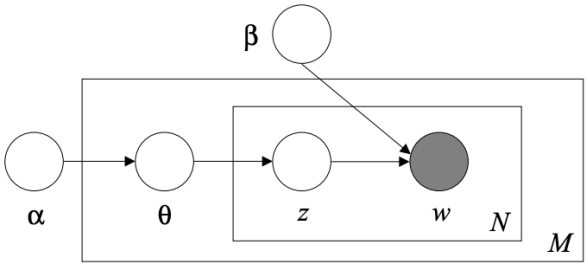


Figure B.1: LDA Generative Model

Here 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!

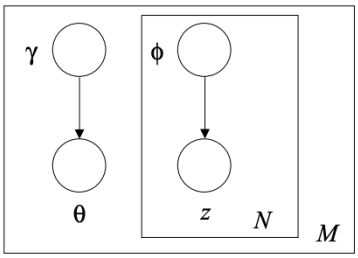


Figure B.2: Graphical Representation of the Variational Distribution

Figure B.2 depicts our chosen variational distribution which we will use to approximate the intractable posterior. Observe that we have dropped the prior α over topic distributions which is shared by all documents;

instead, our approximation assumes that each document's topic distribution θ is drawn as a function of the variational parameter γ. Likewise, each word has its own variational distribution over topics.

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

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

Recall that standard variational inference 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,

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

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)).

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

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni}. \tag{B.4}$$

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

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

where Ψ 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 ϕ 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) α, β, 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 α, β. Once we have learned these parameters, our next step is to find the choice of α, β 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:

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

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^{th} word of the d^{th} document is the j^{th} vocabulary word (recall that β is a matrix such that β_i parameterizes a multinomial distribution over words for topic i). Additionally, the update to α is given as

$$\alpha^{+} = \alpha + \frac{g - c}{h}, \tag{B.7}$$

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 α, respectively. The z is a scalar that also shows up in the Hessian. They can be computed as

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

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

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

Note the use of Ψ', not Ψ! 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 β.
- To check convergence for the E-step, you can use the following criterion:

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

- To check convergence of α^{+} 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

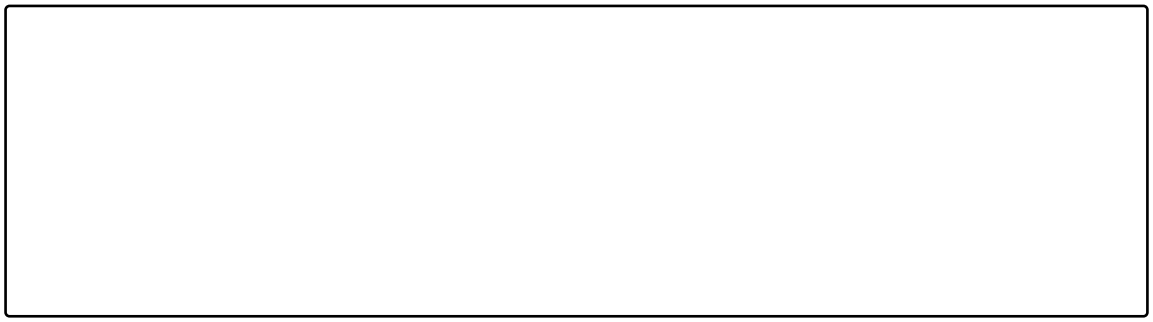
In this assignment, we will be using the above variational EM algorithm to learn the parameters of an LDA model for a corpus of articles by the New York Times. You'll also get to do this for an article of your choosing!

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.

(a) (2 points) 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`.

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 α, the Dirichlet prior over document topics. The second is β, which parameterizes each topic's multinomial distribution over words. The last one is θ, 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 $q(\theta \mid \gamma)$.

(b) (8 points) 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?

(c) (4 points) Now you get to see the payoff of choosing your own article! Having inferred the distribution over topics θ for your specific article, find the two or three topics with the highest likelihood under θ, 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?

(d) (6 points) Using the inferred θ 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 θ, and then draw a word according to the multinomial β_t. Paste the generated document below. Can you identify any topics in the "document"? Do they match the themes of your chosen article?

(e) (0 points) Further reading: there also exists deep-learning counterpart of LDA called neural topic models, which are essentially VAEs with different priors. If you would like to learn more about this type of models, Srivastava and Sutton (2017), Miao et al. (2017) and Dieng et al. (2020) are some works you can start with; there are also variants using the trending pre-trained LM (Bianchi et al. (2020a), Bianchi et al. (2020b)), or with additional modules to capture the topics varying over time (Dieng et al. (2019)). One can refer to the recent survey paper (Zhao et al. (2021)) for more details.

C 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

F. Bianchi, S. Terragni, and D. Hovy. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. *arXiv preprint arXiv:2004.03974*, 2020a.

F. Bianchi, S. Terragni, D. Hovy, D. Nozza, and E. Fersini. Cross-lingual contextualized topic models with zero-shot learning. *arXiv preprint arXiv:2004.07737*, 2020b.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

A. B. Dieng, F. J. Ruiz, and D. M. Blei. The dynamic embedded topic model. *arXiv preprint arXiv:1907.05545*, 2019.

A. B. Dieng, F. J. Ruiz, and D. M. Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 2020.

Y. Miao, E. Grefenstette, and P. Blunsom. Discovering discrete latent topics with neural variational inference. In *International Conference on Machine Learning*, pages 2410–2419. PMLR, 2017.

A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du, and W. Buntine. Topic modelling meets deep neural networks: A survey. *arXiv preprint arXiv:2103.00498*, 2021.