HOMEWORK 5
VARIATIONAL INFERENCE[1]

10-418 / 10-618 MACHINE LEARNING FOR STRUCTURED DATA (FALL 2019)
https://piazza.com/cmu/fall2019/1041810618

OUT: Nov. 20, 2019
DUE: Dec 02, 2019 11:59 PM
TAs: Aakanksha, Austin, Karthika

START HERE: Instructions

Summary In this assignment, we will walk through the basics of Variational Inference, Mean-Field Approximation and the Coordinate Ascent Variational Inference (CAVI) algorithm for simple distributions such as multivariate Gaussians and Gaussian Mixture Models. Finally, we will wrap up with a brief comparison between variational methods and sampling-based methods such as MCMC.

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 2.1"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only. See the Academic Integrity Section on the course site for more information: http://www.cs.cmu.edu/~mgormley/courses/10418/about.html#7-academic-integrity-policies
- **Late Submission Policy:** See the late submission policy here: http://www.cs.cmu.edu/~mgormley/courses/10418/about.html#6-general-policies
- **Autolab:** You will submit your code for programming questions on the homework to Autolab (https://autolab.andrew.cmu.edu/). After uploading your code, we will manually grade your code by hand. We will not use Autolab to autograde your code.
- **Submitting your work to Gradescope:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, we will be using Gradescope (https://gradescope.com/). Please use the provided template. Submissions can be handwritten, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Alternatively, submissions can be written in LaTeX. Regrade requests can be made, however this gives the TA the opportunity to regrade your entire paper, meaning if additional mistakes are found then points will be deducted. For short answer questions you **should not** include your work in your solution. If you include your work in your solutions, your assignment may not be graded correctly by our AI assisted grader.

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.

[1] Compiled on Tuesday 3rd December, 2019 at 04:11

1 Written Questions [44 pts]

Answer the following questions in the template provided. Then upload your solutions to Gradescope. You may use LaTeX or print the template and hand-write your answers then scan it in. Failure to use the template may result in a penalty. There are 44 points and 19 questions.

1.1 Mean-Field Approximation for Multivariate Gaussians

In this question, we'll explore how accurate a Mean-Field approximation can be for an underlying multivariate Gaussian distribution.

Assume we have observed data $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^n$ that was drawn from a 2-dimensional Gaussian distribution $p(\mathbf{x}; \boldsymbol{\mu},\ \boldsymbol{\Lambda}^{-1})$.

$$p(\mathbf{x}; \boldsymbol{\mu},\ \boldsymbol{\Lambda}) = \mathcal{N}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}^{-1} \right) \tag{1.1}$$

Note here that we're using the *precision* matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$. An additional property of the precision matrix is that it is symmetric, so $\Lambda_{12} = \Lambda_{21}$. This will make your lives easier for the math to come.

We will approximate this 2-dimensional Gaussian with a mean field approximation, $q(\mathbf{x}) = q(\mathbf{x}_1)q(\mathbf{x}_2)$, the product of two 1-dimensional distributions $q(\mathbf{x}_1)$ and $q(\mathbf{x}_2)$. For now, we won't assume any form for this distributions.

1. (1 point) **Short Answer:** Write down the equation for $\log p(\mathbf{X})$. For now, you can leave all of the parameters in terms of vectors and matrices, not their subcomponents.

$$\log p(\mathbf{X}) = -\frac{n}{2}\log\left((2\pi)^2|\boldsymbol{\Lambda}^{-1}|\right) - \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - \boldsymbol{\mu})^T\boldsymbol{\Lambda}(\mathbf{x}^{(i)} - \boldsymbol{\mu})$$

2. (2 points) **Short Answer:** Group together everything that involves $\mathbf{X}_1$ and remove anything involving $\mathbf{X}_2$. We claim that there exists some distribution $q^*(\mathbf{X}) = q^*(\mathbf{X}_1)q^*(\mathbf{X}_2)$ that minimizes the KL divergence $q^* = \text{argmin}_q \text{KL}(q||p)$. And further, said distribution will have a component $q^\star(\mathbf{X}_1)$ will be proportional to the quantity you find below.

Let $\mathbf{d}^{(i)} = \mathbf{x}^{(i)} - \boldsymbol{\mu}$

$$\exp\left(-\frac{1}{2}\sum_{i=1}^{n}\mathbf{d}^{(i),T}\boldsymbol{\Lambda}\mathbf{d}^{(i)}\right) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{n}d_1^{(i),2}\Lambda_{11} + 2d_1^{(i)}d_2^{(i)}\Lambda_{12}\right) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{n}d_1^{(i),2}\Lambda_{11} + 2d_1^{(i)}d_2^{(i)}\Lambda_{12} + d_2^{(i),2}\frac{\Lambda_{12}^2}{\Lambda_{11}}\right)$$
$$= \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\left(d_1^{(i)} + d_2^{(i)}\frac{\Lambda_{12}}{\Lambda_{11}}\right)^2\Lambda_{11}\right) = \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\left(x_1^{(i)} - (\mu_1 - \Lambda_{12}\Lambda_{11}^{-1}(x_2^{(i)} - \mu_2))\right)^2\Lambda_{11}\right)$$

It can be shown that this implies that $q(\mathbf{X}_1)$ (and therefore $q(\mathbf{X}_2)$) is a Gaussian distribution.

$$q(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1; m_1, \Lambda_{11}^{-1})$$

Where $m_1 = \mu_1 - \Lambda_{11}^{-1}\Lambda_{12}(\mathbb{E}[x_2] - \mu_2)$

Using these facts, we'd like to explore how well our approximation can model the underlying distribution.

3. Suppose the parameters of the true distribution are $\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\boldsymbol{\Lambda} = \begin{pmatrix} 1 & 0 \\ 0 & 1/4 \end{pmatrix}$.

(a) (1 point) **Numerical Answer:** What is the value of the mean of the Gaussian for $q^*(\mathbf{X}_1)$?

0

(b) (1 point) **Numerical Answer:** What is the value of the variance of the Gaussian for $q^*(\mathbf{X}_1)$?

1

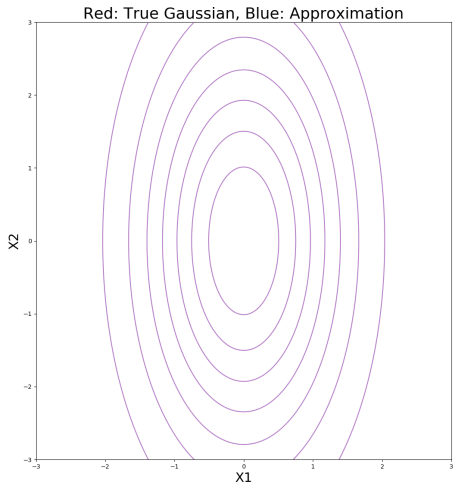
(c) (1 point) **Numerical Answer:** What is the value of the mean of the Gaussian for $q^*(\mathbf{X}_2)$?

0

(d) (1 point) **Numerical Answer:** What is the value of the variance of the Gaussian for $q^*(\mathbf{X}_2)$?

4

(e) (2 points) **Plot:** Provide a *computer-generated* contour plot to show the result of our approximation $q^*(\mathbf{X})$ and the true underlying Gaussian $p(\mathbf{X}; \boldsymbol{\mu}, \boldsymbol{\Lambda})$ for the parameters given above.



4. Suppose the parameters of the true distribution are $\boldsymbol{\mu} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $\boldsymbol{\Lambda} = \begin{pmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{pmatrix}$.

(a) (1 point) **Numerical Answer:** What is the value of the mean of the Gaussian for $q^*(\mathbf{X}_1)$?

1

(b) (1 point) **Numerical Answer:** What is the value of the variance of the Gaussian for $q^*(\mathbf{X}_1)$?

2

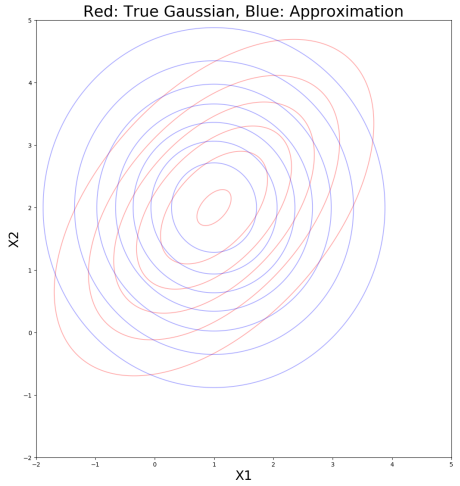
(c) (1 point) **Numerical Answer:** What is the value of the mean of the Gaussian for $q^*(\mathbf{X}_2)$?

2

(d) (1 point) **Numerical Answer:** What is the value of the variance of the Gaussian for $q^*(\mathbf{X}_2)$?

2

(e) (2 points) **Plot:** Provide a *computer-generated* contour plot to show the result of our approximation $q^*(\mathbf{X})$ and the true underlying Gaussian $p(\mathbf{X}; \boldsymbol{\mu}, \boldsymbol{\Lambda})$ for the parameters given above.



5. (1 point) Describe in words how the plots you generated provide insight into the behavior of minimization of $KL(q||p)$ with regards to the low probability and high probability regions of the the true vs. approximate distributions.

In minimizing $KL(q||p)$, we can see that the approximate distribution tries to cover as much of the high probability regions of the original distribution as it can, even if that means having relatively higher density in areas where the original distribution had low density.
In the case of the first distribution with diagonal precision matrix, the approximated distribution is exactly the same as a multivariate normal distribution with diagonal covariance is just a product of univariate normals, which is our approximation.

1.2 Variational Inference for Gaussian Mixture Models

Now that we have seen how the mean-field approximation works for a multivariate Gaussian, let's look at the case of Gaussian Mixture Models. Suppose we have a Bayesian mixture of unit-variance univariate Gaussian distributions. This mixture consists of 2 components each corresponding to a Gaussian distribution, with means $\boldsymbol{\mu} = \{\mu_1, \mu_2\}$. The mean parameters are drawn independently from a Gaussian prior distribution $\mathcal{N}(0, \sigma^2)$. The prior variance σ^2 is a hyperparameter. Generating an observation x_i from this model is done

according to the following generative story:

1. Choose a cluster assignment c_i for the observation. The cluster assignment is chosen from the distribution Categorical($\frac{1}{2}, \frac{1}{2}$) and indicates which latent cluster x_i comes from. Encode c_i as a one-hot vector where $[1, 0]$ indicates that x_i is assigned to cluster 0 and vice versa.
2. Generate x_i from the corresponding Gaussian distribution $\mathcal{N}(c_i^T \boldsymbol{\mu}, 1)$

The complete hierarchical model is as follows:

$$\begin{aligned} \mu_k &\sim \mathcal{N}(0, \sigma^2), k \in \{1, 2\} \\ c_i &\sim \text{Categorical}(\frac{1}{2}, \frac{1}{2}), i \in [1, n] \\ x_i | c_i, \boldsymbol{\mu} &\sim \mathcal{N}(c_i^T \boldsymbol{\mu}, 1), i \in [1, n] \end{aligned}$$

where n is the number of observations generated from the model.

1. (1 point) What are the observed and latent variables for this model?

   The observed variables are x_i.
   The latent variables are $c_i, \boldsymbol{\mu}$

2. (1 point) Write down the joint probability of observed and latent variables under this model

   $$p(\mathbf{x}, \mathbf{c}, \boldsymbol{\mu}) = p(\mathbf{x} \mid \mathbf{c}, \boldsymbol{\mu}) p(\mathbf{c}) p(\boldsymbol{\mu})$$

3. (3 points) Let's calculate the ELBO (evidence lower-bound) for this model. Recall that the ELBO is given by the following equation:

   $$\text{ELBO}(q) = \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_q[\log q(\mathbf{z})]$$

   To calculate $q(\mathbf{z})$, we will now use the mean-field assumption. Under this assumption, each latent variable is governed by its own latent factor, resulting in the following probability distribution:

   $$q(\boldsymbol{\mu}, \mathbf{c}) = \left(\prod_{k=1}^{2} q(\mu_k; m_k, v_k^2) \right) \left(\prod_{i=1}^{n} q(c_i; a_i) \right)$$

   Here $q(\mu_k; m_k, v_k^2)$ is the Gaussian distribution for the k-th mixture component with mean and variance m_k and v_k^2. $q(c_i; a_i)$ is the categorical distribution for the i-th observation with assignment probabilities a_i (a_i is a 2-dimensional vector). Given this assumption, write down the ELBO as a function of the variational parameters $\mathbf{m}, \mathbf{v}^2, \mathbf{a}$.

$$\begin{aligned}\text{ELBO}(q) = &\mathbb{E}[\log p(\mathbf{x} \mid \mathbf{c}, \boldsymbol{\mu}) \mid \mathbf{a}, \mathbf{m}, \mathbf{v}^2] + \mathbb{E}[\log p(\mathbf{c}) \mid \mathbf{a}] + \mathbb{E}[\log p(\boldsymbol{\mu}) \mid \mathbf{m}, \mathbf{v}^2] \\ &- \mathbb{E}_q[\log q(\boldsymbol{\mu} \mid \mathbf{m}, \mathbf{v}^2) + \log q(\mathbf{c} \mid \mathbf{a})]\end{aligned}$$

4. Now that we have the ELBO formulation, let's try to compute coordinate updates for our latent variables. Remember that the optimal variational density of a latent variable z_i is proportional to the exponentiated expected log of the complete conditional given all other latent variables in the model and the observed data. In other words:

$$q_i(z_i) \propto \exp\left(\mathbb{E}_{-j}[\log p(z_j|\mathbf{z}_{-j}, \mathbf{x})]\right)$$

Equivalently, you can also say that the variational density is proportional to the exponentiated expected log of the joint $\mathbb{E}_{-j}[\log p(z_j, \mathbf{z}_{-j}, \mathbf{x})]$. This is a valid coordinate update since the expectations on the right side of the equation do not involve z_j due to the mean-field assumption.

(a) (4 points) Show that the variational update for $a_{i1} \propto \exp\left(\mathbb{E}[\mu_1; m_1, s_1^2]x_i - \frac{\mathbb{E}[\mu_1^2; m_1, s_1^2]}{2}\right)$.

(**Hint:** We can write the optimal variational density for cluster assignment variables as $q(c_i; a_{i1}) \propto \exp\left(\log p(c_i) + \mathbb{E}_{\boldsymbol{\mu}}[\log p(x_i|c_i, \boldsymbol{\mu}); \mathbf{m}, \mathbf{v}^2]\right)$. Feel free to drop added constants along the way.)

$$\log p(c_i) = \log \frac{1}{2} = -\log 2$$

$$\begin{aligned}\log p(x_i \mid c_i, \boldsymbol{\mu}) &= \log\left(\prod_{k=1}^{2} p(x_i \mid \mu_k)^{c_{ik}}\right) = \sum_{k=1}^{2} c_{ik} \log p(x_i \mid \mu_k) \\ &= \sum_{k=1}^{2} c_{ik}\left(C - \frac{(x_i - \mu_k)^2}{2}\right) = C - \sum_{k=1}^{2} c_{ik}\frac{(x_i - \mu_k)^2}{2} \\ &= C - \sum_{k=1}^{2} c_{ik}\frac{x_i^2 - 2\mu_k x_i + \mu_k^2}{2} = C - \frac{x_i^2}{2} + \sum_{k=1}^{2} c_{ik}\left(\mu_k x_i - \frac{\mu_k^2}{2}\right)\end{aligned}$$

where C is a constant from the log pdf of the normal distribution.

$$\begin{aligned}q(c_i \mid a_{i1}) &\propto \exp\left(-\log 2 + \mathbb{E}_{\boldsymbol{\mu}}\left[C - \frac{x_i^2}{2} + \sum_{k=1}^{2} c_{ik}\left(\mu_k x_i - \frac{\mu_k^2}{2}\right) \mid \mathbf{m}, \mathbf{v}^2\right]\right) \\ &\propto \exp\left(\sum_{k=1}^{2} c_{ik}\left(\mathbb{E}[\mu_k \mid m_k, v_k^2]x_i - \frac{\mathbb{E}[\mu_k^2 \mid m_k, v_k^2]}{2}\right)\right) \\ &= \prod_{k=1}^{2}\left(\exp\left(\mathbb{E}[\mu_k \mid m_k, v_k^2]x_i - \frac{\mathbb{E}[\mu_k^2 \mid m_k, v_k^2]}{2}\right)\right)^{c_{ik}} \\ \implies a_{i1} &\propto \exp\left(\mathbb{E}[\mu_1 \mid m_1, v_1^2]x_i - \frac{\mathbb{E}[\mu_1^2 \mid m_1, v_1^2]}{2}\right)\end{aligned}$$

(b) (6 points) Show that the variational updates for the k-th mixture component are $m_k = \frac{\sum_i a_{ik} x_i}{1/\sigma^2 + \sum_i a_{ik}}$ and $v_k^2 = \frac{1}{1/\sigma^2 + \sum_i a_{ik}}$.
(**Hint:** We can write the optimal variational density for the k-th mixture component as $q(\mu_k) \propto \exp\left(\log p(\mu_k) + \sum_{i=1}^n \mathbb{E}_{c_i}[\log p(x_i | c_i, \boldsymbol{\mu}); a_i, \mathbf{m}_{-k}, \mathbf{v}_{-k}^2] \right)$. Feel free to drop added constants along the way.)

$$
\begin{aligned}
q(\mu_k) &\propto \exp\left(\log p(\mu_k) + \sum_{i=1}^n \mathbb{E}_{c_i}[\log p(x_i \mid c_i, \boldsymbol{\mu}) \mid a_i, \mathbf{m}_{-k}, \mathbf{v}_{-k}^2] \right) \\
&\propto \exp\left(-\frac{\mu_k^2}{2\sigma^2} + \sum_{i=1}^n \mathbb{E}_{c_i}[c_{ik} \log p(x_i \mid \mu_k) \mid a_i] \right) \\
&\propto \exp\left(-\frac{\mu_k^2}{2\sigma^2} + \sum_{i=1}^n \mathbb{E}_{c_i}[c_{ik} \mid a_i] \left(\mu_k x_i - \frac{\mu_k^2}{2} \right) \right) \\
&= \exp\left(-\frac{\mu_k^2}{2\sigma^2} + \sum_{i=1}^n a_{ik} \left(\mu_k x_i - \frac{\mu_k^2}{2} \right) \right) \\
&= \exp\left(\left(\sum_{i=1}^n a_{ik} x_i \right) \mu_k - \frac{1}{2} \left(\frac{1}{\sigma^2} + \sum_{i=1}^n a_{ik} \right) \mu_k^2 \right) \\
&\propto \exp\left(-\frac{1}{2\left(\frac{1}{\sigma^2} + \sum_{i=1}^n a_{ik} \right)} \left(\mu_k - \left(\frac{\sum_{i=1}^n a_{ik} x_i}{\frac{1}{\sigma^2} + \sum_{i=1}^n a_{ik}} \right) \right)^2 \right) \\
\implies m_k &= \frac{\sum_{i=1}^n a_{ik} x_i}{\frac{1}{\sigma^2} + \sum_{i=1}^n a_{ik}}, \\
v_k^2 &= \left(\frac{1}{\sigma^2} + \sum_{i=1}^n a_{ik} \right)
\end{aligned}
$$

1.3 Running CAVI: Toy Example

Let's now see this in action!

Recall that the CAVI update algorithm for a Gaussian Mixture Model is as follows:

Algorithm 2: CAVI for a Gaussian mixture model

Input: Data $x_{1:n}$, number of components K, prior variance of component means σ^2

Output: Variational densities $q(\mu_k; m_k, s_k^2)$ (Gaussian) and $q(c_i; \varphi_i)$ (K-categorical)

Initialize: Variational parameters $\mathbf{m} = m_{1:K}$, $\mathbf{s}^2 = s^2_{1:K}$, and $\boldsymbol{\varphi} = \varphi_{1:n}$

while *the* ELBO *has not converged* **do**

  for $i \in \{1, \ldots, n\}$ **do**

    Set $\varphi_{ik} \propto \exp\{\mathbb{E}[\mu_k; m_k, s_k^2] x_i - \mathbb{E}[\mu_k^2; m_k, s_k^2]/2\}$

  end

  for $k \in \{1, \ldots, K\}$ **do**

    $$\text{Set } m_k \longleftarrow \frac{\sum_i \varphi_{ik} x_i}{1/\sigma^2 + \sum_i \varphi_{ik}}$$

    $$\text{Set } s_k^2 \longleftarrow \frac{1}{1/\sigma^2 + \sum_i \varphi_{ik}}$$

  end

  Compute ELBO$(\mathbf{m}, \mathbf{s}^2, \boldsymbol{\varphi})$

end

return $q(\mathbf{m}, \mathbf{s}^2, \boldsymbol{\varphi})$

Note that our notation differs slightly, with φ corresponding to a and s^2 corresponding to v^2. We also have $K = 2$. Assume initial parameters, $\mathbf{m} = [0.5, 0.5]$, $\mathbf{v}^2 = [1, 1]$ and $a_i = [0.3, 0.7]$ for all $i \in n$ and a sample $x = [0.1, -0.3, 1.2, 0.8, -0.5]$. Also assume prior variance $\sigma^2 = 0.01$

Write a python script implementing the above procedure and run it for 5 epochs. You should submit your code to autolab as a .tar file named cavi.tar containing a single file cavi.py. You can create that file by running:

```
tar -cvf cavi.tar cavi.py
```

from the directory containing your code.

After the fifth epoch, report

1. (2 points) The variational parameters $\mathbf{m}$.

$\mathbf{m}$	0.00634	0.00634

2. (2 points) The variational parameters $\mathbf{v}^2$.

$\mathbf{v}^2$	0.00976	0.00976

3. (2 points) The variational parameters $\mathbf{a}$.

a_1	0.5	0.5
a_2	0.5	0.5
a_3	0.5	0.5
a_4	0.5	0.5
a_5	0.5	0.5

Hint:

1. Note that the expectation update for $\mathbf{a}$ does not depend on μ. (Why?)
2. The expectation of the square of a Gaussian random variable is $\mathbb{E}[X^2] = Var[X] + \mathbb{E}([X])^2$.

1.4 Variational Inference vs. Monte Carlo Methods

Let's end with a brief comparison between variational methods and MCMC methods. We have seen that both classes of methods can be used for learning in scenarios involving latent variables, but both have their own sets of advantages and disadvantages. For each of the following statements, specify whether they apply more suitably to VI or MCMC methods:

1. (1 point) Transforms inference into optimization problems.
   - ● Variational Inference
   - ○ MCMC
2. (1 point) Is easier to integrate with back-propagation.
   - ● Variational Inference
   - ○ MCMC
3. (1 point) Involves more stochasticity.
   - ○ Variational Inference
   - ● MCMC
4. (1 point) Non-parametric.
   - ○ Variational Inference
   - ● MCMC
5. (1 point) Is higher variance under limited computational resources.
   - ○ Variational Inference
   - ● MCMC

1.5 Wrap-up Questions

1. (1 point) **Multiple Choice:** Did you correctly submit your code to Autolab?
   - ● Yes
   - ○ No
2. (1 point) **Numerical answer:** How many hours did you spend on this assignment?.

8

1.6 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.

   No

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.

   No

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.

   No