CS 330 Autumn 2022 Homework 4
Advanced Meta-Learning Topics
Due Monday, November 14th, 11:59 PM PT

SUNet ID:
Name: Seok, Jeongeum
Collaborators:

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Overview

This assignment will explore advanced meta-learning topics. In particular, you will answer conceptual questions about whether memorization can occur in a few realistic meta-learning scenarios. You will also derive an objective for a Bayesian meta-learning setting where, in addition to a dataset, metadata is available for all tasks. This assignment does not involve programming.

Grading policy: This assignment is optional, and accounts for up to 15% of your grade. Your grade for this homework will replace the grade of either one prior homework or part of the final project, whichever choice results in the highest final grade. Attempting this homework will never *harm* your grade.

Submission: To submit your homework, submit one PDF report to Gradescope containing written answers to the questions below. The PDF should also include your name and any students you talked to or collaborated with.

Problem 1: Memorization in Meta-Learning (6 Points)

In this problem, we will examine four task distributions in a meta-learning problem setting to determine whether or not memorization can occur. Specifically, we denote tasks as $\mathcal{T}_i$ and the task distribution as $p(\mathcal{T})$. We denote the dataset corresponding to the ith meta-training task as $\mathcal{D}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)$ and the dataset corresponding to the ith meta-testing task as $\mathcal{D}_i^* = (\boldsymbol{x}_i^*, \boldsymbol{y}_i^*)$. Here, $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iK}), \boldsymbol{y}_i = (y_{i1}, \ldots, y_{iK})$, and $p(\mathcal{T})$ determines the distributions that the datasets are sampled from. We denote the collection of all meta-training and meta-testing datasets as $\mathcal{M} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots\} \cup \{\mathcal{D}_1^*, \mathcal{D}_2^*, \ldots, \}$.

We adopt the definition of *complete meta-learning memorization* from [1]. We say that a task distribution can suffer from complete memorization if a meta-learning algorithm can achieve perfect performance for all meta-training datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots$ while ignoring the task training data (i.e. the support set).

For each of the following meta-learning task settings, answer **two questions** and explain your reasoning for each answer. First, state whether or not complete memorization can occur. Second, would the performance on the meta-test set be (1) equal or worse than random guessing (2) worse than meta-train but better than random guessing (3) as good as meta-train? For both questions, you are not required to provide a formal mathematical proof but you should explain your reasoning in one or two sentences. Some of the questions may have multiple plausible answers depending on how you interpret the scenario and what additional assumptions you make. Any logically consistent answer will receive a full grade.

(a) **Regression tasks from basis functions.** Consider regression tasks where the input and output domains are $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \mathbb{R}$, respectively. Each task contains 10 train and 10 test datapoints, and is constructed from a finite set of basis functions $\phi_i : \mathbb{R} \to \mathbb{R}$ for $i \in \{1, \ldots, n\}$ that are linearly independent, i.e., there does not exist $w_1, \ldots, w_n$ such that $\sum_i w_i \phi_i = 0$. Let $p(w)$ be a distribution over $\mathbb{R}^n$. Each task is constructed by sampling $w \sim p(w)$ and then using $x \mapsto \sum_i w_i \phi_i(x)$ as the ground-truth function, where the inputs x are sampled from Unif$([0, 1])$. (2 points)

Complete memorization can occur because algorithm can memorize all the datapoints.
The performance on the meta-test set would be as good as meta-train because basis functions are learnable.

(b) **Medical image classification.** Consider a meta-learning dataset for medical imaging, where the goal is for a model to be able to recognize a novel disease given a small number of labeled cases. The input is a medical image, such as an X-ray or CT scan. The output is a binary label $y \in \{0, 1\}$ where 0 represents a sample from a healthy person and 1 represents a sample with a specific disease. Assume we have a large dataset of images corresponding to N different diseases, where N is a large number. To construct a meta-training task, we randomly sample one of the N diseases, then sample 5 healthy and 5 diseased images. Meta-testing tasks are constructed similarly from images corresponding to a set of M held-out diseases. (2 points)

Complete memorization cannot occur because the algorithm cannot achieve perfect performance for medical images while ignoring the support set.
The performance on the meta-test set would be as good as meta-train because medical images would share the similar distribution.

(c) **Robotic grasping.** Consider a simplified robotic grasping task, where the goal is to predict the three-dimensional coordinates ($\in \mathbb{R}^3$) of a target object to grasp. The robot is a movable arm firmly attached to a desk. Many different objects are placed on the desk, and each task consists of grasping a specific object. The meta-testing tasks correspond to held-out objects on the desk. The input is an image that shows everything in the robot's field of view, including multiple objects on a desk and a computer monitor. The target object for each task is written on a computer monitor, inside the camera's field of view. As task-specific adaptation, the robot is allowed

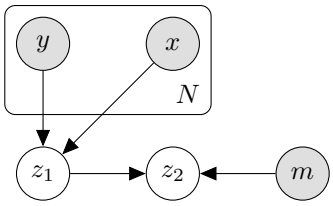

Figure 1: Plate notation diagram of Neural Processes.

$K = 5$ attempts to predict the grasp location; i.e. the support set includes 5 grasp attempts. During each attempt, the robot makes object coordinate predictions and receives a reward signal based on how close its prediction was to the object. At meta-test time, we test the robot on novel objects with the monitor turned off, and measure the average grasping success rate after the initial 5 trials. (2 points)

Complete memorization cannot occur because the algorithm cannot achieve perfect performance for images while ignoring the support set.
The performance on the meta-test set would be as good as meta-train because grasping a speicific object would share the similar distribution.

Problem 2: Bayesian Meta-Learning (9 Points)

In this problem, we consider a two-level variant of Neural Processes [2]. A novel component in our setup is a *task metadata* variable $m \in \mathbb{R}^D$ which succinctly summarizes aspects of each task. We assume that m is available for all tasks in the meta-training set.

Our probabilistic model involves two latent variables $z_1 \in \mathbb{R}^D$ and $z_2 \in \mathbb{R}^D$ and an observable task metadata variable $m \in \mathbb{R}^D$. We assume a prior $p(z_2)$ over the top-level variable z_2, and that z_1 is sampled according to the conditional distribution $p(z_1|z_2, m)$. The label corresponding to an example x follows the distribution $p(y|z_1, x)$. We use networks $q(z_1|x_{1:N}, y_{1:N})$ and $q(z_2|z_1, m)$ to perform amortized inference of the two hidden variables z_1 and z_2.

(a) Draw a plate notation diagram for this model. Your diagram should include all variables z_1, z_2, x, y, m with node colors reflecting whether each node is hidden or observable, and a plate representing the N examples inside each dataset. You can hand-draw a diagram or modify the provided `tikz` diagram for Neural Processes (Figure 1). (2 points)

(b) Complete the following derivation of an evidence lower bound (ELBO) for this model, given the dataset $(x_{1:N}, y_{1:N})$ and metadata m from each task. Your final objective must only involve terms that can be directly computed. Your derivation can use standard lemmas such as Jensen's inequality without proof. (7 points)

(*Hint 1: related derivations appear in section 2.1 of* [2] *and section 2∼2.1 of* [3].)

(*Hint 2: introducing new symbols such as* $X = x_{1:N}$ *and* $Y = y_{1:N}$ *can simplify your equations.*)

Your answer goes here. It should be possible to rearrange your final objective into the structure below:

$$\log p(y_{1:N}|x_{1:N}, m) = \cdots \geq \mathbb{E}_{q(z|x_{1:N}, y_{1:N})} \left[\sum_{i=1}^{N} \log p(y_i|z, x_i) \right] - D_{\mathrm{KL}} \left(q(z|x, y) || \, p(z) \right). \tag{1}$$

References

[1] Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, Chelsea Finn. *Meta-Learning without Memorization* https://arxiv.org/abs/1912.03820

[2] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S.M. Ali Eslami, Yee Whye Teh *Neural Processes* https://arxiv.org/abs/1807.01622

[3] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, Ole Winther *Ladder Variational Autoencoders* https://arxiv.org/abs/1602.02282