Policy Metrics for Hindsight MAML (IDEA)

Arsh Zahed[1]

***Abstract*— In this document, I outline a rough idea of Policy Metrics for Hindsight MAML (name not final). The goal is to utilize task-specific experience gained at the end of the meta-learning stage and after to educate the task-specific training in the future with guided exploration. This is done by storing a buffer of trajectories with their corresponding tasks. During and after the meta-learning stage, an additional network is trained to learn a metric between tasks. This metric is determined by the policies of those tasks, and so we additionally derive and define metrics on the space of policies for a given Markov Decision Process. This method is anticipated to to reduce to amount exploration required when training to a specific task, as it explores within a specific class of tasks where a local optima is hypothesized to lay. For sparse reward tasks, this should hopefully reduce the attempts required to find a policy that achieves a positive reward, while also helping convergence speeds for tasks that don't have sparse reward.**

I. Introduction and Motivation

The goal of this approach is to utilize significant experience gained in the meta-learning stage to further improve the learning during task-specific specific learning. For motivation, we look to the way humans process information and learn to perform new tasks. When humans attempt to learn by trial and error, we often times explore by using paths that worked for different tasks. For example, suppose you are new to the UC Berkeley campus. You have access to a map, but not any source for directions. You spend a day learning the performance of paths between specific buildings, say you attempt several paths for pairs in Soda, Dwinelle, Etcheverry, and VLSB. Now you are tasked with finding an optimal path between MLK and Cory. Your natural instinct would say, "MLK is awfully close to Etcheverry and Cory is awfully close to Cory, so I will take a very similar path," and your instinct would be correct.

II. Some Formalization

Thus, the proposal is to store past episodes with the task for which they were sampled for, $(\mathcal{T}, s_0, a_0, s_1, a_1, ..., a_{T-1}, s_T)$, in a "Replay Buffer" R to be used later with importance sampling. This isn't a true experience replay buffer per say, but rather a dataset of episodes. Now we don't simply store any set of trajectories. Rather, the trajectories are selected are ones that exemplified great performance when training during the meta-learning stage. However, we can't just pick any set of past trajectories to learn from. The trajectories must be picked from tasks that are similar to the one we are now specifically learning for, and must be trajectories that performed significantly well. Thus, we now need to learn a metric between arbitrary tasks. This itself is a meta-learning problem.

There are several ways to approach this, but I propose Siamese Network for this task. Let θ represent the parameters of our agaent, ϕ represent the parameters of our Siamese Network, π_θ the policy under parameters θ, $\mathcal{T}$ the task, and $\mathcal{L}_\mathcal{T}$ the loss function for task T. During our initial meta-learning via MAML, for each meta-gradient, we take a gradient step for each sampled task $\mathcal{T}_1, ..., \mathcal{T}_N$. Formally, for task $\mathcal{T}_i$ with learning rate α, our task-specific update rule is

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\pi_\theta)$$

This invokes, a set of N new policies, $\pi_{\theta_1'}, ..., \pi_{\theta_N'}$. It then becomes a task of training the Siamese Network, $f_\phi(\mathcal{T}_i, \mathcal{T}_j)$ to differentiate between these N tasks, where the network takes in two tasks and outputs a vale in the range $[0, 1]$ representing it's confidence the two tasks are similar. If two tasks result in the same or similar policy under the update rule, we wish to train the network to pair them together. An information-theoretic approach would be to utilize a statistical metric. Let $J(p, q)$ denote the Jensen-Shannon divergence (a true metric on the set of distributions), defined as

$$J(p, q) := \frac{D(p||\frac{p+q}{2}) + D(q||\frac{p+q}{2})}{2}$$

(Note, this can be approximated with the square root symmetric Kullback-Leibler Divergence for our case. Also note that technically $\sqrt{J(\cdot, \cdot)}$ is the metric). The choice of the Jensen-Shannon Divergence is supported in section 3, and the sampling method for states is given in section 4. Then we can define two tasks, $\mathcal{T}_i$ and $\mathcal{T}_j$, to be of the same "class" if

$$\frac{1}{N} \sum_{k=1}^{N} \sqrt{J\left(\pi_{\theta_i'}(s_k), \pi_{\theta_j'}(s_k)\right)} < \epsilon$$

for $\epsilon > 0$ chosen as a threshold. This allows us to apply cross-entropy loss on the Siamese Network and run Stochastic Gradient Descent as usual.

After meta-learning and creating a replay buffer, we have a trained initial policy π_θ and trained Siamese Network f_ϕ. During task-specific training, we sample k trajectories from R along with their corresponding tasks, filtered to be trajectories whose corresponding tasks are determined to be similar by f_ϕ. These datapoints can then be used alongside traditional exploration methods using importance sampling, and gradient descent can be run as usual.

*University of California, Berkeley

Algorithm 1 Hindsight MAML - Task Metric Training

$\theta \leftarrow$ MAML
Initialize buffer R
Initialize Siamese Net f_ϕ
for numIter **do**
  for $i \in [N]$ **do**
    $\tau \leftarrow$ Sample$(\pi_\theta, \mathcal{T}_i)$
    $\theta_i' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\pi_\theta, \tau)$
    $\tau' \leftarrow$ Sample$(\pi_{\theta_i'}, \mathcal{T}_i)$
    if $\mathcal{L}_{\mathcal{T}_i}(\pi_{\theta_i'}, \tau') < \delta$ **then**
      $R \leftarrow R \bigcup \{\tau', \mathcal{T}_i\}$
  for $i, j \in [N] \times [N]$ **do**
    if $P(\pi_{\theta_i'}, \pi_{\theta_j'}) < \epsilon$ **then**
      label $\leftarrow 1$
    else
      label $\leftarrow 0$
    $\phi \leftarrow \phi - \beta \nabla_\phi \mathcal{L}_\phi \left(f_\phi\left(\mathcal{T}_i, \mathcal{T}_j\right), \text{label}\right)$
return R, f_ϕ

III. Interpretation of Jensen-Shannon Divergence

So far, the Jensen-Shannon Divergence has been mentioned simply as it's use as a metric between distributions, with little explanation as to why this exact metric serves useful. To understand the choice of this, we first explain entropy, conditional entropy, and mutual information. For the ease of understanding, we restrict to the discrete case, but the LDPP/differential entropy interpretations follow similarly.

The entropy of a discrete random variable is a measure of the uncertainty in the random variable. Intuitively, for a discrete distribution the entropy function should be maximized by the uniform distribution, be invariant to permutations in the PMF, and distribute over addition. As shown by Shannon, for a discrete random variable X, the only function H that satisfies this is

$$H(X) := \sum_x P_X(x) \log \frac{1}{P_X(x)} = \mathbb{E}_{P_X(x)}\left[\log \frac{1}{P_X(x)}\right]$$

Assuming the log is base 2, this can be thought of as the number of bits required to optimally encode a random variable X (such a coding can be done with a Huffman Tree).

Conditional entropy $H(X|Y)$ is the expected entropy of random variable X after observing random variable Y. Thus,

$$H(X|Y) := \mathbb{E}_{P_Y(y)}\left[\mathbb{E}_{P_X(x)}\left[\log \frac{1}{P_{X|Y}(x)}\right]\right]$$

This then leads us to Mutual Information. Mutual Information is simply a measure of the amount of information one random variables gives about another. Intuitively, it is given by

$$I(X;Y) := H(X) - H(X|Y) = H(Y) - H(Y|X)$$

$$= \mathbb{E}_{P_{XY}(x,y)}\left[\log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)}\right]$$

So how does this come into play with the Jensen-Shannon Divergence? Let's look at the function for $J(P,Q)$. The divergence is basically the average KL-Divergence of each distribution to a mixture of the two. Letting X be an appropriate random variable for P and Q and letting Z be an indicator variable such that, $X \sim P$ if $Z = 0$ and $X \sim Q$ if $Z = 1$, then we have the identity

$$J(P,Q) = I(X;Z)$$

(Note that with no information on Z, $X \sim (P+Q)/2$). We can now relate this equality to our policies from above. If we have

$$\frac{1}{N}\sum_{k=1}^{N} \sqrt{J\left(\pi_{\theta_i'}(s_k), \pi_{\theta_j'}(s_k)\right)} < \epsilon$$

then we can say that on average, if we sample our actions from a mixture of policies $\pi_{\theta_i'}$ and $\pi_{\theta_j'}$, knowing from which policy the action was sampled gives us no additional information on what the action is. That is to say, on average the policies are indistinguishable. This is powerful, as it confirms the notion of similarity in decisions that is needed to compare policies.

IV. Policy Metrics

A. Equivalence Classes on Policies

In definition, two policies π_θ and π_ϕ are equivalent if $\pi_\theta(s) = \pi_\phi(s)$, for all states $s \in S$. However, consider two policies the MDP with two states, 0 and 1, with self loops and transitions both ways with deterministic transition probabilities. Then consider two policies, π_θ which always stays on 0 and 1, and π_ϕ which always stays on 0 but transitions on 1. If the initialization distribution, $p_0(s)$ has a non-zero probability of initiating at 1, then the policies may behave differently during their trajectories. However, if the initialization distribution always initializes at 0, both the policies will always behave the same. Our goal is to be able to identify policies that behave the same under an MDP, regardless of whether or not they are actually equal. In other words, we wish to form an equivalence relation.

Definition 1 - Policy Equivalence Relation
Let Π be the set of all policies on the given MDP, and let $\pi_\theta, \pi_\phi \in \Pi$. Let $p_\theta(\tau)$ be the joint probability distribution of trajectory τ under policy π_θ. We say $\pi_\theta \sim \pi_\phi$ if

$$p_\theta(\tau) = p_\phi(\tau), \;\; \forall \tau \in (S \times A)^H$$

Additionally, we define

$$[\pi_\theta] := \{\pi_\phi \in \Pi \;\mid\; \pi_\theta \sim \pi_\phi\}$$

Lastly, define the set of all equivalence classes on Π

$$\Pi/\!\sim := \{[\pi] \;\mid\; \pi \in \Pi\}$$

This definition is trivially a valid equivalence relation that maintains the property of equivalence in behavior.

B. Metric Space of Policies

Our goal is now to find a metric between policies. Without regard for equivalence relations, we can define a true metric on the set of policies.

Definition 2 - Policy Metrics
Let S be the set of states, and A be the set of possible actions. Additionally let $J(\cdot,\cdot)$ be the Jensen-Shannon Divergence, p_0 be the distribution for starting states, $\pi_\theta, \pi_\phi \in \Pi$ be policies on the MDP, and q_S be a distribution over the set of states where $q_S(s) \neq 0$, $\forall s \in S$. Then, we define a Policy Metric as

$$d_P(\pi_\theta, \pi_\phi) := \mathbb{E}_{s\sim q_S}\left[\sqrt{J\left(\pi_\theta(s), \pi_\phi(s)\right)}\right]$$

It can be checked that d_P is a true metric over Π, thus making (Π, d_P) a metric space. However, this comes with the drawbacks that two equivalent but unequal policies, $\pi_\theta \sim \pi_\phi$, will have a positive distance by the metric. For many purposes, this is undesirable, as these two policies realistically will never differ in behavior. To help combat this, we can instead form a metric over the set of equivalence classes, $\Pi/\!\sim$.

Definition 3 - Equivalence Policy Metric
Let S be the set of states, and A be the set of possible actions. Additionally let $J(\cdot,\cdot)$ be the Jensen-Shannon Divergence, p_0 be the distribution for starting states, and $[\pi_\theta], [\pi_\phi] \in \Pi/\!\sim$ be policy equivalence classes on the MDP. Then let

$$S' := \big\{s \in S \;\mid\; \exists\, \tau_\theta, \tau_\phi, \text{ s.t. } s \in \tau_\theta, s \in \tau_\phi, \; p_\theta(\tau_\theta) \neq 0,\; p_\phi(\tau_\phi) \neq 0\big\}$$

S' is necessarily non-empty as all policies obey the same initialization distribution. Let $q_{S'}$ be a distribution over S such that $q_{S'}(s) \neq 0$ for all $s \in S'$ and $q_{S'}(s) = 0$ for all $s \notin S'$. Then we can define the Equivalence Policy Metric

$$d_E([\pi_\theta], [\pi_\phi]) := \mathbb{E}_{s\sim q_{S'}}\left[\sqrt{J\left(\pi_\theta(s), \pi_\phi(s)\right)}\right]$$

This is a true metric over $\Pi/\!\sim$, thus making $(\Pi/\!\sim, d_E)$ a metric space. Forming this metric over the set of equivalence classes bypasses the issue of differentiating between policies that behave the same. However, computing this metric empirically can often be intractable or impossible, especially in continuous cases, as the set S' is hard to compute. For this reason, we are often unable to use this metric. To get the best of both worlds in applications, it is possible to sacrifice the property of true metric in order preserve the property of distance in behavior.

NOTE: If valid, a better description would be

$$d_E([\pi_\theta], [\pi_\phi]) := \inf\left\{\mathbb{E}_{\tau\sim p_\theta}\left[\sum_{s\in\tau}\sqrt{J\left(\pi_\theta(s), \pi_\phi(s)\right)}\right]\right\} + \inf\left\{\mathbb{E}_{\tau\sim p_\phi}\left[\sum_{s\in\tau}\sqrt{J\left(\pi_\theta(s), \pi_\phi(s)\right)}\right]\right\}$$

where we assume the infimums are taken with $\pi_\theta \in [\pi_\theta]$, and $\pi_\phi \in [\pi_\phi]$. I have yet to prove this for triangle inequality, but the rest of the properties of metrics follow trivially. Additionally, this would make the definition of Policy Divergence a lot more theoretically backed, as it is then an upper-bound on the metric distance between the equivalence classes of the two policies.

Definition 4 - Policy Divergence
Let S be the set of states, and A be the set of possible actions. Additionally let $J(\cdot,\cdot)$ be the Jensen-Shannon Divergence, $\pi_\theta, \pi_\phi \in \Pi$ be policies on the MDP, τ be a trajectory of a set of states and actions at given time steps, and $p_\theta(\tau)$ be the probability of trajectory τ given policy π_θ. Then, we define the Policy Divergence as

$$P(\pi_\theta, \pi_\phi) := \mathbb{E}_{\tau\sim p_\theta}\left[\sum_{s\in\tau}\sqrt{J\left(\pi_\theta(s), \pi_\phi(s)\right)}\right] + \mathbb{E}_{\tau\sim p_\phi}\left[\sum_{s\in\tau}\sqrt{J\left(\pi_\theta(s), \pi_\phi(s)\right)}\right]$$

This definition is neither a metric, nor a semi-metric, as it does not obey triangle inequality. However, it does preserve the idea of difference in behavior, while giving a rough estimate on the distance between the two policies. From this definition, if $\pi_\theta \sim \pi_\phi$, then we get $P(\pi_\theta, \pi_\phi) = 0$, as we desire. Additionally, if $\pi_\theta \nsim \pi_\phi$, we get $P(\pi_\theta, \pi_\phi) > 0$, as desired. In general, this should be treated as an upper bound on the metric between equivalence classes of the two policies.

Additionally, in the continuous case, calculating the Jensen-Shannon Divergence can often be intractable. Thus, we introduce the Symmetric Kullback-Leibler Divergence

$$S(p, q) := \frac{D(p||q) + D(q||p)}{2}$$

We can then obtain a loose bound with

$$J(p, q) \leq \frac{1}{2}S(p, q)$$

Or a tighter bound with

$$J(p, q) \leq \ln\left(\frac{2}{1 + e^{-S(p,q)}}\right)$$

If the two distributions are multivariate Gaussians, these values have closed form solutions, and are much more tractable to compute.

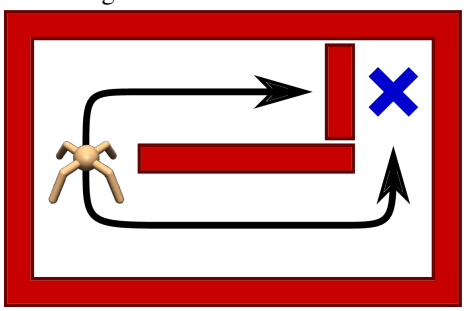
V. Todos and Experiments

Theres a range of things to complete and areas to explore. (a)

1) Experimental Results to Support the Theory of the Metrics and Policy Divergence

   This is nearly complete. Plots will be added soon.

2) Expert Learning On Tasks



The most promising experimental results will come from here. The goal is to train N experts for N tasks using regular Natural Policy Gradients. These tasks should ”span” the space of tasks in a sense (this is to be defined in rigor later). We can then use the proposed metric learning and importance sampling method on these tasks specifically.
For example, in the image above, we train agents to go to several places in the mini-maze, train the Siamese Network to distinguish between tasks based off those expert policies, and then train new agents using that Siamese Newtork and trajectory samples.

3) Siamese Network on Tasks

Before deploying the Siamese Network in the algorithm, we will first train it to see if it can distinguish between known optimal policies. This is to be tested on both discrete and continuous action spaces, including mazes and motion control.

4) Integrate Into MAML/Reptile

Basically what it says. Add the Importance Sampling and Siamese Network to add to the MAML algorithm.

VI. Notes

First, I’d like to say this document is far from complete. Next, I believe there are more viable approaches than just Siamese Networks. One area to look into would be Exemplar SVMs or Amortized Exemplar Networks. I also will soon add the actual algorithm to this once the idea is fleshed out. Lastly, I will soon go back and add references as needed.

p