Covariate Distribution Aware Meta-learning

Amrith Setlur [*1] **Saket Dingliwal** [*1] **Barnabas Poczos** [1]

Abstract

Meta-learning has proven to be successful at few-shot learning across the regression, classification and reinforcement learning paradigms. Recent approaches have adopted Bayesian interpretations to improve gradient based meta-learners by quantifying the uncertainty of the post-adaptation estimates. Most of these works almost completely ignore the latent relationship between the covariate distribution $(p(x))$ of a task and the corresponding conditional distribution $p(y|x)$. In this paper, we identify the need to explicitly model the meta-distribution over the task covariates in a hierarchical Bayesian framework. We begin by introducing a graphical model that explicitly leverages very few samples drawn from $p(x)$ to better infer the posterior over the optimal parameters of the conditional distribution $(p(y|x))$ for each task. Based on this model we provide an inference strategy and a corresponding meta-algorithm that explicitly accounts for the meta-distribution over task covariates. Finally, we demonstrate the significant gains of our proposed algorithm on a synthetic regression dataset.

1. Introduction

Learning quickly or with very few samples has been a long-term goal of the machine learning community. The field of meta-learning has recently made significant strides towards achieving that goal. Meta-learning (Nichol et al., 2018; Ravi & Larochelle, 2016; Finn et al., 2017) comprises of a set of algorithms designed to exploit prior experiences from multiple tasks (drawn from a task distribution) for improving sample-efficiency on a new but related task from the same distribution. Given the increasing cost of getting annotated samples on an ever-increasing variety of related tasks, the practical scope of these algorithms is immense.

Most meta-learning methods can be classified into two broad categories (i) *gradient-based* (Ravi & Beatson, 2018; Denevi et al., 2019; Finn et al., 2017) approaches that meta-learn parameters of optimization algorithms (like initialization and learning rate) in a way that the meta-learner (optimizer) is amenable to quickly adapt on a new task by performing gradient descent on a very small number of labeled samples, and (ii) *amortized-inference* (Snell et al., 2017; Lee et al., 2019; Bertinetto et al., 2018) based approaches that directly infer the optimal parameters of a new task without performing any gradient based optimization. In general, such algorithms learn to adapt the parameters of a complex neural network using only a few samples from an unseen task in a way that the *adapted network* generalizes well (Wang et al., 2019). In this work, although we focus on improving gradient-based methods, we believe that our core idea can be adapted to the latter as well. Recent works (Finn et al., 2018), (Ravi & Beatson, 2018), (Kim et al., 2018) have used a Bayesian framework to learn a suitable prior over the network parameters by leveraging the inherent structure of the task distribution. By viewing the parameters of a meta-learner through a Bayesian lens we

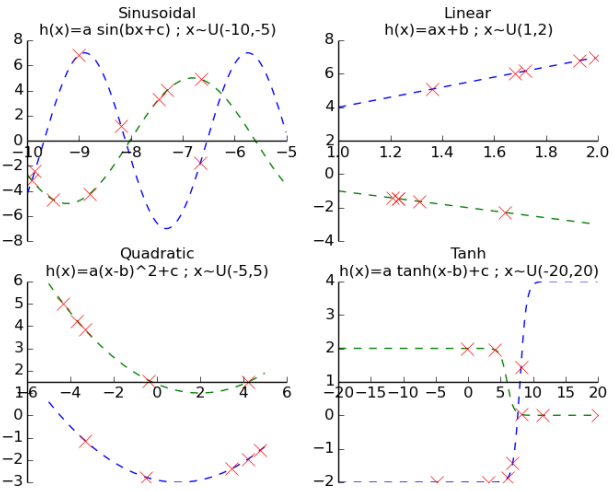

Figure 1. Samples of tasks drawn from a meta-distribution categorized based on its corresponding hypothesis class. The red crosses indicate the labeled sampled points and the blue, green dashed lines represent the true hypothesis for two different tasks from the same class. We can clearly see that the hypothesis class is not independent of the task specific input distribution $p(x)$. For example, the covariate distribution for *sinusoidal* tasks have a higher support in the negative region (in R) where as the support is confined to a much narrower positive region for *linear* tasks.

[*]Equal contribution [1]Carnegie Mellon University. Correspondence to: Amrith Setlur <asetlur@cs.cmu.edu>.

can use the predictive posterior (Gal & Ghahramani, 2016) to estimate the uncertainty of the adapted parameters for each task (Ravi & Beatson, 2018).

In a Bayesian meta-learner (Kim et al., 2018; Finn et al., 2018; Ravi & Beatson, 2018), the posterior over the adapted network parameters for a new task is typically inferred using a few samples from the task along with a meta-learned prior. In this work, we hypothesize that the covariate distribution of a task can also influence the posterior over the adapted network parameters. To the best of our knowledge none of the existing meta-learning algorithms like Bertinetto et al. (2018); Rajeswaran et al. (2019); Ravi & Beatson (2018); Finn et al. (2018) explicitly utilize the information present in the covariates to improve the estimate of the adapted parameters. This is done by modeling the latent factors of the covariate distribution. We define a prior not only on the network parameters (which determine the conditional $p(y|x)$) but also on the covariate distribution $p(x)$. Our meta-learning objective involves maximizing the joint likelihood $p(x, y)$ as opposed to just $p(y|x)$ which leads to meta-parameters sharing information about the covariates across tasks, in addition to the optimal network parameters. This way the latent factors of the covariate distribution $p(x)$ of a new task can be quickly inferred from very few covariates. Finally, the inferred latent covariate factors are used to infer the posterior over the adapted network parameters.

For simplicity, we motivate the need to model covariates via a synthetic example in Fig. 1. The meta-distribution consists of tasks with optimal hypothesis $h \in \mathcal{H}{=}\bigcup \{\mathcal{H}_s, \mathcal{H}_l, \mathcal{H}_t, \mathcal{H}_q\}$ that can be classified into four hypothesis classes: *sinusoidal* ($\mathcal{H}_s$), *linear* ($\mathcal{H}_l$), *tanh* ($\mathcal{H}_t$) and *quadratic* ($\mathcal{H}_q$). We note that the support of the input distribution is vastly different for each of the four hypothesis classes. Thus, intuitively we can see that inferring properties of the covariate distribution of a task can be helpful in adjusting the posterior over the network parameters. For example, for a given task if we only observe negative covariates in the range $[-10, -5]$ we can adapt the posterior to have a higher measure for $\mathcal{H}_s$ (sinusoidal hypothesis).

We can generalize the above intuition to real-world meta-learning problems as well, especially for high-dimensional data like images. Particularly, in the limited availability of labeled data, semi-supervised methods (Chapelle et al., 2009) tend to leverage unlabeled data (and hence covariate distribution) to attain generalizable models (Berthelot et al., 2019) On similar lines, modeling the meta-distribution over the covariates can help inferring the task-specific covariate distribution which can then better inform task-specific features for image classification. In few-shot classification, images for different tasks can lie on different manifolds (Saul & Roweis, 2003) of varying complexities. Information about the covariates (by modeling the unlabeled data) can help us better estimate the required complexity of the discriminative features for a given task. For example, distinguishing species of plants can be considered harder than classifying mammals since the features of plant images may be cluttered on a low-dimensional manifold as opposed to the possibly well separated features in the case of mammals. Thus the former may require having complex (non-linear) decision boundaries as opposed to simpler linear classifiers in the case of the latter.

The main contributions of our work are as follows: (1) we identify the need to model the latent structure present in the covariate distributions $(p(x))$ for a sequence of tasks (2) to the best of our knowledge we are the first to propose a Bayesian framework which exploits this latent information to better infer the posterior over the adapted network parameters (that define $p(y|x)$) (3) we propose a gradient based model-agnostic meta-learning algorithm that is an instantiation of our probabilistic theory and demonstrate its benefits on synthetic regression datasets.

2. Related Work

Our methodology is complementary to most existing works in the probabilistic meta-learning literature. We borrow the basic hierarchical Bayes framework from Ravi & Beatson (2018); Finn et al. (2018) and extend it to model Bayesian variables that generate the covariate distribution for a task. This enables our method to be model-agnostic while having the ability to benefit from the latent relationship between the task covariates and the optimal parameters as mentioned in Sec. 1. In the non-Bayesian setting, the M-MAML algorithm proposed by Vuorio et al. (2019) is mildly similar to our approach in the sense that they learn task specific initializations instead of a single one as originally introduced by Finn et al. (2017). M-MAML uses the labeled samples to infer an initialization for a given task and hence one can view the covariate distribution as being used indirectly. But they fail to explicitly model the mutual information between the covariates and adapted parameters. On the other hand, our approach is more direct since it *first* infers the posterior over the latent factors of the covariate distribution via a maximum likelihood objective and then uses the inferred posterior to improve the adaptation of network parameters. Additionally, our framework is capable of modeling the uncertainty of the adaptation which can prove to be be critical in the few shot scenario.

3. Methodology

We begin by introducing some notations for the meta-learning setup used in the rest of the paper followed by the proposed probabilistic framework which explicitly exploits (i) the structure of the covariate distributions across

tasks (ii) the relation between the covariate distribution and optimal hypothesis for a given task. We then derive the Maximum Likelihood Estimation (MLE) objectives for the observed variables in our model and show how the MLE derivations can inform a novel meta-learning objective. Finally, we discuss a specific meta-learning algorithm that can efficiently optimize the proposed objective. We do this via an instantiation of the generic approach obtained by making certain simplifying assumptions in the original framework.

Notations We are given a sequence of n tasks $\{\mathcal{T}_i\}_{i=1}^n$ with each task $\mathcal{T}_i$ having m labeled samples given by the dataset $\mathcal{D}_i = \{\mathbf{x}_j^{(i)}, \mathbf{y}_j^{(i)}\}_{j=1}^m$ where $\mathbf{x}_j^{(i)} \in \mathcal{X} \subset \mathrm{R}^k$ and $\mathbf{y}_j^{(i)} \in \mathcal{Y} \subset \mathrm{R}$. Following the definitions introduced by Finn et al. (2018) we split the dataset $\mathcal{D}_i := \{\mathcal{D}_i^S, \mathcal{D}_i^Q\}$ into support ($\mathcal{D}_i^S$) and query ($\mathcal{D}_i^Q$) sets respectively with $|\mathcal{D}_i^S| = m', |\mathcal{D}_i^Q| = m - m'$. Each sample in $\mathcal{D}_i$ is drawn from the joint distribution $p_i(x, y)$ over $\mathcal{X} \times \mathcal{Y}$ with the marginals given by $p_i(x)$ and $p_i(y)$.

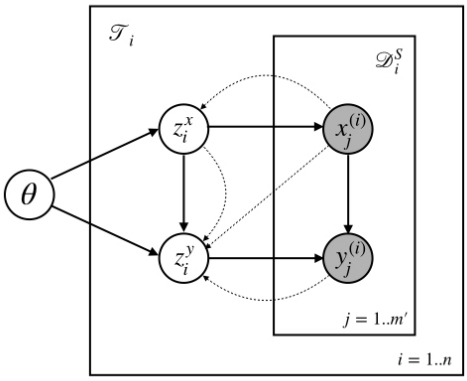

Figure 2. A graphical model representing our hierarchical Bayes framework with task-agnostic meta-parameters θ and task-specific latent variables z_i^x, z_i^y which influence the marginal $p_i(x)$ and conditional $p_i(y|x)$ distributions respectively. The dotted lines denote the variational approximations introduced over the true posteriors.

The probabilistic model we consider in our work is summarized in Fig. 2. Without making any assumptions on the nature of $p_i(x, y)$, we assume the existence of meta-parameters θ that govern the common structure shared across the set of joint distributions $\{p_i(x, y)\}_{i=1}^n$. Within each task the generative model for the input $x^{(i)}$ involves a random variable z_i^x which we shall refer to as the latent factors of the covariate distribution. Also, each task has an additional latent variable z_i^y which plays a role in the generative model for the response variable $y^{(i)}$ given the input $x^{(i)}$. In most settings, a naive assumption of independence is made over the latent factors z_i^x and z_i^y. On the other hand, we refrain from making such assumptions and instead exploit the information present in the covariates $\{\mathbf{x}_j^{(i)}\}_{j=1}^m$ to better infer the posterior over the latent variable z_i^y which influences the conditional $p_i(y|x)$.

3.1. Formal Derivations

In this section, we derive a lower bound for the likelihood of the observed data $\{\mathcal{D}_i^S\}_{i=1}^n$ using hierarchical variational inference. This gives us the meta-learning objective that can be optimized using standard gradient-based approaches.

$$\log\, p(\{\mathcal{D}_i^S\}_{i=1}^n) = \log \int_\theta p(\{\mathcal{D}_i^S\}_{i=1}^n|\theta)p(\theta)\, d\theta \tag{1}$$
$$\geq \mathrm{E}_{q(\theta;\beta)}\left[\sum_{i=1}^n \log p(D_i^S|\theta)\right] - \mathrm{KL}(q(\theta;\beta)||p(\theta))$$

In the above equation, the distribution $q(\theta;\beta)$ is a variational approximation (with parameters β) for the true posterior over the meta-parameter θ. For the derivations henceforth we shall drop the notations i and S when understood from context. The log-likelihood of the dataset $\mathcal{D}_i$ given by $p(\mathcal{D}_i|\theta)$, can be written as an integral over the factors $p(\mathcal{D}_i|z_i^x, z_i^y)$, $p(z_i^y|z_i^x, \theta)$ and $p(z_i^x|\theta)$.

$$p(\mathcal{D}_i|\theta) = \int_{z_i^x}\int_{z_i^y} p(\mathcal{D}|z_i^x, z_i^y)p(z_i^y|z_i^x, \theta)p(z_i^x|\theta)\;\; dz_i^y dz_i^x$$

To lower bound the log of the above objective we introduce two variational approximations (i) $q(z_i^x;\kappa_i)$ with parameters κ_i for the true posterior $p(z_i^x|\{\mathbf{x}_j^{(i)}\}_{j=1}^{m'})$ and (ii) $q(z_i^y;\lambda_i)$ with parameters λ_i for the true posterior $p(z_i^y|z_i^x, \mathcal{D}_i^S)$.

$$\log p(\mathcal{D}_i|\theta) \geq \mathrm{E}_{q(z_i^x;\kappa_i)}\left[\log \int p(\mathcal{D}_i|z_i^x, z_i^y)p(z_i^y|z_i^x, \theta)dz_i^y\right]$$
$$- \mathrm{KL}(q(z_i^x;\kappa_i)||p(z_i^x|\theta)) \tag{2}$$

Since z_i^x is the latent factor in the generative model for $x^{(i)}$ and z_i^y is the corresponding latent variable for $y^{(i)}$, we arrive at the independence: $\mathbf{y}_j^{(i)} \perp z_i^x|z_i^y, \mathbf{x}_j^{(i)}$ and $\mathbf{x}_j^{(i)} \perp z_i^y|z_i^x$. Based on this, we finally arrive at the following Evidence Lower Bound (ELBO) for $\log p(\mathcal{D}_i|\theta)$ which we shall refer to as $\mathcal{L}_{\mathcal{D}_i}(\kappa_i, \lambda_i, \theta)$.

$$\mathcal{L}_{\mathcal{D}_i}(\kappa_i, \lambda_i, \theta) = \mathrm{E}_{q(z_i^x;\kappa_i)}\left[\sum_{j=1}^{m'} \log p(\mathbf{x}_j^{(i)}|z_i^x)\right] \tag{3}$$
$$-\mathrm{KL}(q(z_i^x;\kappa_i)||p(z_i^x|\theta)) + \mathrm{E}_{q(z_i^x;\kappa_i)}\; \mathcal{L}'_{\mathcal{D}_i}(z_i^x, \lambda_i, \theta)$$
$$\mathcal{L}'_{\mathcal{D}_i}(z_i^x, \lambda_i, \theta) = \mathrm{E}_{q(z_i^y;\lambda_i)}\left[\sum_{j=1}^{m'} \log p(\mathbf{y}_j^{(i)}|\mathbf{x}_j^{(i)}, z_i^y)\right]$$
$$- \mathrm{KL}(q(z_i^y;\lambda_i)||p(z_i^y|z_i^x, \theta)) \tag{4}$$

Therefore, the ELBO on the likelihood of the dataset for i^{th} task is a function of the task-specific variational parameters

κ_i, λ_i and the variational meta-parameter β. For each task, the optimal variational parameters that approximate the true posteriors are distinct. Hence, κ_i, λ_i need to be adapted for each task individually. Given $\mathcal{L}_{\mathcal{D}_i}(\kappa_i, \lambda_i, \theta)$ we can re-write the lower bound in Eq. 1 as:

$$\log\ p(\{\mathcal{D}_i^{(S)}\}_{i=1}^n) \geq \mathcal{L}(\beta) \tag{5}$$
$$= \mathrm{E}_{q(\theta;\beta)}\left[\sum_{i=1}^n \mathcal{L}_{\mathcal{D}_i}(\kappa_i, \lambda_i, \theta)\right] - \mathrm{KL}(q(\theta;\beta)||p(\theta))$$

3.2. Algorithm

The primary aim of any meta-learning algorithm is to optimize for the meta-parameter θ given the datasets $\{\mathcal{D}_i\}_{i=1}^n$ from the corresponding sequence of tasks. This is generally a two step process where step-I involves identifying the optimal task-specific parameters using θ and the support set $\mathcal{D}_i^S$. In step-II, based on the task-specific adapted parameters from step-I the meta-parameter θ is optimized over the query set $\mathcal{D}_i^Q$. Within our framework, since both the meta-parameter θ and the task-specific parameters z_i^x, z_i^y are Bayesian random variables with variational parameters given by $\beta, \lambda_i, \kappa_i$ respectively, we instead define an algorithm to optimize the ELBO in Eq. 5.

$$\beta^* = \arg\min_\beta\ -\mathcal{L}(\beta) \tag{6}$$
$$= \arg\min_\beta\ -\mathrm{E}_{q(\theta;\beta)} \sum_{i=1}^n \mathcal{L}_{\mathcal{D}_i^Q}(\kappa_i^*, \lambda_i^*, \theta) + \mathrm{KL}(q(\theta;\beta)||p(\theta))$$
$$\kappa_i^*, \lambda_i^* = \arg\min_{\kappa_i, \lambda_i}\ -\mathrm{E}_{q(\theta;\beta)} \mathcal{L}_{\mathcal{D}_i^S}(\kappa_i, \lambda_i, \theta) \tag{7}$$

In order to optimize the objectives in Eqs. 6, 7 we introduce certain simplifying assumptions over each of the variational approximations in Sec. 3.1. [1]

Assumption 1. *The variational approximation $q(\theta;\beta)$ follows a $\delta-$distribution given by δ_β and the prior $p(\theta)$ is given by $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We note that the assumptions taken are effective in arriving at a computationally feasible algorithm and are*

Assumption 2. *The variational approximation $q(z_i^y;\kappa_i)$ follows a normal distribution with mean, diagonal covariance matrix given by $\kappa_i^* = (\gamma_\mu(\mathcal{D}_i^S;\beta), \gamma_\sigma^2(\mathcal{D}_i^S;\beta))$. The prior $p(z_i^x|\theta)$ is taken as $\mathcal{N}(\mathbf{0}, \mathbf{I})$. On the other hand, the distribution $q(z_i^x;\lambda_i)$ is given by a $\delta-$distribution: δ_{λ_i}.*

Using Assms. 1, 2 the Eqs. 6, 7 can be re-written as a two layer log-likelihood objective with an l_2 regularization term.

$$\beta^* = \arg\min_\beta\ -\sum_{i=1}^n \mathcal{L}_{\mathcal{D}_i^Q}(\kappa_i^*, \lambda_i^*, \beta) + \frac{1}{2}\|\beta\|_2^2 \tag{8}$$
$$\lambda_i^* = \arg\min_{\lambda_i}\ -\mathcal{L}_{\mathcal{D}_i^S}(\kappa_i^*, \lambda_i, \beta) \tag{9}$$
$$\kappa_i^* = (\kappa_i^*(\mu), \kappa_i^*(\sigma)) = (\gamma_\mu(\mathcal{D}_i^S;\beta), \gamma_\sigma(\mathcal{D}_i^S;\beta)) \tag{10}$$

To be concrete, the task-specific random variable z_i^y represents the parameters of the neural network for the i^{th} task, that takes as input $\mathbf{x}_j^{(i)}$ and outputs a prediction $\hat{\mathbf{y}}_j^{(i)}$. On the other hand, as stated previously z_i^x represents the latent variable for the covariate distribution $p_i(x)$. Furthermore, the optimal variational parameters for the distribution $q(z_x^i;\kappa_i)$ is given by κ_i^* which consists of the mean $\gamma_\mu(\mathcal{D}_i^S;\beta)$ and std. deviation $\gamma_\sigma(\mathcal{D}_i^S;\beta)$ of a normal distribution. Here, $\gamma_\mu(\cdot,\beta), \gamma_\sigma(\cdot,\beta)$ represent neural networks which take as input the support set $\mathcal{D}_i^S$ and output κ_i^*. It is important to note that even though the parameters of $\gamma_\mu, \gamma_\sigma$ are task-agnostic, the variational parameter κ_i^* is still different for each task and is determined using the covariates in $\mathcal{D}_i^S$.

Having identified κ_i^* we now describe the optimization algorithm for λ_i in Eq. 9. Notice that to obtain λ_i^* it is sufficient to only minimize the objective $-\mathrm{E}_{q(z_i^x;\kappa^*)}\mathcal{L}'_{\mathcal{D}_i}(z_i^x, \lambda_i, \beta)$. In Eq. 4 the KL term acts as a regularizer in the optimization objective for λ_i. Since the most common algorithm for optimization is Stochastic Gradient Descent (SGD) many meta-learning algorithms avoid the KL term by choosing a regularization specific to SGD. In most works (Ravi & Beatson, 2018; Finn et al., 2018; Kim et al., 2018), the KL term is a function of only the meta-parameter θ (or β given Assm. 1). Hence the regularization is induced by letting the initialization for the optimization of λ_i (given by $\lambda_i^{(0)}$) be determined by β. In our framework, we realize that the KL term is a function of both β and the latent variable z_i^x for the task specific covariate distribution. Hence we model the initialization $\lambda_i^{(0)}$ using a neural-network whose parameters are subsumed in β and thus without loss of expressivity $\lambda_i^{(0)} = f_\beta(z_i^x)$. Thus, the optimal parameters of the variational approximation (λ_i^*) would be given by performing K steps of SGD on the MLE objective in Eq. 4 with the initialization given by $f_\beta(z_i^x)$.

$$\lambda_i^* = \mathrm{E}_{q(z_i^x;\kappa_i^*)}\ \mathrm{SGD}^2(l_{\mathcal{D}_i}(\lambda_i), \lambda_i^{(0)} = f_\beta(z_i^x), K) \tag{11}$$
$$l_{\mathcal{D}_i}(\lambda_i) = -\sum_{j=1}^{m'} \log p(\mathbf{y}_j^{(i)}|\mathbf{x}_j^{(i)}, \lambda_i)$$

The expectation in Eq 11 is computed using monte-carlo approximation. We find that sampling a single value of $z_i^x \sim q(z_i^x;\kappa_i^*)$ is sufficient to optimize for λ_i.

[1] We note that the assumptions we consider are merely to induce a computationally feasible meta-learner and the theory involving a meta-distribution over covariates (in Sec. 3.1) is barely undermined by it.

[2] $\mathrm{SGD}(l(\lambda), \lambda^{(0)}, K)$ is the parameter obtained by performing K steps of SGD on the objective $l(\lambda)$ with initialization $\lambda^{(0)}$.

Algorithm 1 Meta-training Algorithm

Given: n datasets: $\{\mathcal{D}_i^S, \mathcal{D}_i^Q\}_{i=1}^n$, learning rates: η_0, η_1, number of update steps: K.
$p(\theta), p(z_i^x|\theta) \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$
for i=1 to n **do**
  $\kappa_i^*(\mu), \kappa_i^*(\sigma) \leftarrow \gamma_\mu(\mathcal{D}_i^S; \beta), \gamma_\sigma(\mathcal{D}_i^S; \beta)$
  $\lambda_i^{(0)} = f_\beta(\kappa_i^*(\mu) + \epsilon \circ \kappa_i^*(\sigma)); \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for t=0 to K-1 **do**
    $\lambda_i^{(t+1)} \leftarrow \lambda_i^{(t)} - \eta_0 \nabla_{\lambda_i^{(t)}} \sum_{j=1}^{m'} \log p(\mathbf{y}_j^{(i)}|\lambda_i^{(t)}, \mathbf{x}_j^{(i)})$
  end for
  $\lambda_i^* \leftarrow \lambda_i^{(K)}$
  $\beta \leftarrow \beta - \eta_1 \nabla_\beta \left[\sum_{i=1}^{n} \mathcal{L}_{\mathcal{D}_i^Q}(\kappa_i^*, \lambda_i^*, \beta) - \frac{1}{2}\|\beta\|_2^2 \right]$
end for

Finally, we note that the meta-parameter β constitutes the parameters of the network f_β which determines the initialization $\lambda_i^{(0)}$ as well as the parameters of $\gamma_\mu(\cdot;\cdot), \gamma_\sigma(\cdot;\cdot)$ which output κ_i^*. Thus, β is optimized to jointly maximize the likelihood of the covariates of a sequence of tasks as well as for learning to choose covariate dependent initializations suitable for few-shot adaptation. For the optimization objective in Eq. 8, we use the standard re-parameterization trick (2^{nd} step of the outer *for-loop* in Algorithm 1) commonly used in Variational Auto-Encoders (VAEs). This is done so as to be able to differentiate through the expectation over $q(z_i^x; \kappa_i^*)$ in Eqs. 3, 11. The step-by-step procedure for the meta-training and meta-testing phases are given by Algorithm 1 and Algorithm 2 respectively.

Algorithm 2 Meta-testing Algorithm on test task $\mathcal{T}$

Given: dataset $\mathcal{D}_\mathcal{T} = \{\mathbf{x}_j^{(\mathcal{T})}, \mathbf{y}_j^{(\mathcal{T})}\}_{j=1}^m$, parameter: β^*, learning rate η_0, number of update steps: K.
$\kappa_\mathcal{T}^*(\mu), \kappa_\mathcal{T}^*(\sigma) \leftarrow \gamma_\mu(\mathcal{D}_\mathcal{T}; \beta^*), \gamma_\sigma(\mathcal{D}_\mathcal{T}; \beta^*)$
$\lambda_\mathcal{T}^{(0)} = f_{\beta^*}(\kappa_\mathcal{T}^*(\mu) + \epsilon \circ \kappa_\mathcal{T}^*(\sigma)); \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for t=0 to K-1 **do**
  $\lambda_\mathcal{T}^{(t+1)} \leftarrow \lambda_\mathcal{T}^{(t)} - \eta_0 \nabla_{\lambda_\mathcal{T}^{(t)}} \sum_{j=1}^{m} \log p(\mathbf{y}_j^{(\mathcal{T})}|\lambda_\mathcal{T}^{(t)}, \mathbf{x}_j^{(\mathcal{T})})$
end for
$\lambda_\mathcal{T}^* \leftarrow \lambda_\mathcal{T}^{(K)}$

4. Experiments and Results

In order to first litmus test our approach on a simpler task we begin by evaluating it on a synthetic regression dataset borrowed from Vuorio et al. (2019) and defer further experimentation on more complicated real world datasets to future work. Most meta-learning algorithms have been tested on regression datasets where the covariate distribution is same across all tasks. We thus describe how we suitably modify the original dataset so that there exists a structure over the set of covariate distributions across tasks. This enables us to fairly evaluate our method against the baselines in the proposed setting.

We compare our algorithm with the popular gradient based meta-learning approach MAML introduced by Finn et al. (2017). Additionally, we also chose as baselines Amortized MAML (Ravi & Beatson, 2018) and M-MAML (Vuorio et al., 2019) which are exemplars of the Bayesian and task-specific initialization type approaches respectively. These methods either model the task-parameters as Bayesian random variables (former) or adapt the parameters of the optimizer based on the input dataset (latter) and hence warrant a close comparison with our proposed methodology.

Model	*sine*	*sine-quad-linear*	*five*
MAML	0.05	1.27	1.69
Amortized MAML	0.07	1.39	1.13
M-MAML	0.04	0.59	0.93
Ours	**0.008**	**0.39**	**0.89**

Table 1. Comparison of post adaptation test performance (MSE loss) on three regression datasets when the true hypothesis class depends on the covariate distribution.

Model	*sine*	*sine-quad-linear*	*five*
MAML	0.04	1.15	1.73
Amortized MAML	0.07	1.41	1.08
M-MAML	**0.03**	0.51	0.88
Ours	0.04	**0.51**	**0.84**

Table 2. Comparison of post adaptation test performance (MSE loss) on three regression datasets when the true hypothesis class is independent of the covariate distribution.

Dataset The covariate distribution for each task $\mathcal{T}$ is given by a normal $\mathcal{N}(\mu_\mathcal{T}, \sigma_\mathcal{T})$ whose parameters are sampled from a discrete distribution over P pairs $\{(\mu_p, \sigma_p)\}_{p=1}^P$. The pairs $p_1, \ldots p_P$ are fixed in the beginning once they are sampled from a pair of independent uniform priors, $p_i \sim (\mathcal{U}(-10, 10), \ \mathcal{U}(0, 10))$. The parameters of the discrete distribution are sampled from a Dirichlet prior (Dir($\alpha_p = 1$)). Following (Vuorio et al., 2019), the optimal hypothesis for each task is sampled from one of many modalities or hypothesis classes. The hypothesis classes considered are : *sine*, *linear*, *quad*, *transformed-L_1* and *tanh*. For each task, having chosen a hypothesis class, the parameters of the optimal hypothesis (like slope of a linear functions) is chosen based on uniform distributions over the following [3]:

[3]Taken as is from (Vuorio et al., 2019). Re-iterated for the sake of completion.

1. *sine*: $f(x) = A \cdot (sin(w \cdot x) + b)$, with $A \in [0.1, 5.0], w \in [0.5, 2.0]$ and $b \in [0, 2\pi]$.
2. *quad*: $f(x) = A \cdot (x - c)^2 + b$ with $A \in [-0.15, -0.02] \cup [0.02, 0.15], c \in [-3.0, 3.0]$ and $b \in [-3.0, 3.0]$.
3. *linear*: $f(x) = A \cdot x + b$, $A \in [-3, 3], b \in [-3, 3]$.
4. *transformed-L_1*: $f(x) = A \cdot |x - c| + b$, with $A \in [-0.15, -0.02] \cup [0.02, 0.15], c \in [-3.0, 3.0]$ and $b \in [-3.0, 3.0]$.
5. *tanh*: $f(x) = A \cdot tanh(x - c) + b$ with $A \in [-3.0, 3.0], c \in [-3.0, 3.0], b \in [-3.0, 3.0]$.

For each of the true hypothesis classes above the final value of $\mathbf{y}_j^{(i)}$ is generated by adding an independent error term ϵ sampled from a normal distribution with mean 0 and standard deviation of 0.3 *i.e* $\mathbf{y}_j^{(i)} = f(\mathbf{x}_j^{(i)}) + \epsilon$.

We consider two cases, *first* where there exists a relation between the parameters (mean, variance) of the covariate distribution and the optimal hypothesis class chosen for a task and *second* when the optimal hypothesis class is chosen independent of the mean, variance of the covariate distribution. We highlight the results in each case separately.

Case-I: With a specific relation This setting conforms to the case when $z_y \not\perp\!\!\!\perp z_x|\theta$ in Fig. 2. We experiment with three different meta-distributions which are of different complexities owing to the variety (number) of hypothesis classes each of them span. The datasets are listed as follows:

1. *sine*: true hypothesis for each class is given by a sinusoidal function with parameters sampled from distributions mentioned previously. The range for each of the parameters is split into $P = 3$ disjoint sets, each one corresponding to a specific covariate distribution.
2. *sine-quad-linear*: each of the three hypothesis classes are mapped to a specific covariate distribution. Thus, once the parameters of the covariate distribution are sampled based on the discrete prior, the corresponding hypothesis class is also chosen and a task is sampled from it.
3. *five*: similar to the previous case with the distinction that all five hypothesis classes are considered in this dataset *i.e* $P = 5$.

Table 1 highlights the Mean Squared Errors (MSE) achieved by our method and the baselines on the three regression datasets described above. We can see that when there exists a relation between the true hypothesis class and the covariate distribution our approach performs significantly better than other state-of-the-art approaches for regression. Interestingly, improvements are also observed for the *five* dataset which was specifically introduced by Vuorio et al. (2019) for evaluating M-MAML. The Bayesian model Amortized MAML performs poorly since unlike our approach it fails to acknowledge the latent relationship between the covariate distribution and the posterior over the five hypothesis classes.

Case-II: Independent This setting conforms to the case when $z_y \perp\!\!\!\perp z_x|\theta$ in Fig. 2. In Table 2 we demonstrate that the performance of our approach is no worse (if not better) than other methods which assume the independence by default. Once again we experiment with the same three types of meta-distribution mentioned in the previous case.

Implementation The conditional $p(y|x)$ is modeled using a three-layered neural network with hidden sizes $100, 100, 100$. The networks $\gamma_\mu(\cdot;\beta), \gamma_\sigma(\cdot;\beta)$ which take as input the sequence of labeled samples in a dataset $\mathcal{D}_i^S$ are modeled using a common Recurrent Neural Network RNN backbone with output of dimension 28. This can be referred to as the *task-embedding* (Vuorio et al., 2019). The mapping of task-embedding to $\lambda_i^{(0)}$, given by $f_\beta(\cdot)$ is modeled similar to M-MAML as well as the modulation over $\lambda_i^{(0)}$ which gives us the final initialization of the network parameters [4]. Following prior work (Finn & Levine, 2017), a bias-transformation of size 20 was appended to the inputs. The total number of training tasks used were $10,000$ with $m' = m - m' = 5$ samples for the support and query sets for each task. Adam optimizer with an initial learning rate of 0.001 was used to train the meta-learner. Additionally, all KL terms were re-weighted with a weight of 0.01.

5. Conclusion

Cognizant of the fact that the generalization performance of few-shot algorithms depends on a varying number of factors ranging from sample size, hypothesis class complexity to the optimization algorithm, input distribution; in this work we focus our efforts on improving meta-learning algorithms by using the covariate distribution to infer the adapted parameters via a principled Bayesian approach. We begin by deriving ELBO bounds for the hierarchical Bayes formulation and follow it up with a meta-learning algorithm to infer the posterior over the network parameters. Finally, we show some preliminary results on a synthetic regression dataset designed to test the usefulness of our method.

Motivated by the empirical gains observed, we plan to extend our work to more challenging few-shot image classification benchmarks like mini-imagenet (Ravi & Larochelle, 2016) and FC100 (Oreshkin et al., 2018). We also acknowledge that the proposed framework warrants a more rigorous theoretical analysis to understand exactly how inferring the covariate distribution can impact regret bounds (Khodak et al., 2019b;a) in the online few-shot setting or even generalization error bounds in the classical one.

[4] The code for our implementation was in part borrowed from https://github.com/vuoristo/MMAML-Regression

References

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 5049–5059, 2019.

Bertinetto, L., Henriques, J. F., Torr, P. H., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Chapelle, O., Scholkopf, B., and Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Denevi, G., Ciliberto, C., Grazzi, R., and Pontil, M. Learning-to-learn stochastic gradient descent with biased regularization. *arXiv preprint arXiv:1903.10399*, 2019.

Finn, C. and Levine, S. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. *arXiv preprint arXiv:1710.11622*, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 9516–9527, 2018.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Khodak, M., Balcan, M.-F., and Talwalkar, A. Provable guarantees for gradient-based meta-learning. *arXiv preprint arXiv:1902.10644*, 2019a.

Khodak, M., Balcan, M.-F. F., and Talwalkar, A. S. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, pp. 5915–5926, 2019b.

Kim, T., Yoon, J., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836*, 2018.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Oreshkin, B., López, P. R., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pp. 113–124, 2019.

Ravi, S. and Beatson, A. Amortized bayesian meta-learning. 2018.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. 2016.

Saul, L. K. and Roweis, S. T. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of machine learning research*, 4(Jun):119–155, 2003.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pp. 4077–4087, 2017.

Vuorio, R., Sun, S.-H., Hu, H., and Lim, J. J. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pp. 1–12, 2019.

Wang, Y., Yao, Q., Kwok, J., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. In *arXiv: 1904.05046*. 2019.