Probabilistic Mixture of Model-Agnostic Meta-Learners

Prasanna Sattigeri [*†]
psattig@us.ibm.com

Soumya Ghosh [*†]
ghoshso@us.ibm.com

Abhishek Kumar[‡]
abhishek@inductivebias.net

Karthikeyan Natesan Ramamurthy [†]
knatesa@us.ibm.com

Samuel Hoffman [†]
shoffman@ibm.com

Inkit Padhi [†]
inkpad@ibm.com

Youssef Drissi [†]
youssefd@us.ibm.com

1 Introduction

Meta-learning or learning to learn is a promising direction for generalizing machine learning models beyond the specific tasks they were trained on. Meta-learning ideas are at least a few decades old [1, 12], and recent approaches include metric learning [13], soft-attention models [2], memory-augmented networks [11], and learning optimizers [10]. Our focus is on gradient-based meta-learning algorithms that attempt to learn good initializations for a variety of tasks. Examples of this class of algorithms include Model-Agnostic Meta-Learning (MAML) [3] and Reptile [8]. Learning to learn new tasks using only a few training examples is a key concern for recent meta-learning approaches.

Generalizations of gradient-based meta learning methods using probabilistic approaches have been proposed [6, 4, 5], and their advantage is that they can handle the uncertainty and ambiguity arising from few-shot learning in a principled way. In [4, 5] the authors re-interpret MAML as a hierarchical Bayesian model and utilize variational methods for posterior inference over the model parameters. While [5] models the individual task parameters, [4] also provide an efficient scheme for inferring the meta-task parameters. In [6], the authors use a non-parametric variational inference scheme — Stein variational gradient descent to infer the parameters of the hierarchical Bayesian model underlying MAML.

While the approaches mentioned are well-founded and show promising results, they implicitly assume that the set of tasks used for meta-learning are similar and thus a single shared initialization between the tasks is effective. In the real world, however, such homogeneous grouping of tasks rarely exist. Hence we take a step further, and relax this assumption. In particular, we build on the probabilistic view of MAML [4] and develop a mixture of MAML models (see figure 1).

Our proposed Probabilistic Mixture of MAML (ProMix-MAML) approach learns a distinct initialization for each of the mixture components, instead of a single shared initialization. Each mixture component contains a group of similar tasks, and task component memberships are inferred jointly with the other parameters of the model by maximizing a lower bound to the marginal likelihood of the model.

Using synthetic data, we demonstrate the performance of our approach in learning initializations for disparate groups of tasks. We show that our ProMix-MAML approach outperforms MAML and also learns an interpretable mixture of tasks.

[*] Contributed equally.

[†] IBM Research

[‡] Work done at IBM Research. Author is now at Google.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

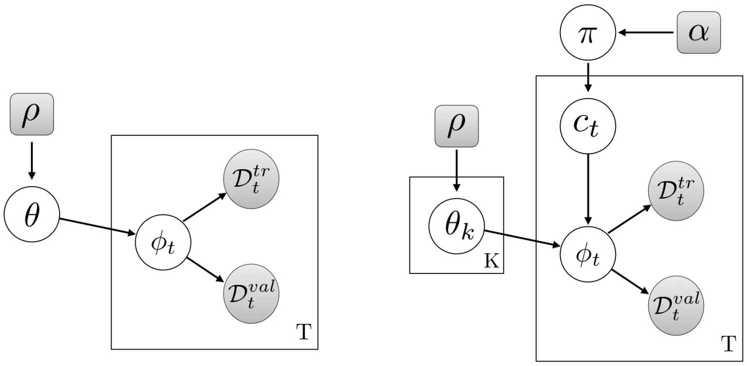

Figure 1: Graphical models for: (a) Left: Probabilistic MAML (variation of [4]), (b) Right: The proposed Probabilistic Mixture of MAML (ProMix-MAML).

2 Probabilitic Mixture of MAML for meta learning from heterogeneous tasks

The generative model can be described as follows. We endow each task t with a categorical indicator variable c_t which indicates the component membership for the task,

$$c_t \mid \pi \sim \text{Cat}(\pi), \quad \pi \mid \alpha \sim \text{Dirichlet}(\alpha) \tag{1}$$

Task specific parameters, ϕ_t are then drawn,

$$\phi_t \mid c_t, \{\theta_k\}_{k=1}^{K} \sim p(\phi_t \mid \theta_{c_t}), \quad \theta_k \sim p(\theta_k \mid \rho), \forall k \tag{2}$$

Finally, the data $\mathcal{D}_t^{tr} = \{x_{t,n}, y_{t,n}\}_{n=1}^{N_{train}}$, $\mathcal{D}_t^{val} = \{x_{t,m}, y_{t,m}\}_{m=1}^{N_{val}}$ is conditionally independent given ϕ_t,

$$p(\mathcal{D}_t^{tr}, \mathcal{D}_t^{val} \mid \phi_t) = \prod_{n=1}^{N_{train}} p(x_n \mid y_n, \phi_t) \prod_{m=1}^{N_{val}} p(x_m \mid y_m, \phi_t) \tag{3}$$

The conditional independencies are summarized in Figure 1, and the joint distribution is given by,

$$\begin{aligned} p(\mathcal{D}_t^{tr}, \mathcal{D}_t^{val}, \{c_t, \phi_t\}_{t=1}^{T}, \{\theta_k\}_{k=1}^{K}, \pi \mid \alpha, \rho) = p(\pi \mid \alpha) \prod_{k=1}^{K} p(\theta_k \mid \rho) \prod_{t=1}^{T} p(\mathcal{D}_t^{tr} \mid \phi_t) p(\mathcal{D}_t^{val} \mid \phi_t) \\ \prod_{k=1}^{K} \{p(c_t = k \mid \pi) p(\phi_t \mid \theta_k)\}^{\mathbf{1}_{c_t = k}} \end{aligned} \tag{4}$$

Both $p(\theta_k \mid \rho)$ and $p(\phi_t \mid \theta_k)$ are modeled as Gaussians with diagonal covariances and means specified by ρ and θ_k. For the synthetic regression problems considered in this paper, we use Gaussian likelihoods $p(\mathcal{D}_t^{tr} \mid \phi_t) = \prod_{n=1}^{N_{train}} \mathcal{N}(y_{t,n} \mid g_{\phi_t}(x_{t,n}), \sigma^2)$, where g is a network parameterized by ϕ_t, $p(\mathcal{D}_t^{val} \mid \phi_t)$ is analogously defined. For the synthetic experiments, we fix σ^2 to the true noise variance.

3 Inference

We rely on variational inference to infer both the task specific parameters ϕ_t and c_t, as well as cluster specific meta-parameters, θ_k. We assume the variational approximation to factorize as follows,

$$q(\{\theta_k\}_{k=1}^{K}, \{\phi_t, c_t\}_{t=1}^{T} \mid \lambda) = \prod_{k=1}^{K} \mathcal{N}(\theta_k \mid \mu_k, \Lambda_k) \prod_{t=1}^{T} q(\phi_t, c_t \mid \lambda_t, \mathcal{D}_t^{tr}), \tag{5}$$

where Λ_k is constrained to be a diagonal matrix, and λ denotes the collection of all variational parameters. Note that while the posterior factorizes between the meta and task specific parameters,

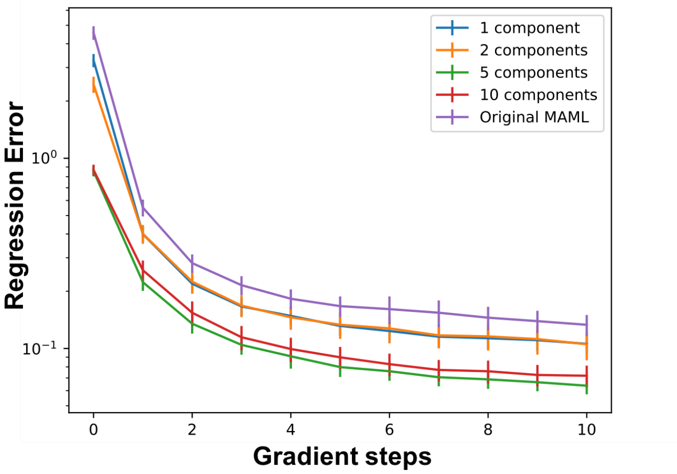

Figure 2: Regression error for varying number of components and increasing number of gradient steps. ProMix-MAML with 5 components preforms the best. The initial weights (at gradient step = 0) obtained from the ProMix-MAML also perform better when compared to the plain MAML.

we find it useful to maintain structure between the task specific parameters ϕ_t and c_t, $q(\phi_t, c_t \mid \lambda_t, \mathcal{D}_t^{tr}) = q(c_t \mid \lambda_{c_t})q(\phi_t \mid c_t, \mathcal{D}_t^{tr}, \lambda_{\phi_t})$, where $\lambda_t = \{\lambda_{\phi_t}, \lambda_{c_t}\}$. We use a categorical distribution for $q(c_t \mid \lambda_{c_t}) = \text{Categorical}(c_t \mid \lambda_{c_t})$, and conditioned on c_t and the training data $\mathcal{D}_t^{tr}$, we define,

$$q(\phi_t|D^{tr}, c_t = k, \lambda_{\phi_t}) = \mathcal{N}(\mu_k - \nabla_{\mu_k} L(D^{tr}), \Psi), \tag{6}$$

where Ψ is again constrained to a diagonal matrix. For the global mixture weights π, instead of inferring a full distribution we only infer the MAP estimate. We use stochastic gradient variational Bayes to optimize the lower bound to the marginal likelihood of the model,

$$\begin{aligned} &p(\{\mathcal{D}_t^{tr}, \mathcal{D}_t^{val}\}_{t=1}^T \mid \pi, \alpha, \rho) \geq \mathcal{L}(\pi, \lambda; \rho, \alpha) \\ &= \ln p(\pi \mid \alpha) + \mathbb{E}_{q_\lambda}[\ln p(\{\mathcal{D}_t^{tr}, \mathcal{D}_t^{val}, c_t, \phi_t\}_{t=1}^T, \{\theta_k\}_{k=1}^K \mid \rho)] + \mathbb{H}[q(\{\theta_k\}_{k=1}^K, \{\phi_t, c_t\}_{t=1}^T \mid \lambda)] \end{aligned} \tag{7}$$

4 Experiments

Synthetic data. We present results on synthetic regression tasks to illustrate the working of the proposed ProMix-MAML approach. Each task corresponds to few-shot regression of a sinusoid function, with a different phase and amplitude. Following [3], we generated tasks by uniformly sampling amplitudes in the range $[0.1, 5.0]$ and phase between $[0, \pi]$. For each task t, the datapoints $\{x_{t,n}, y_{t,n}\}_{n=1}^{N_{train}+N_{test}}$ are sampled uniformly between $[-5.0, 5.0]$.

The architecture for the regressor neural network is identical to that in [3] comprising of 2 hidden layers with 40 units each. The K mixture components in ProMix-MAML are K independently initialized regressors. Note that both input and output are scalars. We use one-gradient step with 10 samples for training both MAML and ProMix-MAML and fix the step size $\alpha = 0.01$. Figure 2 shows the error for varying number of mixture components and gradient steps.

Our ProMix-MAML approach outperforms MAML as the number of components increase, until the number of mixture components is 5 at which point performance saturates. To gain further insights into the results, in Figure 3 (see appendix), we visualize the tasks assigned to various mixture components. We find that sinusoids with similar amplitude ***or*** phase are typically assigned to the same mixture component. Our next steps include careful validation of the proposed model on real data, as well as the application of ProMix-MAML for few-shot learning with high dimensional images [7], and in reading comprehension applications [9].

References

[1] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche opérationnelle, 1990.

[2] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in neural information processing systems*, pages 1087–1098, 2017.

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

[4] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

[5] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

[6] Taesup Kim, Jaesik Yoon, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836*, 2018.

[7] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[8] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR, abs/1803.02999*, 2018.

[9] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[10] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[11] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

[12] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[13] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

A Visualizing Clustering of Tasks in ProMix-MAML

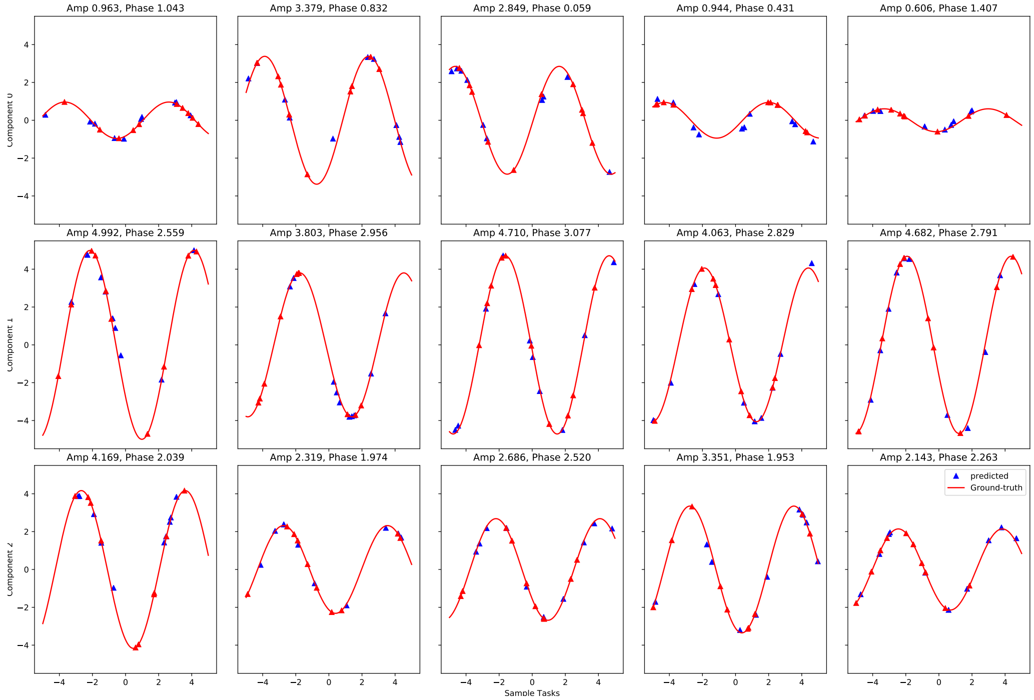

Figure 3: Clustering of tasks when the task distribution comprises of sinusoids with varying amplitude and phase. Number of mixture components is set to 3. It can be seen that sinusoids with similar amplitude or phase are assigned to same mixture component. The first component (top row) groups together sinusoids with similar phase. The second and third components both contain sinusoids of similar phases but while component two groups together sinusoids of larger magnitude, component three focuses on smaller amplitude sinusoids. The Red line is the ground truth, red triangles are training data points sampled from the ground truth, blue triangles are predictions at novel test points.

B Entropy of $q(\phi_t, c_t \mid \lambda_\phi, \lambda_c)$

$$\begin{aligned}
\mathbb{H}[q] &= \int q(\phi_t, c_t \mid \lambda_\phi, \lambda_c)\ln q(\phi_t, c_t \mid \lambda_\phi, \lambda_c) d\phi_t dc_t \\
&= \int \sum_k q(\phi_t \mid c_t = k) q(c_t = k) \ln q(\phi_t \mid c_t = k) q(c_t = k) d\phi_t \\
&= \int \sum_k q(\phi_t \mid c_t = k) q(c_t = k) \ln q(\phi_t \mid c_t = k) d\phi_t + \int \sum_k q(\phi_t \mid c_t = k) q(c_t = k) \ln q(c_t = k) d\phi_t
\end{aligned} \tag{8}$$

Denoting $q(c_t = k) = r_{tk}$, we have,

$$\begin{aligned}
&= \int \sum_k q(\phi_t \mid c_t = k) r_{tk} \ln q(\phi_t \mid c_t = k) d\phi_t + \int \sum_k q(\phi_t \mid c_t = k) r_{tk} \ln r_{tk} d\phi_t \\
&= \int \sum_k q(\phi_t \mid c_t = k) r_{tk} \ln q(\phi_t \mid c_t = k) d\phi_t + \sum_k r_{tk} \ln r_{tk} \\
&= \sum_k r_{tk} \int q(\phi_t \mid c_t = k) \ln q(\phi_t \mid c_t = k) d\phi_t + \sum_k r_{tk} \ln r_{tk} \\
&= \sum_k r_{tk} \mathbb{H}(q(\phi_t \mid c_t = k)) + \sum_k r_{tk} \ln r_{tk},
\end{aligned} \tag{9}$$

Since $q(\phi_t \mid c_t = k)$ is just a Gaussian, so $\mathbb{H}(q(\phi_t \mid c_t = k))$ is available in closed form, allowing us to compute the entire expression above in closed form.