Subspace Learning for Effective Meta-Learning

Weisen Jiang◇†, James T. Kwok†, Yu Zhang◇

◇Southern University of Science and Technology †Hong Kong University of Science and Technology

Introduction

- To improve data efficiency, **meta-learning** extracts meta-knowledge from historical tasks to accelerate learning unseen tasks. One representative algorithm MAML [1] learns a globally-shared initialization for all tasks.
- However, real-world environments are usually complex, where task models are diverse and a common meta-model is insufficient to capture all meta-knowledge.
- Recently, TSA-MAML [2] based on k-means clustering learns an intialization for tasks in each cluster. However, task model parameters may lie in a subspace mixture. In a linear regression setting where parameters are from a single subspace, previous work [3, 4] uses a moment-based estimator to recover the underlying subspace. However, extension to nonlinearity (such as deep networks) is difficult.
- In this paper, we propose a **model-agnostic** algorithm (called **MUSML**) to learn **a subspace mixture** for constructing task model parameters. For each task, the base learner builds a task model from each subspace, then the meta-learner updates the subspace bases by minimizing a weighted validation loss of the task models.

Our Approach

- Notations:
  - $\mathcal{T}$ is a collection of tasks for meta-training. Each task $\tau \in \mathcal{T}$ contains a training set $\mathcal{D}_\tau^{tr}$ and a validation set $\mathcal{D}_\tau^{vl}$.
  - $\mathcal{L}(\mathcal{D};\mathbf{w}) = \frac{1}{|\mathcal{D}|}\sum_{(\mathbf{x},y)\in\mathcal{D}} \ell(f(\mathbf{x};\mathbf{w}),y)$ is the loss on $\mathcal{D}$ for model $f(\cdot;\mathbf{w})$.
  - Assume task model parameters lie in a subspace mixture $\{\mathbb{S}_1,\dots,\mathbb{S}_K\}$. Let $\mathbf{S}_k \in \mathbb{R}^{d\times m}$ be the basis of $\mathbb{S}_k$, then $\{\mathbf{S}_1,\dots,\mathbf{S}_K\}$ are meta-parameters to be learned.
- **Base learner**: In each subspace $\mathbb{S}_k$, we search for a linear combination to form the task model $\mathbf{w}_\tau = \mathbf{S}_k\mathbf{v}^\star_{\tau,k}$:

$$\mathbf{v}^\star_{\tau,k} = \arg\min_{\mathbf{v}_\tau\in\mathbb{R}^m} \mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_\tau).$$

  - When $\mathcal{L}(\mathcal{D};\mathbf{w})$ is convex, use convex program.
  - In noncovex case, we seek an approximate minimizer $\mathbf{v}_{\tau,k} = \mathbf{v}_{\tau,k}^{(T_{in})}$: $\mathbf{v}_{\tau,k}^{(t'+1)} = \mathbf{v}_{\tau,k}^{(t')} - \alpha\nabla_{\mathbf{v}_{\tau,k}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')})$, for $t'=0,\dots,T_{in}-1$.
- **Meta-learner**:
  - At meta-training, one can assign τ to the subspace with the best training set performance, but such one-hot selection is inefficient for learning meta-parameters as only one subspace is updated at each step.
  - We relax the categorical selection to soft selection and all subspaces can be updated simultaneously. Specifically, let $o_{\tau,k} = \mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,k})$ be the training loss for task τ when the kth subspace (where $k=1,\dots,K$) is used to construct its task model. The meta-learner updates $\{\mathbf{S}_1,\dots,\mathbf{S}_K\}$ by performing one gradient update on the weighted validation loss

$$\mathcal{L}_{vl}(\mathbf{S}_1,\dots,\mathbf{S}_K) \equiv \sum_{k=1}^{K}\frac{\exp(-o_{\tau,k}/\gamma)}{\sum_{k'=1}^{K}\exp(-o_{\tau,k'}/\gamma)}\mathcal{L}(\mathcal{D}_\tau^{vl};\mathbf{S}_k\mathbf{v}_{\tau,k}),$$

  where $\gamma>0$ is the temperature ($\gamma\to 0$, the selection becomes one-hot; $\gamma\to\infty$, the selection becomes uniform).
- At meta-testing, we assign each testing task to the subspace with the lowest training loss.

Proposed Algorithm

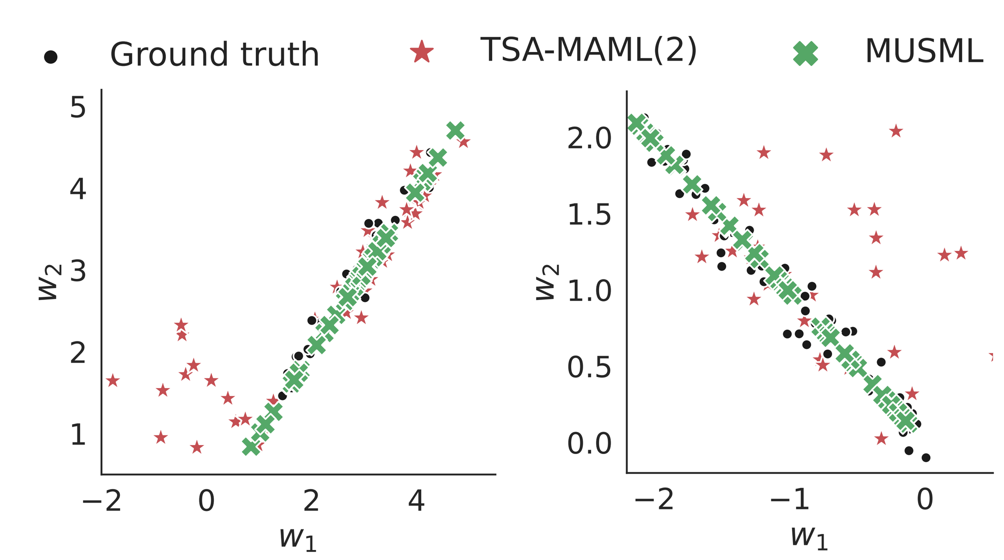
Algorithm 1 MUSML.

Require: stepsize α, $\{\eta_t\}$; #subspaces K, subspace dimension m; $\mathbf{v}^{(0)}$, $\{\gamma_t\}$;

for $t=0,1,\dots,T-1$ **do**
  sample a task τ with $\mathcal{D}_\tau^{tr}$ and $\mathcal{D}_\tau^{vl}$;
  base learner:
  for $k=1,\dots,K$ **do**
    initialize $\mathbf{v}_{\tau,k}^{(0)}=\mathbf{v}^{(0)}$;
    for $t'=0,1,\dots,T_{in}-1$ **do**
      $\mathbf{v}_{\tau,k}^{(t'+1)} = \mathbf{v}_{\tau,k}^{(t')} - \alpha\nabla_{\mathbf{v}_{\tau,k}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_{k,t}\mathbf{v}_{\tau,k}^{(t')})$;
    end for
    $\mathbf{v}_{\tau,k}\equiv\mathbf{v}_{\tau,k}^{(T_{in})}$;
    $o_{\tau,k}=\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_{k,t}\mathbf{v}_{\tau,k})$;
  end for
  meta-learner:
  $\mathcal{L}_{vl}=\sum_{k=1}^{K}\frac{\exp(-o_{\tau,k}/\gamma_t)}{\sum_{k'=1}^{K}\exp(-o_{\tau,k'}/\gamma_t)}\mathcal{L}(\mathcal{D}_\tau^{vl};\mathbf{S}_{k,t}\mathbf{v}_{\tau,k})$;
  $\{\mathbf{S}_{1,t+1},\dots,\mathbf{S}_{K,t+1}\}=\{\mathbf{S}_{1,t},\dots,\mathbf{S}_{K,t}\}-\eta_t\nabla_{\{\mathbf{S}_{1,t},\dots,\mathbf{S}_{K,t}\}}\mathcal{L}_{vl}$;
end for
Return $\mathbf{S}_{1,T},\dots,\mathbf{S}_{K,T}$.

Few-shot Regression

1. Synthetic data: (i) a nonlinear model $f(x;\mathbf{w}_\tau)=\exp(0.1w_{\tau,1}x)+w_{\tau,2}|\sin(x)|$ in which $\mathbf{w}_\tau=[w_{\tau,1};w_{\tau,2}]$ is randomly sampled from one of the two subspaces (*Line-A* and *Line-B*). (ii) samples are generated by $y=f(x;\mathbf{w}_\tau)+0.05\times\mathcal{N}(0,1)$.
2. *Pose* data: a real-world pose prediction dataset.



(a): Tasks from *Line-A* (b): Tasks from *Line-B*

Figure 1: Visualization of task model parameters on synthetic data. *MUSML can discover the underlying subspaces.*

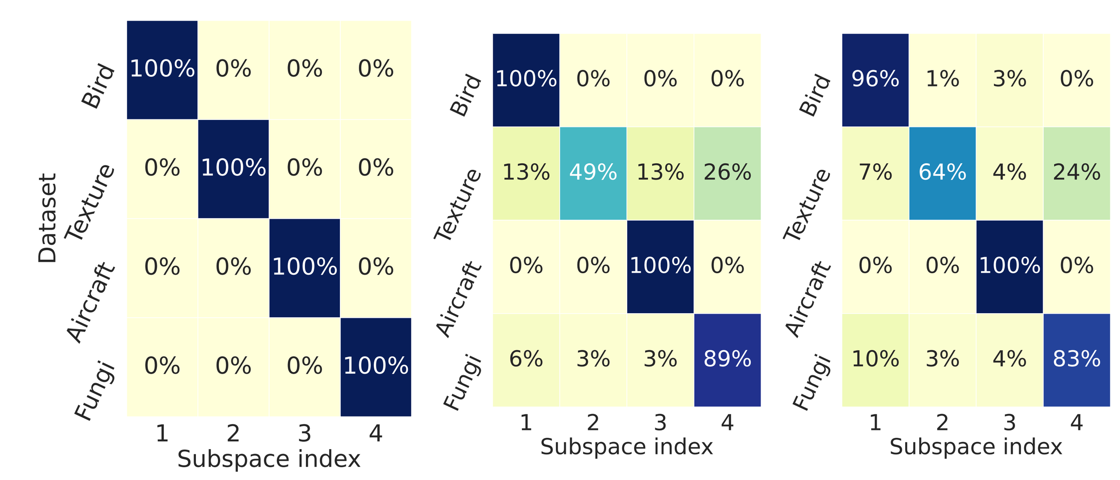
Table 1: Meta-testing MSE on synthetic data and *Pose* data. *MUSML performs the best.*

	Synthetic data	*Pose* data
MAML	0.74 ± 0.03	5.39 ± 1.31
MR-MAML	-	2.26 ± 0.09
BMG	0.67 ± 0.03	2.16 ± 0.15
DPMM	0.56 ± 0.09	1.99 ± 0.08
HSML	0.49 ± 0.10	2.04 ± 0.13
ARML	0.60 ± 0.07	2.21 ± 0.15
TSA-MAML	0.58 ± 0.10	1.96 ± 0.07
MUSML	$\mathbf{0.07}\pm0.01$	$\mathbf{1.83}\pm0.05$

Few-shot Classification

Table 2: Accuracies of 5-way 5-shot classification on meta-datasets. *MUSML is more accurate than both structured and unstructured meta-learning methods.*

	Meta-Dataset-BTAF	*Meta-Dataset-ABF*	*Meta-Dataset-CIO*
MAML	57.78	63.86	74.46
ProtoNet	62.29	65.62	76.51
ANIL	58.57	64.43	74.61
BMG	60.10	65.80	77.46
DPMM	63.00	66.26	76.63
TSA-MAML	63.20	68.17	76.89
HSML	62.39	64.17	75.54
ARML	63.95	64.52	76.12
TSA-ProtoNet	63.57	68.77	77.27
MUSML	**66.18**	**71.10**	**77.83**



(a): Meta-training. (b): Meta-validation. (c): Meta-testing.

Figure 2: Task assignment to the learned subspaces in 5-way 5-shot setting on *Meta-Dataset-BTAF*. *As can be seen, MUSML can discover the task structure.*

Table 3: Accuracies of 5-way 5-shot classification on meta-datasets. *MUSML is beneficial for other meta-learning algorithms.*

	Meta-Dataset-BTAF	*Meta-Dataset-ABF*	*Meta-Dataset-CIO*
Meta-SGD	58.93	64.19	75.95
MUSML-SGD	65.72	69.15	77.48
Meta-Curvature	50.02	64.51	76.13
MUSML-Curvature	**66.10**	**69.23**	**77.96**

Table 4: Accuracies of cross-domain 5-way 5-shot classification (*Meta-Dataset-BTAF* → *Meta-Dataset-CIO*). *MUSML is also effective on unseen domains.*

MAML	ProtoNet	ANIL	BMG	DPMM	TSA-MAML	HSML	ARML	TSA-ProtoNet	MUSML
64.25	66.13	65.19	66.98	66.73	66.85	65.18	65.37	66.92	**67.41**

Summary

1. We study meta-learning in complex environments, where task models are diverse.
2. We proposed a model-agnostic algorithm (called MUSML) to learn a subspace mixture for constructing task model parameters.
3. Each subspace can be viewed as a type of meta-knowledge.
4. Experimental results confirm the effectiveness of the proposed MUSML.

Reference

[1] Finn, C., Abbeel, P., and Levine, S. Model-agnostic metalearning for fast adaptation of deep networks. In *ICML* 2017.

[2] Zhou, P., Zou, Y., Yuan, X., Feng, J., Xiong, C., and Hoi, S. Task similarity aware meta learning: Theory-inspired improvement on MAML. In *UAI* 2021.

[3] Kong, W., Somani, R., Song, Z., Kakade, S., and Oh, S. Meta-learning for mixed linear regression. In *ICML* 2020.

[4] Tripuraneni, N., Jin, C., and Jordan, M. Provable metalearning of linear representations. In *ICML* 2021.