How Important is the Train-Validation Split in Meta-Learning?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Meta-learning aims to perform fast adaptation on a new task through learning a "prior" from multiple existing tasks. A common practice in meta-learning is to perform a *train-validation split* where the prior adapts to the task on one split of the data, and the resulting predictor is evaluated on another split. Despite its prevalence, the importance of the train-validation split is not well understood either in theory or in practice, particularly in comparison to the more direct *non-splitting* method, which uses all the per-task data for both training and evaluation.

We provide a detailed theoretical study on whether and when the train-validation split is helpful on the linear centroid meta-learning problem, in the asymptotic setting where the number of tasks goes to infinity. We show that the splitting method converges to the optimal prior as expected, whereas the non-splitting method does not in general without structural assumptions on the data. In contrast, if the data are generated from linear models (the realizable regime), we show that both the splitting and non-splitting methods converge to the optimal prior. Further, perhaps surprisingly, our main result shows that the non-splitting method achieves a *strictly better* asymptotic excess risk under this data distribution, even when the regularization parameter and split ratio are optimally tuned for both methods. Our results highlight that data splitting may not always be preferable, especially when the data is realizable by the model. We validate our theories by experimentally showing that the non-splitting method can indeed outperform the splitting method, on both simulations and real meta-learning tasks.

1 Introduction

Meta-learning, also known as "learning to learn", has recently emerged as a powerful paradigm for learning to adapt to unseen tasks (Schmidhuber, 1987). The high-level methodology in meta-learning is akin to how human beings learn new skills, which is typically done by relating to certain prior experience that makes the learning process easier. More concretely, meta-learning does not train one model for each individual task, but rather learns a "prior" model from multiple existing tasks so that it is able to quickly adapt to unseen new tasks. Meta-learning has been successfully applied to many real problems, including few-shot image classification (Finn et al., 2017; Snell et al., 2017), hyper-parameter optimization (Franceschi et al., 2018), low-resource machine translation (Gu et al., 2018) and short event sequence modeling (Xie et al., 2019).

A common practice in meta-learning algorithms is to perform a *sample splitting*, where the data within each task is divided into a *training split* which the prior uses to adapt to a task-specific predictor, and a *validation split* on which we evaluate the performance of the task-specific predictor (Nichol et al., 2018; Rajeswaran et al., 2019; Fallah et al., 2020; Wang et al., 2020a). This sample splitting is believed to be crucial, as it matches the evaluation criterion at meta-test time,

Submitted to the 4th Workshop on Meta-Learning at NeurIPS 2020, Vancouver, Canada. Do not distribute.

where we perform adaptation on training data from a new task but evaluate its performance on unseen data from the same task. However, despite the aforementioned importance, performing the train-validation split has a potential drawback from the data efficiency perspective — Because of the split, neither the training nor the evaluation stage is able to use all the available per-task data. Further, performing the train-validation split is also not the only option in practice: there exist algorithms such as Reptile (Nichol & Schulman, 2018) and Meta-MinibatchProx (Zhou et al., 2019) that can instead use all the per-task data for training the task-specific predictor and also perform well empirically on benchmark tasks. These algorithms modify the loss function in the outer loop so that the training loss no longer matches the meta-test loss, but may have the advantage in terms of data efficiency for the overall problem of learning the best prior. So far it is theoretically unclear how these two approaches (with/without train-validation split) compare with each other, which motivates us to ask the following

Question: Is the train-validation split *necessary* and *optimal* in meta-learning?

In this paper, we perform a detailed theoretical study on the importance of the train-validation split. We consider the linear centroid meta-learning problem (Denevi et al., 2018b), where for each task we learn a linear predictor that is close to a common centroid in the inner loop, and find the best centroid in the outer loop (see Section 2 for the detailed problem setup). We compare two outer-loop algorithms of either performing the train-validation split (the *train-val method*) or using all the per-task data for both training and evaluation (the *train-train method*). Specifically, we compare the two methods when the number of tasks T is large, and examine if and how fast they converge to the (properly defined) best centroid at meta-test time. We summarize our contributions as follows:

- On the linear centroid meta-learning problem, we show that the train-validation split is necessary in the general agnostic setting: As $T \to \infty$, the train-val method converges to the optimal centroid for test-time adaptation, whereas the train-train method does not without further assumptions on the tasks (Section 3). The convergence of the train-val method is expected since its (population) training loss is equivalent to the meta-test time loss, whereas the non-convergence of the train-train method is because these two losses are not equivalent in general.
- Our main theoretical contribution is to show that the train-validation split is not necessary and even non-optimal, in the perhaps more interesting regime when there are structural assumptions on the tasks: When the data are generated from noiseless linear models, both the train-val and train-train methods converge to the common best centroid, and the train-train method achieves strictly better (asymptotic) estimation error and test loss than the train-val method (Section 4). This is in stark contrast with the agnostic case, and suggests that data efficiency may indeed be more important when the tasks have a nice structure.
- We perform meta-learning experiments on simulations and benchmark few-shot image classification tasks, showing that the train-train method can consistently outperform the train-val method (Section 5 & Appendix A). This validates our theories and presents empirical evidence that sample splitting may not be crucial; methods that utilize the per-task data more efficiently may be preferred.

1.1 Related work

Meta-learning and representation learning theory Baxter (2000) provided the first theoretical analysis of meta-learning via covering numbers, and Maurer et al. (2016) improved the analysis via Gaussian complexity techniques. Another recent line of theoretical work analyzed gradient-based meta-learning methods (Denevi et al., 2018a; Finn et al., 2019; Khodak et al., 2019) and showed guarantees for convex losses by using tools from online convex optimization. Saunshi et al. (2020) proved the success of Reptile in a one-dimensional subspace setting. Wang et al. (2020b) compared the performance of train-train and train-val methods for learning the learning rate. Denevi et al. (2018b) proposed the linear centroid model studied in this paper, and provided generalization error bounds for train-val method; the bounds proved also hold for train-train , so are not sharp enough to compare the two algorithms. On the representation learning end, Du et al. (2020); Tripuraneni et al. (2020) showed that ERM can successfully pool data across tasks to learn the representation. Yet the focus is on the accurate estimation of the common representation, not on the fast adaptation of the learned prior. Lastly, we remark that there are analyses for other representation learning schemes (McNamara & Balcan, 2017; Galanti et al., 2016; Alquier et al., 2016). **Multi-task learning** Multi-task learning also exploits structures and similarities across multiple tasks. The earliest idea dates back to Caruana (1997); Thrun & Pratt (1998); Baxter (2000), initially in connections to

neural network models. They further motivated other approaches using kernel methods (Evgeniou et al., 2005; Argyriou et al., 2007) and multivariate linear regression models with structured sparsity (Liu et al., 2009, 2015). More recent advances on deep multi-task learning focus on learning shared intermediate representations across tasks Ruder (2017). These multi-task learning approaches usually minimize the joint empirical risk over all tasks, and the models for different tasks are enforced to share a large amount of parameters. In contrast, meta-learning only requires the models to share the same "prior", which is more flexible than multi-task learning.

2 Preliminaries

Linear centroid meta-learning We instantiate our study on the *linear centroid meta-learning problem* (also known as learning to learn around a common mean (Denevi et al., 2018b)), where we wish to learn a task-specific linear predictor $\mathbf{w}_t \in \mathbb{R}^d$ in the inner loop for each task t, and learn a "centroid" $\mathbf{w}_0$ in the outer loop that enables fast adaptation to $\mathbf{w}_t$ within each task:

Find the best centroid $\mathbf{w}_0 \in \mathbb{R}^d$ for adapting to a linear predictor $\mathbf{w}_t$ on each task t.

Formally, we assume that we observe training data from $T \geq 1$ tasks, where for each task index t we sample a task p_t (a distribution over $\mathbb{R}^d \times \mathbb{R}$) from some distribution of tasks Π, and observe n examples $(\mathbf{X}_t, \mathbf{y}_t) \in \mathbb{R}^{n \times d} \times \mathbb{R}^n$ that are drawn i.i.d. from p_t:

$$p_t \sim \Pi, \quad (\mathbf{X}_t, \mathbf{y}_t) = \{(\mathbf{x}_{t,i}, y_{t,i})\}_{i=1}^n \quad \text{where} \quad (\mathbf{x}_{t,i}, y_{t,i}) \overset{\text{iid}}{\sim} p_t. \tag{1}$$

We do not make further assumptions on (n, d); in particular, we allow the underdetermined setting $n \leq d$, in which there exists (one or many) interpolators $\widetilde{\mathbf{w}}_t$ that perfectly fit the data: $\mathbf{X}_t \widetilde{\mathbf{w}}_t = \mathbf{y}_t$.

Inner loop: Ridge solver with biased regularization towards the centroid Our goal in the inner loop is to find a linear predictor $\mathbf{w}_t$ that fits the data in task t while being close to the given "centroid" $\mathbf{w}_0 \in \mathbb{R}^d$. We instantiate this through ridge regression (i.e. linear regression with L_2 regularization) where the regularization biases $\mathbf{w}_t$ towards the centroid. Formally, for any $\mathbf{w}_0 \in \mathbb{R}^d$ and any dataset $(\mathbf{X}, \mathbf{y})$, we consider the algorithm

$$\mathcal{A}_\lambda(\mathbf{w}_0; \mathbf{X}, \mathbf{y}) := \arg\min_{\mathbf{w}} \frac{1}{n} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w} - \mathbf{w}_0\|^2 = \mathbf{w}_0 + \left(\mathbf{X}^\top \mathbf{X} + n\lambda \mathbf{I}_d\right)^{-1} \mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w}_0),$$

where $\lambda > 0$ is the regularization strength (typically a tunable hyper-parameter).

Outer loop: finding the best centroid In the outer loop, our goal is to find the best centroid $\mathbf{w}_0$. The standard approach in meta-learning is to perform a *train-validation split*, that is, (1) execute the inner solver on a first split of the task-specific data, and (2) evaluate the loss on a second split, yielding a function of $\mathbf{w}_0$ that we can optimize. This two-stage procedure can be written as

$$\text{Compute } \mathbf{w}_t(\mathbf{w}_0) = \mathcal{A}_\lambda(\mathbf{w}_0; \mathbf{X}_t^{\mathsf{train}}, \mathbf{y}_t^{\mathsf{train}}), \quad \text{and} \quad \text{evaluate } \left\|\mathbf{y}_t^{\mathsf{val}} - \mathbf{X}_t^{\mathsf{val}} \mathbf{w}_t(\mathbf{w}_0)\right\|^2.$$

where $(\mathbf{X}_t^{\mathsf{train}}, \mathbf{y}_t^{\mathsf{train}}) = \{(\mathbf{x}_{t,i}, y_{t,i})\}_{i=1}^{n_1}$ and $(\mathbf{X}_t^{\mathsf{val}}, \mathbf{y}_t^{\mathsf{val}}) = \{(\mathbf{x}_{t,i}, y_{t,i})\}_{i=n_1+1}^{n}$ are two disjoint splits of the per-task data $(\mathbf{X}_t, \mathbf{y}_t)$ of size (n_1, n_2), with $n_1 + n_2 = n$. Written concisely, this is to consider the "split loss"

$$\ell_t^{\mathsf{tr\text{-}val}}(\mathbf{w}_0) := \frac{1}{2n_2} \left\|\mathbf{y}_t^{\mathsf{val}} - \mathbf{X}_t^{\mathsf{val}} \mathcal{A}_\lambda(\mathbf{w}_0; \mathbf{X}_t^{\mathsf{train}}, \mathbf{y}_t^{\mathsf{train}})\right\|^2. \tag{2}$$

In this paper, we will also consider an alternative version, where we do not perform the train-validation split, but instead use *all the per-task data for both training and evaluation*. Mathematically, this is to look at the "non-split loss"

$$\ell_t^{\mathsf{tr\text{-}tr}}(\mathbf{w}_0) := \frac{1}{2n} \|\mathbf{y}_t - \mathbf{X}_t \mathcal{A}_\lambda(\mathbf{w}_0; \mathbf{X}_t, \mathbf{y}_t)\|^2. \tag{3}$$

Our overall algorithm is to solve the empirical risk minimization (ERM) problem on the T observed tasks, using either one of the two losses above:

$$\begin{aligned} &\widehat{L}_T^{\mathsf{tr\text{-}val}}(\mathbf{w}_0) := \frac{1}{T} \sum_{t=1}^T \ell_t^{\mathsf{tr\text{-}val}}(\mathbf{w}_0) \quad \text{and} \quad \widehat{L}_T^{\mathsf{tr\text{-}tr}}(\mathbf{w}_0) := \frac{1}{T} \sum_{t=1}^T \ell_t^{\mathsf{tr\text{-}tr}}(\mathbf{w}_0), \\ &\widehat{\mathbf{w}}_{0,T}^{\{\mathsf{tr\text{-}val}, \mathsf{tr\text{-}tr}\}} := \arg\min_{\mathbf{w}_0} \widehat{L}_T^{\{\mathsf{tr\text{-}val}, \mathsf{tr\text{-}tr}\}}(\mathbf{w}_0). \end{aligned} \tag{4}$$

Let $L^{\{\mathsf{tr\text{-}val}, \mathsf{tr\text{-}tr}\}}(\mathbf{w}_0) := \mathbb{E}_{p_t \sim \Pi, (\mathbf{X}_t, \mathbf{y}_t) \sim p_t} \left[\ell_t^{\{\mathsf{tr\text{-}val}, \mathsf{tr\text{-}tr}\}}(\mathbf{w}_0)\right]$ be the population risks.

(Meta-)Test time The meta-test time performance of any meta-learning algorithm is a joint function of the (learned) centroid $\mathbf{w}_0$ and the inner algorithm Alg. Upon receiving a new task $p_{T+1} \sim \Pi$ and training data $(\mathbf{X}_{T+1}, \mathbf{y}_{T+1}) \in \mathbb{R}^{n\times d} \times \mathbb{R}^n$, we run the inner loop Alg with prior $\mathbf{w}_0$ on the training data, and evaluate it on an (unseen) test example $(\mathbf{x}', y') \sim p_{T+1}$:

$$L^{\mathsf{test}}(\mathbf{w}_0; \mathsf{Alg}) := \mathbb{E}_{p_{T+1}\sim\Pi}\mathbb{E}_{(\mathbf{X}_{T+1},\mathbf{y}_{T+1}),(\mathbf{x}',y')\overset{\text{iid}}{\sim}p_{T+1}}\left[\frac{1}{2}\left(\mathbf{x}'^{\top}\mathsf{Alg}(\mathbf{w}_0; \mathbf{X}_{T+1}, \mathbf{y}_{T+1}) - y'\right)^2\right].$$

Additionally, for both train-val and train-train methods, we need to ensure that the inner loop used for meta-testing is exactly the same as that used in meta-training. Therefore, the meta-test performance for the train-val and train-train methods above should be evaluated as $L^{\mathsf{test}}_{\lambda,n_1}(\widehat{\mathbf{w}}^{\mathsf{tr\text{-}val}}_{0,T}) := L^{\mathsf{test}}(\widehat{\mathbf{w}}^{\mathsf{tr\text{-}val}}_{0,T}; \mathcal{A}_{\lambda,n_1})$, and $L^{\mathsf{test}}_{\lambda,n}(\widehat{\mathbf{w}}^{\mathsf{tr\text{-}tr}}_{0,T}) := L^{\mathsf{test}}(\widehat{\mathbf{w}}^{\mathsf{tr\text{-}tr}}_{0,T}; \mathcal{A}_{\lambda,n})$, where $\mathcal{A}_{\lambda,m}$ denotes the ridge solver with regularization strength $\lambda > 0$ on $m \le n$ data points. Finally, we let

$$\mathbf{w}_{0,\star}(\lambda; n) = \arg\min_{\mathbf{w}_0} L^{\mathsf{test}}_{\lambda,n}(\mathbf{w}_0) \tag{5}$$

denote the best centroid if the inner loop uses $\mathcal{A}_{\lambda,n}$. The performance of the train-val algorithm $\widehat{\mathbf{w}}^{\mathsf{tr\text{-}val}}_{0,T}$ should be compared against $\mathbf{w}_{0,\star}(\lambda, n_1)$, whereas the train-train algorithm $\widehat{\mathbf{w}}^{\mathsf{tr\text{-}tr}}_{0,T}$ should be compared against $\mathbf{w}_{0,\star}(\lambda, n)$.

2.1 Task-abundant setting through asymptotic analysis

In this paper we are interested in the *task-abundant* setting where we fix some finite (d, n) and let T be very large. We analyze such a task-abundant setting through the asymptotic analysis framework, that is, examine the limiting properties of the algorithm (e.g. $\widehat{\mathbf{w}}^{\{\mathsf{tr\text{-}val},\mathsf{tr\text{-}tr}\}}_{0,T}$) as $T \to \infty$. Here we set up the basic notation of asymptotic analysis required in this paper, and refer the readers to (Van der Vaart, 2000) for a detailed treatment.

Asymptotic rate of estimation & excess risk Let L be any population risk with minimizer $\mathbf{w}_{0,\star}$ (which we assume is unique), $\widehat{L}_T$ be the empirical risk on the observed data from T tasks, and $\widehat{\mathbf{w}}_{0,T}$ be the minimizer of $\widehat{L}_T$ (i.e. the ERM). We say that $\widehat{\mathbf{w}}_{0,T}$ is **consistent** if $\widehat{\mathbf{w}}_{0,T} \to \mathbf{w}_{0,\star}$ in probability as $T \to \infty$. In typical scenarios, for consistent ERMs, the limiting distribution of $\widehat{\mathbf{w}}_{0,T}$ is asymptotically normal with a known covariance matrix, as is characterized in the following classical result (see, e.g. (Van der Vaart, 2000, Section 5.3) and also (Liang, 2016)).

Proposition 1 (Asymptotic normality and excess risk of ERMs)**.** *Suppose the ERM $\widehat{\mathbf{w}}_{0,T}$ is consistent and certain regularity conditions hold, then we have*

$$\sqrt{T}\cdot(\widehat{\mathbf{w}}_{0,T} - \mathbf{w}_{0,\star}) \xrightarrow{d} \mathsf{N}\left(\mathbf{0}, \nabla^2 L(\mathbf{w}_{0,\star})^{-1}\mathrm{Cov}(\nabla\ell_t(\mathbf{w}_{0,\star}))\nabla^2 L(\mathbf{w}_{0,\star})^{-1}\right) =: P_{\mathbf{w}},$$
$$T\cdot(L(\widehat{\mathbf{w}}_{0,T}) - L(\mathbf{w}_{0,\star})) \xrightarrow{d} \mathbf{\Delta}^{\top}\nabla^2 L(\mathbf{w}_{0,\star})\mathbf{\Delta} \quad \text{where } \mathbf{\Delta} \sim P_{\mathbf{w}}.$$

where $\xrightarrow{d}$ denotes convergence in distribution and $\ell_t : \mathbb{R}^d \to \mathbb{R}$ is the loss function on a single task.

When this happens, we define the asymptotic rate of estimation (in MSE loss) and asymptotic excess risk $\widehat{\mathbf{w}}_{0,T}$ as those of its limiting distribution:

$$\mathrm{AsymMSE}(\widehat{\mathbf{w}}_{0,T}) := \mathbb{E}_{\mathbf{\Delta}\sim P_{\mathbf{w}}}\left[\|\mathbf{\Delta}\|^2\right] = \mathrm{tr}\left(\nabla^2 L(\mathbf{w}_{0,\star})^{-1}\mathrm{Cov}(\nabla\ell_t(\mathbf{w}_{0,\star}))\nabla^2 L(\mathbf{w}_{0,\star})^{-1}\right)$$
$$\mathrm{AsymExcessRisk}(\widehat{\mathbf{w}}_{0,T}) := \mathbb{E}_{\mathbf{\Delta}\sim P_{\mathbf{w}}}\left[\mathbf{\Delta}^{\top}\nabla^2 L(\mathbf{w}_{0,\star})\mathbf{\Delta}\right] = \mathrm{tr}\left(\nabla^2 L(\mathbf{w}_{0,\star})^{-1}\mathrm{Cov}(\nabla\ell_t(\mathbf{w}_{0,\star}))\right).$$

3 The importance of sample splitting

We begin by analyzing whether the algorithms $\widehat{\mathbf{w}}^{\{\mathsf{tr\text{-}val},\mathsf{tr\text{-}tr}\}}_{0,T}$ defined in (4) converge to the best test-time centroid $\mathbf{w}_{0,\star}(\lambda; n_1)$ or $\mathbf{w}_{0,\star}(\lambda; n)$ (defined (5)) respectively as $T \to \infty$, in the general situation where we do not make structural assumptions on the data distribution p_t.

Proposition 2 (Consistency and asymptotics of train-val method)**.** *Suppose $\mathbb{E}_{\mathbf{x}\sim p_t}[\mathbf{x}\mathbf{x}^{\top}] \succ \mathbf{0}$, $\mathbb{E}_{\mathbf{x}\sim p_t}[\|\mathbf{x}\|^4] < \infty$ and $\mathbb{E}_{(\mathbf{x},y)\sim p_t}[\|\mathbf{x}y\|] < \infty$ for almost surely all $p_t \sim \Pi$. Then for any $\lambda > 0$*

and any (n_1, n_2) *such that* $n_1 + n_2 = n$*, the sample splitting algorithm* $\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}val}}$ *converges to the best test-time centroid:* $\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}val}} \to \mathbf{w}_{0,\star}(\lambda, n_1)$ *almost surely as* $T \to \infty$*. Further, we have*

$$\mathrm{AsymMSE}(\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}val}}) = \mathrm{tr}\big(\nabla^{-2} L_{\lambda,n_1}^{\mathsf{test}}(\mathbf{w}_{0,\star}(\lambda,n_1)) \cdot \mathrm{Cov}\big(\nabla \ell_t^{\mathsf{tr\text{-}val}}(\mathbf{w}_{0,\star}(\lambda,n_1))\big) \cdot \nabla^{-2} L_{\lambda,n_1}^{\mathsf{test}}(\mathbf{w}_{0,\star}(\lambda,n_1))\big),$$
$$\mathrm{AsymExcessRisk}_{L_{\lambda,n_1}^{\mathsf{test}}}(\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}val}}) = \mathrm{tr}\big(\nabla^{-2} L_{\lambda,n_1}^{\mathsf{test}}(\mathbf{w}_{0,\star}(\lambda,n_1)) \cdot \mathrm{Cov}\big(\nabla \ell_t^{\mathsf{tr\text{-}val}}(\mathbf{w}_{0,\star}(\lambda,n_1))\big)\big).$$

Proposition 3 (Inconsistency of train-train method)**.** *There exists a distribution of tasks* Π *on* $d = 1$ *satisfying the conditions in Proposition 2 on which the train-train method does not converge to the best test-time centroid: for any* $n \geq 1$ *and any* $\lambda > 0$*, the estimation error* $\|\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}tr}} - \mathbf{w}_{0,\star}(\lambda, n)\|$ *and the excess risk* $L_{\lambda,n}^{\mathsf{test}}(\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}tr}}) - L_{\lambda,n}^{\mathsf{test}}(\mathbf{w}_{0,\star}(\lambda, n))$ *are both bounded away from 0 almost surely as* $T \to \infty$*.*

Propositions 2 and 3 justify the importance of sample splitting: the train-val method converges the best test-time centroid, whereas the train-train method does not converge to the best centroid in general. Appendix B gives the proofs of Proposition 2 and 3.

4 Is sample splitting always optimal?

Proposition 3 states a negative result for the train-train method, showing that it does not converge to the best test-time centroid without further assumptions on the data distribution. However, such a negative result is inherently *worst-case*, and does not preclude the possibility that there exists a data distribution on which the train-train method can also work well. In this section, we construct a simple data distribution in which we can analyze the performance of both the train-val and the train-train methods more explicitly, showing that sample splitting is indeed not optimal, and the train-train method can work better.

Realizable linear model We consider the following instantiation of the (generic) data distribution assumption in (1): We assume that each task p_t is specified by a $\mathbf{w}_t \in \mathbb{R}^d$ sampled from some distribution Π (overloading notation), and the observed data follows the noiseless linear model with ground truth parameter $\mathbf{w}_t$:

$$\mathbf{y}_t = \mathbf{X}_t \mathbf{w}_t, \tag{6}$$

where the inputs $\mathbf{x}_{t,i} \overset{\text{iid}}{\sim} \mathsf{N}(0, \mathbf{I}_d)$ and are independent of $\mathbf{w}_t$. We assume that Π has a finite second moment (i.e. $\mathbb{E}_{\mathbf{w}_t \sim \Pi}[\|\mathbf{w}_t\|^2] < \infty$).

4.1 Consistency and asymptotic rates

We begin by showing that on this task and data distribution, the population best centroids $\mathbf{w}_{0,\star}(\lambda, n) = \arg\min_{\mathbf{w}_0} L_{\lambda,n}^{\mathsf{test}}(\mathbf{w}_0)$ is the same for any (λ, n), and both the train-val and train-train methods are asymptotically consistent and converge to same best centroid.

Theorem 4 (Consistency of both train-val and train-train methods)**.** *On the realizable linear model (6), the test-time meta loss for all* $\lambda > 0$ *and all* n *is minimized at the same point, that is, the mean of the ground truth parameters:*

$$\mathbf{w}_{0,\star}(\lambda, n) = \arg\min_{\mathbf{w}_0} L_{\lambda,n}^{\mathsf{test}}(\mathbf{w}_0) = \mathbf{w}_{0,\star} := \mathbb{E}_{\mathbf{w}_t \sim \Pi}[\mathbf{w}_t], \quad \textit{for all } \lambda > 0,\ n.$$

Further, both the train-val method and the train-train method is asymptotically consistent: for any $\lambda > 0$*,* n*, and* (n_1, n_2)*, we have*

$$\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}val}}(n_1, n_2; \lambda) \to \mathbf{w}_{0,\star} \quad \text{and} \quad \widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}tr}}(n; \lambda) \to \mathbf{w}_{0,\star} \quad \textit{almost surely as} \quad T \to \infty.$$

Theoerem 4 suggests that we are now able to compare performance of the two methods based on their asymptotic parameter estimation error (for estimating $\mathbf{w}_{0,\star}$), which we state in the following result. Throughout the rest of this section, let

$$R^2 := \mathbb{E}\Big[\|\mathbf{w}_t - \mathbf{w}_{0,\star}\|^2\Big]$$

denote the variance of $\mathbf{w}_t$.

Theorem 5 (Exact asymptotic rates of the train-val and train-train methods)**.** *Let* $\rho_{\mathsf{tr\text{-}tr}} = \frac{\mathbb{E}\left[\sum_{i=1}^{d}(\sigma_i^{(n)})^2/(\sigma_i^{(n)}+\lambda)^4\right]}{\left(\mathbb{E}\left[\sum_{i=1}^{d}\sigma_i^{(n)}/(\sigma_i^{(n)}+\lambda)^2\right]\right)^2}$ *and* $\rho_{\mathsf{tr\text{-}val}} = \frac{\mathbb{E}\left[\left(\sum_{i=1}^{d}\lambda^2/(\sigma_i^{(n_1)}+\lambda)^2\right)^2+(n_2+1)\sum_{i=1}^{d}\lambda^4/(\sigma_i^{(n_1)}+\lambda)^4\right]}{\left(\mathbb{E}\left[\sum_{i=1}^{d}\lambda^2/(\sigma_i^{(n_1)}+\lambda)^2\right]\right)^2}$, *where for any* n, $\sigma_1^{(n)} \geq \cdots \geq \sigma_d^{(n)}$ *denotes the eigenvalues of the matrix* $\frac{1}{n}\mathbf{X}_t^\top\mathbf{X}_t \in \mathbb{R}^{d\times d}$, *where we recall* $\mathbf{X}_t \in \mathbb{R}^{n\times d}$ *is a random matrix with i.i.d. standard Gaussian entries. For any* (n, d), *we have on the realizable linear model* (6) *that*

$$\mathrm{AsymMSE}\big(\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}tr}}(n;\lambda)\big) = dR^2\rho_{\mathsf{tr\text{-}tr}}, \quad \mathrm{AsymMSE}\big(\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}val}}(n_1,n_2;\lambda)\big) = \frac{dR^2\rho_{\mathsf{tr\text{-}val}}}{n_2}.$$

See its proof in Appendix C.2. Theorem 5 follows straightforwardly from the classical asymptotic result for empirical risk minimization (Van der Vaart, 2000) and simplifications of certain matrix traces in terms of the spectrum of the empirical covariance matrix $\frac{1}{n}\sum_{t=1}^{n}\mathbf{X}_t\mathbf{X}_t^\top$.

4.2 Comparison of train-val vs. train-train

We now present our main theoretical result, which shows that the train-train method achieves a strictly better asymptotic MSE than the train-val method, in the proportional limit of $d, n \to \infty$ and $d/n \to \gamma \in (0, 1)$.

Theorem 6 (Optimal rates of the train-val and train-train method in the proportional limit)**.** *In the high-dimensional limiting regime* $d, n \to \infty$, $d/n \to \gamma \in (0,\infty)$, *the optimal rate of the train-train method obtained by tuning the regularization* $\lambda \in (0,\infty)$ *satisfies*

$$\inf_{\lambda>0}\lim_{d,n\to\infty,d/n=\gamma}\mathrm{AsymMSE}\big(\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}tr}}(n;\lambda)\big) = \inf_{\lambda>0}\rho_{\lambda,\gamma}R^2 \overset{(\star)}{\leq} \max\left\{1+\frac{5}{27}\gamma, \frac{5}{27}+\gamma\right\}\cdot R^2,$$

and the inequality becomes equality at $\gamma = 1$. *In contrast, the optimal rate of the train-val method by tuning the regularization* $\lambda \in (0,\infty)$ *and split ratio* $s \in (0, 1)$ *is*

$$\inf_{\lambda>0, s\in(0,1)}\lim_{d,n\to\infty,d/n=\gamma}\mathrm{AsymMSE}\big(\widehat{\mathbf{w}}_{0,T}^{\mathsf{tr\text{-}val}}(ns, n(1-s);\lambda)\big) = (1+\gamma)R^2.$$

As $\max\{1+5\gamma/27, 5/27+\gamma\} < 1+\gamma$ $(\forall\gamma > 0)$, *the train-train method achieves a strictly better asymptotic rate than the train-val method when* λ *and* s *are optimally tuned in both methods.*

Implications Comparison between the analytical upper bound $\max\{1+5\gamma/27, 5/27+\gamma\}R^2$ for train-train $(1+\gamma)R^2$ for train-val in Theorem 6 shows that the train-train method achieves a strictly better asymptotic MSE (and thus also asymptotic excess risk) than the train-val method, for any $\gamma > 0$. (See Figure 1(a) for a visualization of the optimal rates.) Perhaps surprisingly, this suggests that the train-train method is not only "correct" (converging to the best centroid), but can be even better than the train-val method, when the data is structured. While we reached such a conclusion on this particular problem of linear centroid meta-learning, we suspect that this phenomenon to be not restricted to this problem, and may hold in more generality when data is structured or when the signal-to-noise ratio is high. The proof of Theorem 6 is deferred to Appendix C.7.

5 Experiments

Simulation. We experiment on the realizable linear model studied in Section 4 and show that the train-train method indeed achieves better performance than the train-val method. Both performances agree well with the asymptotic theoretical prediction, even at a fairly small $(n, d) = (20, 60)$ (Figure 1, more details in Appendix A.1).

Real data. We additionally find that the train-train method consistently outperforms the train-val method on few-shot image classification benchmarks (Table 1, more details in Appendix A.2).

6 Conclusion

We study the importance of train-validation split on the linear-centroid meta-learning problem, and show that the necessity and optimality of train-validation split depends greatly on whether the tasks are structured: the sample splitting is necessary in general situations, and not necessary and non-optimal when the tasks are nicely structured. It would be of interest to study whether a similar conclusion holds on other meta-learning problems such as learning a representation, or whether our insights can be used towards designing meta-learning algorithms with better empirical performance.

References

Pierre Alquier, The Tien Mai, and Massimiliano Pontil. Regret bounds for lifelong learning. *arXiv preprint arXiv:1610.08628*, 2016.

Greg W Anderson, Alice Guionnet, and Ofer Zeitouni. *An introduction to random matrices*, volume 118. Cambridge university press, 2010.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pp. 41–48, 2007.

Zhidong Bai and Jack W Silverstein. *Spectral analysis of large dimensional random matrices*, volume 20. Springer, 2010.

Jonathan Baxter. A model of inductive bias learning. *J. Artif. Int. Res.*, 2000.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997. ISSN 1573-0565. doi: 10.1023/A:1007379606734. URL https://doi.org/10.1023/A:1007379606734.

Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Incremental learning-to-learn with statistical guarantees. *arXiv preprint arXiv:1803.08089*, 2018a.

Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Learning to learn around a common mean. In *Advances in Neural Information Processing Systems*, pp. 10169–10179, 2018b.

Edgar Dobriban, Stefan Wager, et al. High-dimensional asymptotics of prediction: Ridge regression and classification. *The Annals of Statistics*, 46(1):247–279, 2018.

Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of machine learning research*, 6(Apr):615–637, 2005.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1082–1092, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 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, 2017.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. *arXiv preprint arXiv:1806.04910*, 2018.

Tomer Galanti, Lior Wolf, and Tamir Hazan. A theoretical framework for deep transfer learning. *Information and Inference: A Journal of the IMA*, 5(2):159–209, 2016.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*, 2018.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. *arXiv preprint arXiv:1906.02717*, 2019.

A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. pp. 1097–1105, 2012.

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

Percy Liang. Cs229t/stat231: Statistical learning theory (winter 2016), 2016.

Han Liu, Mark Palatucci, and Jian Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 649–656, 2009.

Han Liu, Lie Wang, and Tuo Zhao. Calibrated multivariate regression with application to neural semantic basis discovery. *Journal of machine learning research: JMLR*, 16:1579, 2015.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884, 2016.

Daniel McNamara and Maria-Florina Balcan. Risk bounds for transferring representations with and without fine-tuning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2373–2381. JMLR. org, 2017.

A. Nichol and J. Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2, 2018.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pp. 113–124, 2019.

S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. 2017.

M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. Tenenbaum, H. Larochelle, and R. Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. Imagenet large scale visual recognition challenge. 115(3):211–252, 2015.

Nikunj Saunshi, Yi Zhang, Mikhail Khodak, and Sanjeev Arora. A sample complexity separation between non-convex and convex meta-learning. *arXiv preprint arXiv:2002.11172*, 2020.

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

J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. pp. 4077–4087, 2017.

Sebastian Thrun and Lorien Pratt. *Learning to Learn: Introduction and Overview*, pp. 3–17. Springer US, Boston, MA, 1998. ISBN 978-1-4615-5529-2. doi: 10.1007/978-1-4615-5529-2_1. URL https://doi.org/10.1007/978-1-4615-5529-2_1.

Nilesh Tripuraneni, Chi Jin, and Michael I Jordan. Provable meta-learning of linear representations. *arXiv preprint arXiv:2002.11684*, 2020.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. On the global optimality of model-agnostic meta-learning. *arXiv preprint arXiv:2006.13182*, 2020a.

Xiang Wang, Shuai Yuan, Chenwei Wu, and Rong Ge. Guarantees for tuning the step size using a learning-to-learn approach. *arXiv preprint arXiv:2006.16495*, 2020b.

Yujia Xie, Haoming Jiang, Feng Liu, Tuo Zhao, and Hongyuan Zha. Meta learning with relational information for short sequences. In *Advances in Neural Information Processing Systems*, pp. 9904–9915, 2019.

Pan Zhou, Xiaotong Yuan, Huan Xu, Shuicheng Yan, and Jiashi Feng. Efficient meta learning via minibatch proximal update. In *Advances in Neural Information Processing Systems*, pp. 1534–1544, 2019.