How to Distribute Data across Tasks for Meta-Learning?

Alexandru Cioba, [1] Michael Bromberg, [1] Qian Wang, [1] Ritwik Niyogi, [1]
Georgios Batzolis, [2] Jezabel Garcia, [1] Da-shan Shiu, [1] Alberto Bernacchia [1]

[1] MediaTek Research, [2]University of Cambridge

Abstract

Meta-learning models transfer the knowledge acquired from previous tasks to quickly learn new ones. They are trained on benchmarks with a fixed number of data points per task. This number is usually arbitrary and it is unknown how it affects performance at testing. Since labelling of data is expensive, finding the optimal allocation of labels across training tasks may reduce costs. Given a fixed budget of labels, should we use a small number of highly labelled tasks, or many tasks with few labels each? Should we allocate more labels to some tasks and less to others? We show that: 1) If tasks are homogeneous, there is a uniform optimal allocation, whereby all tasks get the same amount of data; 2) At fixed budget, there is a trade-off between number of tasks and number of data points per task, with a unique solution for the optimum; 3) When trained separately, harder task should get more data, at the cost of a smaller number of tasks; 4) When training on a mixture of easy and hard tasks, more data should be allocated to easy tasks. Interestingly, Neuroscience experiments have shown that human visual skills also transfer better from easy tasks. We prove these results mathematically on mixed linear regression, and we show empirically that the same results hold for few-shot image classification on CIFAR-FS and mini-ImageNet. Our results provide guidance for allocating labels across tasks when collecting data for meta-learning.

Introduction

Deep learning (DL) models require a large amount of data in order to perform well, when trained from scratch, but labeling data is expensive and time consuming. An effective approach to avoid the costs of collecting and labeling a large amount of data is transfer learning: train a model on one big dataset, or a few related datasets that are already available, and then fine-tune the model on the target dataset, which can be of much smaller size (Donahue et al. 2014). In this context, there has been a recent surge of interest in the field of *meta-learning*, which is inspired by the ability of humans to *learn how to learn* (Hospedales et al. 2020). A model is *meta-trained* on a large number of tasks, each characterized by a small dataset, and *meta-tested* on the target dataset.

The number of data points per task is usually set to an arbitrary number in standard meta-learning benchmarks. For example, in few-shot image classification benchmarks, such as *mini*-ImageNet (Vinyals et al. 2017), (Ravi and Larochelle 2017) and CIFAR-FS (Bertinetto et al. 2019), each task has five classes (5-way) and either one or five images per class is used during testing (1-shot or 5-shots). During training, the number of data points per class is usually set to an arbitrary value, and it remains unclear how this number should be set to achieve the best testing performance. We focus on training, rather than testing data, because the former can be optimized by following specific procedures for data partitioning and collection.

Intuitively, one would think that the performance always improves with the number of training data points. However, if the total number of labels is limited, is it better to have a large number of tasks with little data in each task, or a smaller number of highly labelled tasks? Should some tasks be given more labels than other tasks? The answers to these questions remain unknown, although they are important to inform the design of new meta-learning benchmarks and the application of meta-learning algorithms to real problems, especially given that data labelling is costly. Hence, we address these questions for the first time, for a specific meta-learning algorithm: MAML (Finn, Abbeel, and Levine 2017). Our contributions are:

- We introduce the problem of optimizing data allocation in meta-learning, with a fixed budget of total data points to distribute across training tasks. We show that, when tasks are homogeneous, the optimal solution is distributing data uniformly across tasks: all tasks get the same amount of data. This setting is considered in most meta-learning problems (See *'The data allocation problem'* section , Theorem 1).
- When data is distributed uniformly across tasks, we show that the trade-off between number of tasks and number of data points per task, at fixed budget, has a unique solution for the optimum for large budgets (section *'Solution of the uniform allocation'*, Theorems 2, 3, Figures 1, 2).
- Next, we consider the problem of two sets of tasks, easy and hard. When trained separately, we show that hard tasks need more data (per task) than easy tasks. While it is intuitive that hard tasks require more data for training, we emphasize that the total number of data points is fixed by the given budget, therefore the number of tasks is smaller (section *'Separate training'*, Figure 3).

- Finally, we study the problem of training a non-homogeneous mixture of easy and hard tasks. In contrast to when they are trained separately, we show that better performance is obtained by allocating more data to easy tasks. Our interpretation is that, as long as learning transfers from easy to hard tasks, it is better to train more on the former since they are easier to learn. Interestingly, human visual skills also transfer better from easy tasks (Ahissar and Hochstein 1997) (section *'Joint training'* , Figure 4).

We prove results mathematically on mixed linear regression, and confirm those results empirically on few-shot image classification on CIFAR-FS and *mini*-ImageNet (code in the supplementary material).

Related Work

In the context of meta-learning and mixed linear regression, the work of (Kong et al. 2020) asks whether more tasks with a small amount of data can compensate for a lack of tasks with big data. However, they do not address the problem of finding the optimal allocation of data for a fixed budget, which is the main scope of our work. The work of (Shekhar, Javidi, and Ghavamzadeh 2020) studies the problem of allocating a fixed budget of data points to a finite set of discrete distributions. In contrast to our work, they do not study the meta-learning problem and their data has no labels. Similar to us, a few theoretical studies looked at the problem of mixed linear regression in the context of meta-learning ((Bernacchia 2021), (Denevi et al. 2018), (Bai et al. 2021), (Tripuraneni, Jin, and Jordan 2020), (Du et al. 2020), (Collins, Mokhtari, and Shakkottai 2020), (Gao and Sener 2020)). However, none of these studies look into the problem of data allocation, which is our main focus.

An alternative approach to avoid labelling a large amount of data is *active learning*, where a model learns with fewer labels by accurately selecting which data to learn from (Settles 2010). In the context of meta-learning, the option of implementing active learning has been considered in a few recent studies (Bachman, Sordoni, and Trischler 2017), (Garcia and Bruna 2018), (Kim et al. 2018), (Finn, Xu, and Levine 2019), (Requeima et al. 2020). However, they considered the active labeling of data within a given task, for the purpose of improving performance in that task only. Instead, we ask how data should be distributed across tasks.

In the context of recommender systems and text classification, a few studies considered whether labeling a data point, within a given task, may increase performance not only in that task but also in all other tasks. This problem has been referred to as *multi-task active learning* (Reichart et al. 2008), (Zhang 2010), (Saha et al. 2011), (Harpale 2012), (Fang et al. 2017), or *multi-domain active learning* (Li et al. 2012), (Zhang et al. 2016). However, none of these studies consider the problem of meta-learning with a fixed budget. A few studies have looked into actively choosing the next task in a sequence of tasks (Ruvolo and Eaton 2013), (Pentina, Sharmanska, and Lampert 2015), (Pentina and Lampert 2017), (Sun, Cong, and Xu 2018), but they do not look at how to distribute data across tasks.

Meta-Learning

The reader may refer to (Hospedales et al. 2020) for a general introduction to meta-learning with neural networks. In this work, we consider the cross-task setting, where we have a distribution of tasks $\tau \sim p(\tau)$ and a distribution of data points for a given task $\mathcal{D}^\tau \sim p(\mathcal{D}|\tau)$. Each task has a loss function $\mathcal{L}(\theta; \mathcal{D})$ that depends on a set of parameters θ and data $\mathcal{D}$. Here we assume that the loss has the same functional form across tasks (e.g. square loss if they are all regression tasks, cross-entropy if they are all classification tasks). The goal of meta-learning is minimizing the mean of the loss across tasks and data.

In the *meta-training* phase, m tasks $(\tau_i)_{i=1}^m$ are sampled from $p(\tau)$ and, for each task, n_i^t training data points $\mathcal{D}_i^t = (\mathbf{x}_{ij}^t, y_{ij}^t)_{j=1}^{n_i^t}$ and n_i^v validation data points $\mathcal{D}_i^v = (\mathbf{x}_{ij}^v, y_{ij}^v)_{j=1}^{n_i^v}$, are sampled independently from the same distribution $p(\mathcal{D}|\tau_i)$. We assume that the data is given by input $\mathbf{x}$ - label y pairs. The meta-training loss is a function of the data and the meta-parameters $\boldsymbol{\omega}$, is equal to

$$\mathcal{L}^{meta}\left(\boldsymbol{\omega}; \mathcal{D}^t, \mathcal{D}^v\right) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i^v}\sum_{j=1}^{n_i^v}\mathcal{L}\Big(\boldsymbol{\theta}(\boldsymbol{\omega}, \mathcal{D}_i^t); \mathbf{x}_{ij}^v, y_{ij}^v\Big) \tag{1}$$

The parameters are adapted to each task i by using the transformation $\theta(\boldsymbol{\omega}, \mathcal{D}_i^t)$. Different meta-learning algorithms correspond to a different choice of this transformation. Here we use MAML (Finn, Abbeel, and Levine 2017), which performs a fixed number of stochastic gradient descent steps with respect to the data for each task. With a single gradient step, that is equal to

$$\theta(\boldsymbol{\omega}, \mathcal{D}_i^t) = \boldsymbol{\omega} - \frac{\alpha_i}{n_i^t}\sum_{j=1}^{n_i^t}\nabla_{\boldsymbol{\omega}}\mathcal{L}\left(\boldsymbol{\omega}; \mathbf{x}_{ij}^t, y_{ij}^t\right) \tag{2}$$

where α_i is the learning rate for task i. This equation corresponds to a full-batch update, employing all the data for a given task, but mini-batch gradient updates can be performed as well. A number k of gradient steps may be used instead of one. This step is referred to as *inner loop* of meta-learning.

The loss in Eq.(1) is minimized with respect to the meta-parameters $\boldsymbol{\omega}$, namely

$$\boldsymbol{\omega}^\star\left(\mathcal{D}^t, \mathcal{D}^v\right) = \arg\min_{\boldsymbol{\omega}}\mathcal{L}^{meta}\left(\boldsymbol{\omega}; \mathcal{D}^t, \mathcal{D}^v\right) \tag{3}$$

This minimum is searched by stochastic gradient descent, using a distinct learning rate α_{meta}. At each gradient step, Eq.(2) is computed for each task and the gradient of Eq.(1) with respect to $\boldsymbol{\omega}$ is taken. This step is referred to as *outer loop* of meta-learning. Note that Eq.(1) includes all m tasks, which translates into full-batch training when taking the gradient. However, a mini-batch of tasks may be also drawn from the set of m tasks at each step of the optimization. Standard optimization procedures such as early stopping and scheduling of the learning rate α_{meta} can be applied. In the case of mixed linear regression (section *'Solution of the uniform allocation'*), we solve Eq.(3) exactly by linear algebra.

In the *meta-testing* phase, the test loss $\mathcal{L}^{test}$ is computed using the optimal value $\boldsymbol{\omega}^\star$ and test datasets $\tilde{\mathcal{D}}^t, \tilde{\mathcal{D}}^v$

$$\mathcal{L}^{test}\left(\mathcal{D}^t, \mathcal{D}^v, \tilde{\mathcal{D}}^t, \tilde{\mathcal{D}}^v\right) = \mathcal{L}^{meta}\left(\boldsymbol{\omega}^\star\left(\mathcal{D}^t, \mathcal{D}^v\right); \tilde{\mathcal{D}}^t, \tilde{\mathcal{D}}^v\right) \tag{4}$$

The test datasets correspond to a new draw of both tasks and data points. The values of hyperparameters m, n^t, n^v, α, k for meta-testing are not necessarily the same as those used during meta-training. The main focus of this work is optimizing m, n_i^t, n_i^v for meta-training, while they are fixed during meta-testing. To evaluate the performance of the model for a given choice of the hyperparameters, we compute the average test loss, defined as

$$\overline{\mathcal{L}}^{test}(n_1^t, \dots n_m^t, n_1^v, \dots n_m^v) = \mathop{\mathbb{E}}_{\mathcal{D}_t}\mathop{\mathbb{E}}_{\mathcal{D}_v}\mathop{\mathbb{E}}_{\tilde{\mathcal{D}}_t}\mathop{\mathbb{E}}_{\tilde{\mathcal{D}}_v}\mathcal{L}^{test} \tag{5}$$

The Data Allocation Problem

We denote the number of data points per task i during meta-training as $N_i = n_i^t + n_i^v$, equal to the sum of training and validation data. In all experiments we used an equal split of training and validation, $n_i^t = n_i^v = n_i$. We assume that the total number of data points for meta-training, referred to as *budget*, is constant and equal to $b = \sum_{i=1}^m N_i = 2\sum_{i=1}^m n_i$. This is equal to the total number of data points across all training tasks, and is assumed fixed, while the number of data points per task N_i are allowed to vary. We denote by $\mathbf{n}$ the vector of n_i values, $\mathbf{n} = (n_1, \dots, n_m)$, and define the *data allocation* problem of finding the value of $\mathbf{n}$ such that the average test loss is minimized

$$\mathbf{n}^\star = \mathop{\arg\min}_{\mathbf{n}\,:\,\sum_{i=1}^m n_i = b/2} \overline{\mathcal{L}}^{test}(n_1, \dots, n_m) \tag{6}$$

The optimal value $\mathbf{n}^\star$ is referred to as *optimal allocation*, it may depend on the budget and on other hyperparameters of the model. The optimal allocation determines which tasks should get more or less data, for a fixed budget b and number of tasks m. In the following theorem, we provide conditions under which the optimal data allocation is uniform.

Theorem 1. If the test loss $\overline{\mathcal{L}}^{test}$ is invariant under permutations of task allocations, i.e. permutations of its arguments $(n_1, \dots, n_m)$ then the uniform allocation $\mathbf{n} = (n, \dots, n)$ with $n = \frac{b}{2m}$ is a local extremum of the constrained optimization problem, provided that it is non-degenerate.

Furthermore, if

$$\overline{\mathcal{L}}^{test}\left(\frac{n_1+n_2}{2}, \frac{n_1+n_2}{2}, n_3, ..., n_m\right) \leq \mathcal{L}^{test}(n_1, ..., n_m), \tag{7}$$

for all $n_1, ..., n_m$, subject to $\sum_{i=1}^k n_i = \frac{b}{2}$, then the uniform allocation is the global minimum of the data allocation problem.

Proof. The proof of the first part (see Modern Purkiss principle) is given by (Waterhouse 1983), noting that the action of the symmetric group preserves the constraint and is irreducible, while the proof of the second part (global minimum) is given by (Keilson 1967). □

Note that convexity of the test loss is a sufficient condition for the global minimum. We show in the *'Mixed linear regression'* section that the Purkiss principle applies to the case of mixed linear regression with homogeneous tasks. This result motivates, in addition to the data allocation problem (6), the study of the *uniform allocation* problem, in which the number of data points is assumed to be equal across tasks, but now the number of tasks m is allowed to vary. The solution of this problem is defined by

$$n^\star = \mathop{\arg\min}_{n\,:\,nm=b/2} \overline{\mathcal{L}}^{test}(n) \tag{8}$$

In this case, the question is whether to have more data and less tasks, or less data and more tasks, for the fixed budget b. In the next sections, we study both problems of data allocation and uniform allocation on mixed linear regression and few-shot image classification on CIFAR-FS and *mini*-ImageNet.

Computation of the Optimum

In the case of linear regression, we derive exact expressions for $\overline{\mathcal{L}}^{test}$ and $\mathbf{n}^\star$ in some limiting cases. In few-shot image classification, and in further linear regression experiments, we estimate $\overline{\mathcal{L}}^{test}$ empirically by searching a grid of values of $\mathbf{n}$. We average the test loss over multiple repetitions with different data samples and different initial conditions for $\boldsymbol{\omega}$. Then, we determine the mean and standard deviation for the optimum $\mathbf{n}^\star$ by the following procedure: we generate multiple instances of test loss/accuracy vs $\mathbf{n}$ by sampling uniformly from the repetitions at each value of $\mathbf{n}$, we record the optimal $\mathbf{n}^\star$ of each instance and construct a distribution of $\mathbf{n}^\star$ across all instances. We also provide nonlinear (sinusoid) regression experiments in the supplementary material of (Cioba et al. 2021).

Solution of the Uniform Allocation

In this section we consider the problem of uniform allocation, while the non-uniform case is studied in the section *'Easy vs hard tasks'* . We look at the trade-off between having either more tasks or more data per task, for a fixed budget, and we show that this problem has a unique optimum. We study this trade-off on two problems: mixed linear regression, where we compute a closed form expression for the optimum, and few-shot image classification, where we show empirical results.

Mixed Linear Regression

In mixed linear regression, each task is characterized by a different linear function, and the loss is the mean squared error:

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}, y) = \frac{1}{2}\left(y - \boldsymbol{\theta}^T\mathbf{x}\right)^2 \tag{9}$$

where the label y is a scalar, while the input $\mathbf{x}$ and the parameter $\boldsymbol{\theta}$ are vectors of p components. Each task corresponds to a different value of the generating parameter

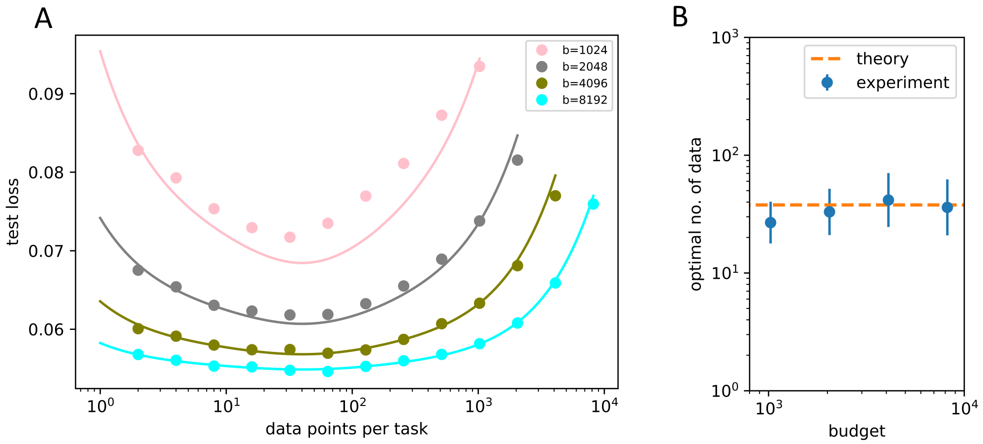

Figure 1: The optimal number of data points per task is constant for large budgets: linear regression. A: Test loss vs. number of data points per task at fixed budget (more data points imply less tasks). Dots: experimental values; Lines: theoretical prediction Eq.(10), different lines correspond to different budgets (legend). As predicted by Theorem 2, theoretical prediction is more accurate for larger budgets. Each curve has a unique optimum. B: Optimal number of data points per task vs. budget, the four points correspond to the four curves in panel A. The theoretical prediction of Eq.(11) (orange line) is close to the estimated experimental optimum (see section *'Computation of the optimum'* for its computation).

$\boldsymbol{\theta}$. Across tasks, that is distributed according to a Gaussian $\boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\theta}_0, \frac{\nu^2}{p} I_p\right)$

where $\boldsymbol{\theta}_0, \nu$ are hyperparameters, and I_p is the $p \times p$ identity matrix. The distribution of data for a given task is given by $y \mid \mathbf{x}, \boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\theta}^T \mathbf{x}, \sigma^2\right)$ and $\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \lambda^2 I_p\right)$

where σ is the label noise and λ is the input variability. Each data point is independently drawn from this distribution, for either training or validation set. We distinguish between the case of *homogeneous* tasks, where all tasks have the same values of (σ, λ), and *non-homogeneous* tasks, where we allow those values to vary across tasks. In the following theorem, we compute an approximate expression for the average test loss for mixed linear regression.

Theorem 2. Consider the algorithm of section (MAML one-step) and data generated according to the mixed linear regression model. Let $\sum_{i=1}^m n_i > p$ (underparameterized model), and let $n_i = n_i(\xi)$, $m = m(\xi)$ be any functions of order $\Theta(\xi)$ as $\xi \to \infty$. Then, the average test loss is equal to

$$\begin{aligned}\overline{\mathcal{L}}^{test} &= \frac{\sigma_r^2}{2}\left(1+\frac{\lambda_r^4\alpha_r^2 p}{n_r}\right)+\frac{\lambda_r^2 h_r \nu^2}{2}+\\ &+\frac{\lambda_r^2 h_r p}{2}\left[\sum_{i=1}^m \lambda_i^2 h_i\right]^{-2}\sum_{i=1}^m \frac{\lambda_i^2}{n_i}\Bigg\{\\ &\sigma_i^2\left[h_i+\frac{\lambda_i^4\alpha_i^2}{n_i}\left[(n_i+1)\,g_{1i}+p\;g_{2i}\right]\right]+\\ &+\frac{\nu^2}{p}\lambda_i^2\left[(n_i+1)\,g_{3i}+p\;g_{4i}\right]\Bigg\}+O\left(\xi^{-3}\right) \qquad (10)\end{aligned}$$

where the subscript i denotes meta-training hyperparameters for task i, while the subscript r denotes meta-testing hyperparameters. We have defined the function $h_i = \left(1-\lambda_i^2\alpha_i\right)^2 + \lambda_i^4\alpha_i^2\frac{p+1}{n_i}$, and the functions g are polynomials in $\lambda_i^2\alpha_i$ with coefficients of order $O(1)$, defined in the appendix of (Cioba et al. 2021).

Proof. The proof is given in (Cioba et al. 2021). It provides a generalization of the results of (Bernacchia 2021) in the case of non-homogeneous tasks and parametric input variability. □

When tasks are homogeneous ($\sigma_i = \sigma$, $\lambda_i = \lambda$) and a fixed learning rate is used for all meta-training tasks ($\alpha_i = \alpha$), we note that the test loss (10) is permutation invariant, thus the Purkiss principle of Theorem 1 applies. Therefore, in the remainder of this section we consider only the case of uniform allocation ($n_i = n$). Non-homogeneous tasks and non-uniform allocation are studied in the *'Easy vs hard tasks'* section . Note also that Theorem 2 assumes an underparameterized model ($p < \sum_{i=1}^m n_i$). For completeness, we also study the overparameterized case in the supplement to (Cioba et al. 2021).

Figure 1A plots the meta-test loss of mixed linear regression as a function of n for different budgets. It shows a good agreement between the experiments and the theoretical prediction of equation (10) (see (Cioba et al. 2021) for details). According to equation (10), the error between theory and experiment is expected to be of order $O\left(b^{-3/2}\right)$, since $b \sim O\left(\xi^2\right)$, indeed theoretical prediction is more accurate for larger budgets. As expected, test loss decreases with budget, since more data implies better performance. We emphasize that curves have a convex shape, implying that there is a unique optimal value of n for each budget. While the curves tend to flatten at large budgets, the optimum remains approximately constant, as shown in Figure 1B. In

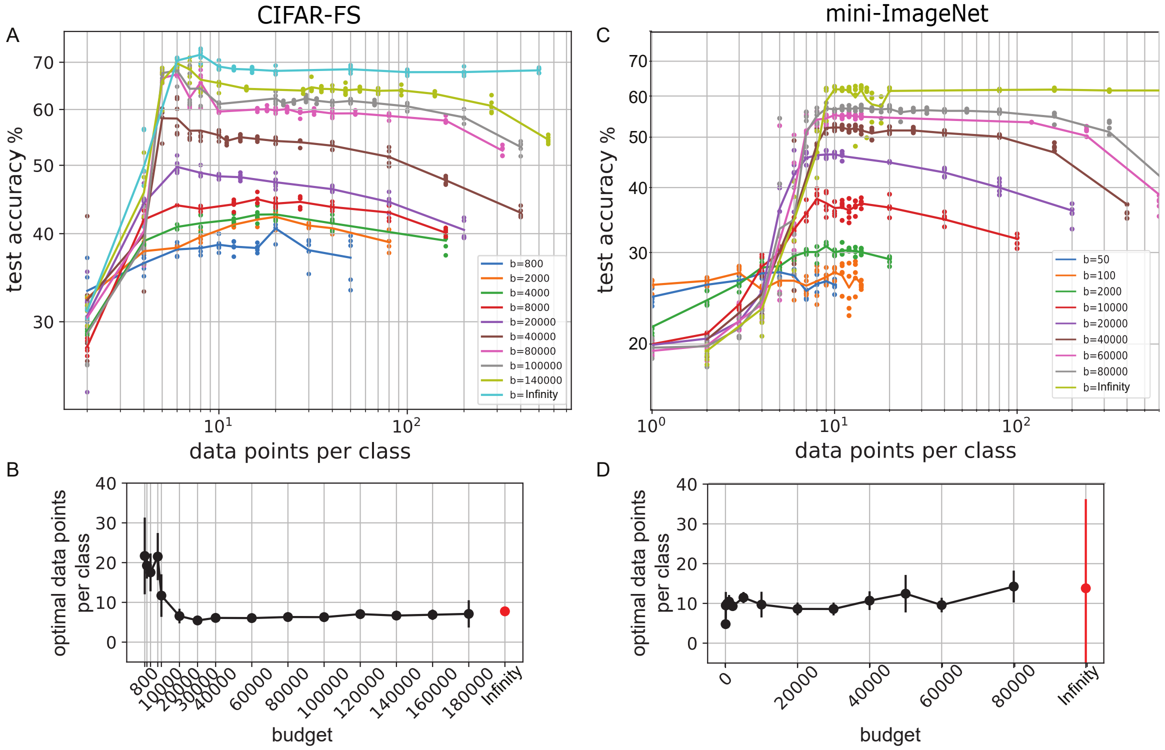

Figure 2: The optimal number of data points per task is constant for large budgets: Few-shot image classification: A,B: CIFAR-FS, D,E: *mini*-ImageNet dataset. Format is the same as of Figure 1. Curves are noisy and tend to flatten at large budgets, but there seems to be a unique optimum for each budget value. The optimum is computed empirically as explained in the *'Computation of the optimum'* section . The optimal number of data points converges to $\sim$ 7 for CIFAR-FS and to $\sim$ 10 for *mini*-ImageNet. Error bars show standard deviation.

the following theorem, we compute the unique solution of the uniform allocation problem for mixed linear regression.

Theorem 3. Under the assumptions of Theorem 2, consider the test loss of Equation (10) and the uniform allocation problem in Equation (8) . Furthermore, let $p = p(\xi)$ be a function of order $\Theta(\xi)$ as $\xi \to \infty$, neglect orders $O\left(\xi^{-2}\right)$ in Equation (10). Then, for all sufficiently small values of the learning rate α, the uniform allocation problem has a unique minimum, which does not depend on the budget and is given by

$$n^{\star} = Cp$$

where the constant C is defined in the appendix of (Cioba et al. 2021).

Proof. The proof is provided in the appendix of (Cioba et al. 2021). □

This theorem implies that once the suitable error terms in the approximation of $\mathcal{L}^{test}$ are ignored, there is a unique and constant optimum for the number of data points per task at large budgets. Note that the magnitude of the error terms does depend on the budget and the relation between n, p and m. While the theoretical optimum does not depend on the budget, it may depend on whether tasks are hard or easy (see section *'Easy vs hard tasks'*). Figure 1B shows the optimal $n^{\star}$ as a function of budget, it shows that the theoretical value of the optimum (orange line) agrees with the experiments.

Few-Shot Image Classification

We next tested whether the results of mixed linear regression generalize to the more interesting problem of few-show image classification. In this case, the loss function is the cross-entropy, $\mathcal{L}\left(\theta; x, y\right) = -y^{T} \log\left(f_{\theta}(x)\right)$, where y is a one-hot encoding of the class label, and $f_{\theta}(x)$ is the output vector of a neural network with parameters θ and input x. We use a convolutional neural network commonly used with MAML on image classification (Finn, Abbeel, and Levine 2017) (see appendix of (Cioba et al. 2021)).

We investigate the CIFAR-FS (Bertinetto et al. 2019) and *mini*-ImageNet (Vinyals et al. 2017) datasets, which are few-shot versions of CIFAR-100 and ImageNet, respectively. Both classification problems are 5-way: each task contains 5 classes. We refer to the number of data points *per class*, which has to be multiplied by 5 to find the number of data points *per task*. As in previous studies, we used 5 *shots* during testing (5 data points per class), while the number of shots during training depends on the data allocation.

In previous work (Vinyals et al. 2017), (Bertinetto et al. 2019), we note that tasks are usually re-sampled indefinitely

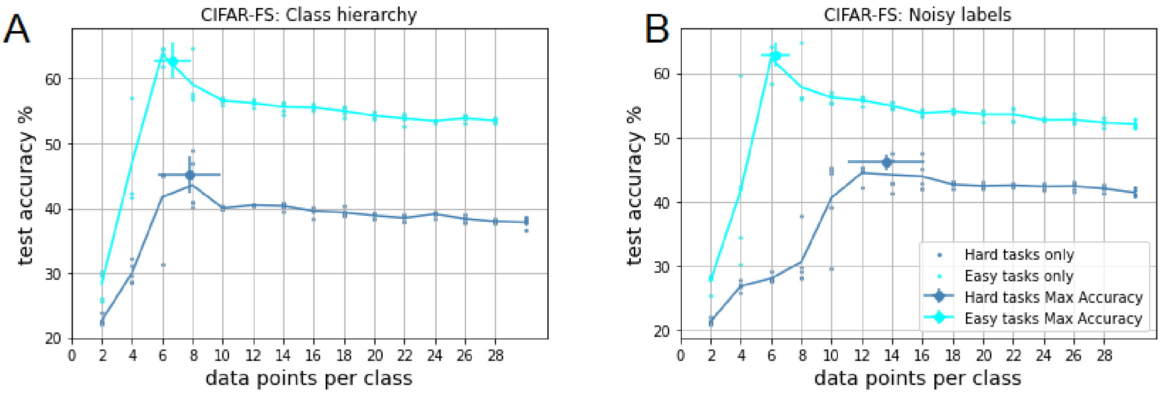

Figure 3: Hard tasks prefer more data (and less tasks) when trained separately. Few shot image classification on CIFAR-FS. A: Tasks are made harder by drawing classes within a hierarchy; B: Tasks are made harder by adding label noise. Both plots show test accuracy versus the number of data points per class, as in Figure 2A. Each plot shows an estimate of the point of maximum accuracy, with error bars showing standard deviation (see the *'Computation of the optimum'* section for its computation). In both cases, performance is lower and the optimal number of data points per class is larger for hard tasks.

until convergence of the model, thus there is no limit on the number of tasks that can be generated. We instead pre-sample a set of tasks in order to fix the budget constraint. For comparison, we also run experiments in the usual way, and we call this the *infinite budget* case. However, the total number of labels is fixed and, even if tasks are re-sampled indefinitely, it does not imply that the amount of data is infinite, rather the same image may appear in multiple tasks.

As expected, Figure 2 shows that test performance improves with the budget, for both CIFAR-FS and *mini*-ImageNet (Figure 2A,C). For infinite budget, the accuracy is similar to previously reported values ($\sim$ 63% for *mini*-ImageNet (Finn, Abbeel, and Levine 2017), $\sim$ 71% for CIFAR-FS (Bertinetto et al. 2019)). For CIFAR-FS, the optimal number of data points per class was $\sim$ 20 at small budgets, but decreased and remained approximately constant at $\sim$ 7 for large budgets (Figure 2B). For *mini*-ImageNet, the optimal number of data points per class was $\sim$ 5 at very small budget and then increased and remain approximately constant at $\sim$ 10 (Figure 2D). The performance curves in Figure 2A,C tend to flatten at higher budgets, but the optimum does not change significantly. Overall, the empirical study of both datasets confirms our prediction that the optimal number of data points per task is constant at large budgets.

Easy vs. Hard Tasks

In this section we consider the case of non-homogeneous tasks. We distinguish between two sets of tasks, easy and hard. We use two independent definitions of hard tasks, one affects the input and the other affects the output (label) of a dataset. We apply this definition in a similar way to both mixed linear regression and few-shot image classification.

For the problem of mixed linear regression, we define *task difficulty* in terms of the hyperparameters σ and λ. A task is harder if it has a larger σ (at equal λ) or smaller λ (at equal σ). The case of larger σ is intuitive, a task is harder to learn if labels are more corrupted by noise. In the case of smaller λ, the smaller input range makes it harder to solve the regression problem in presence of noise.

In few-shot image classification, the first method to make a task harder is to introduce label noise (Song et al. 2021): each input image has 20% probability of having its label swapped with another random class. The second method is similar to (Collins, Mokhtari, and Shakkottai 2020): we take advantage of the hierarchical tree of the CIFAR-100 dataset and we constraint each task to draw classes from one of three superclasses: 1) animals, 2) vegetations, 3) object and scenes. Therefore, we assume that each task has a smaller variability of its input, not in terms of pixels color or intensity, but in terms of semantic relations. Intuitively, it is harder to distinguish inputs when they are more similar to each other. We refer to the two different definitions as, respectively, *noisy labels* and *class hierarchy*.

Separate Training

Before studying the training of a mixture of easy and hard tasks, we ask what is the optimal uniform allocation when the two types of tasks are trained separately. In mixed linear regression, the expression for the optimum of the uniform allocation $n^\star$ is given by Eq.(11), but is hard to evaluate how it depends on σ and λ. Therefore we computed an approximation that holds for small α' (see appendix of (Cioba et al. 2021)):

$$n^\star = \left[2\left(1+\frac{\sigma'^2}{\nu^2}\right)\right]^{\frac{1}{3}} \alpha'^{\frac{4}{3}} p + O\left(\alpha'^{\frac{5}{3}}\right) \qquad (11)$$

where $\alpha' = \lambda^2\alpha$ and $\sigma' = \sigma/\lambda$. The optimum increases with σ, suggesting that harder tasks require more data (and less tasks) at fixed budget. For λ, there are two opposing forces: 1) On one hand a smaller λ is equivalent to amplifying output noise σ' and increasing the optimum $n^\star$; 2) On the other

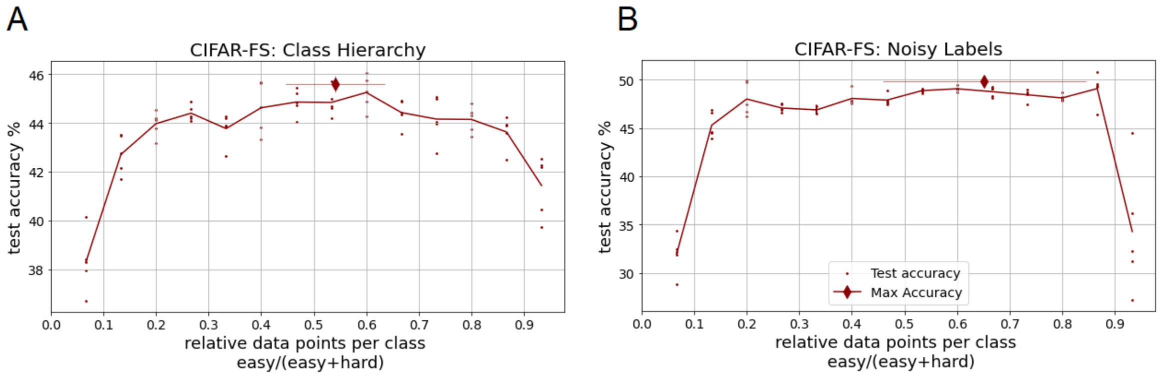

Figure 4: When training on a mixture of easy and hard tasks, it is better to allocate more data to easy tasks. Few shot image classification on CIFAR-FS. A: Tasks are made harder by drawing classes within a hierarchy; B: Tasks are made harder by adding label noise. Both plots show test accuracy versus the relative amount of easy vs hard data points per class. Each plot shows an estimate of the point of maximum accuracy, with error bars showing standard deviation. In both cases, a slightly higher performance is obtained by allocating more data to easy tasks than to hard ones.

hand, λ rescales the learning rate α' with the opposite and stronger effect that a smaller λ decreases the optimum.

Figure 3 shows that introducing task difficulty in few-shot image classification on CIFAR-FS increases the optimum for both methods (A: class hierarchy; B: noisy labels, see (Cioba et al. 2021) *'Easy and hard tasks creation'*). As expected, performance is lower for hard tasks in both cases (note that we train and test on the same set of tasks, either only easy or only hard). While it is intuitive that hard tasks require more data to learn, we emphasize that, for a fixed budget, this comes at the expense of a smaller number of tasks.

Joint Training

We now turn to the problem of training on a mixture of easy and hard tasks. In addition to a fixed budget, we further assume an equal number of easy and hard tasks, and a constant sum of easy and hard data points per task. Therefore, the only hyperparameter of interest is the relative number of data points per task for easy vs hard. Note that we use a mixture of easy and hard tasks also for testing, but we always use an equal number of easy and hard data points and tasks in that case (see (Cioba et al. 2021)).

After the results of section , we expect better results when allocating more data to hard tasks. Surprisingly we find that the opposite is true. Figure 4 shows that a slightly higher performance is obtained when allocating more data to easy tasks, in few-shot image classification on CIFAR-FS for both methods (Panel A: class hierarchy; Panel B: noisy labels). Intuitively, easy tasks are easier to learn than hard tasks. Therefore, it may be that if training on easy tasks transfers to better performance on hard tasks, then it is better to allocate more data to easy tasks.

Discussion

In this paper we analysed the problem of optimal data allocation in meta-learning when the budget of labelled examples is limited. When tasks are homogeneous, we showed that uniform data allocation across tasks is optimal (under the assumptions of Theorem 1). We further studied whether one should use less tasks with more data or more tasks and less data. For mixed linear regression, we found a unique solution for the optimum at large budgets. We confirmed this finding empirically on few-shot image classification (an example of nonlinear regression is also included in the supplementary material of (Cioba et al. 2021)).

In the case of non-homogeneous tasks, with a mixture of easy and hard tasks, we showed how to optimally allocate data between the two types of tasks. In particular, we found that it is better to allocate more data to easy tasks. This result echoes findings in experimental neuroscience, where it was found that human visual skills indeed transfer better from easy tasks than from hard ones (Ahissar and Hochstein 1997). Our findings provide a guideline for collecting meta-learning data in a way that achieves the best performance under a fixed budget. We do not expect our study to have a negative societal impact, at least not in a direct way.

Overall, our study exemplifies the importance of optimal data allocation in meta-learning and gives a series of empirical and theoretical insights on the relation between model performance and data allocation for MAML. While the behaviour of other meta-learners need not be the same, we surmise that the problem of training models close to optimal allocation is important, and leave much space for empirical study in a variety of contexts, as well as for the development of a more general theoretical framework. For example, we have only scratched the surface of the problem of non-uniform allocation, which requires much further study.

References

Ahissar, M.; and Hochstein, S. 1997. Task difficulty and the specificity of perceptual learning. *Nature*, 387(6631): 401–406.

Bachman, P.; Sordoni, A.; and Trischler, A. 2017. Learning Algorithms for Active Learning. *arXiv:1708.00088 [cs]*. ArXiv: 1708.00088.

Bai, Y.; Chen, M.; Zhou, P.; Zhao, T.; Lee, J. D.; Kakade, S.; Wang, H.; and Xiong, C. 2021. How Important is the Train-Validation Split in Meta-Learning? *arXiv:2010.05843 [cs, stat]*. ArXiv: 2010.05843.

Bernacchia, A. 2021. Meta-learning with negative learning rates. *arXiv:2102.00940 [cs]*. ArXiv: 2102.00940.

Bertinetto, L.; Henriques, J. F.; Torr, P. H. S.; and Vedaldi, A. 2019. Meta-learning with differentiable closed-form solvers. *arXiv:1805.08136 [cs, stat]*. ArXiv: 1805.08136.

Cioba, A.; Bromberg, M.; Wang, Q.; Niyogi, R.; Batzolis, G.; Garcia, J.; Shiu, D.-s.; and Bernacchia, A. 2021. How to distribute data across tasks for meta-learning? *arXiv:2103.08463 [cs]*. ArXiv: 2103.08463.

Collins, L.; Mokhtari, A.; and Shakkottai, S. 2020. Why Does MAML Outperform ERM? An Optimization Perspective. *arXiv:2010.14672 [cs, math, stat]*. ArXiv: 2010.14672.

Denevi, G.; Ciliberto, C.; Stamos, D.; and Pontil, M. 2018. Learning To Learn Around A Common Mean. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 10169–10179. Curran Associates, Inc.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *ICML*, 9.

Du, S. S.; Hu, W.; Kakade, S. M.; Lee, J. D.; and Lei, Q. 2020. Few-Shot Learning via Learning the Representation, Provably. *arXiv:2002.09434 [cs, math, stat]*. ArXiv: 2002.09434.

Fang, M.; Yin, J.; Hall, L. O.; and Tao, D. 2017. Active Multitask Learning With Trace Norm Regularization Based on Excess Risk. *IEEE Transactions on Cybernetics*, 47(11): 3906–3915.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv:1703.03400 [cs]*. ArXiv: 1703.03400.

Finn, C.; Xu, K.; and Levine, S. 2019. Probabilistic Model-Agnostic Meta-Learning. *arXiv:1806.02817 [cs, stat]*. ArXiv: 1806.02817.

Gao, K.; and Sener, O. 2020. Modeling and Optimization Trade-off in Meta-learning. *arXiv:2010.12916 [cs, math, stat]*. ArXiv: 2010.12916.

Garcia, V.; and Bruna, J. 2018. Few-Shot Learning with Graph Neural Networks. *arXiv:1711.04043 [cs, stat]*. ArXiv: 1711.04043.

Harpale, A. 2012. Multi-Task Active Learning. *PhD thesis*, 124.

Hospedales, T.; Antoniou, A.; Micaelli, P.; and Storkey, A. 2020. Meta-Learning in Neural Networks: A Survey. *arXiv:2004.05439 [cs, stat]*. ArXiv: 2004.05439.

Keilson, J. 1967. On global extrema for a class of symmetric functions. *Journal of Mathematical Analysis and Applications*, 18(2): 218–228.

Kim, T.; Yoon, J.; Dia, O.; Kim, S.; Bengio, Y.; and Ahn, S. 2018. Bayesian Model-Agnostic Meta-Learning. *arXiv:1806.03836 [cs, stat]*. ArXiv: 1806.03836.

Kong, W.; Somani, R.; Song, Z.; Kakade, S.; and Oh, S. 2020. Meta-learning for mixed linear regression. *arXiv:2002.08936 [cs, stat]*. ArXiv: 2002.08936.

Li, L.; Jin, X.; Pan, S. J.; and Sun, J.-T. 2012. Multi-domain active learning for text classification. *KDD*, 9.

Pentina, A.; and Lampert, C. H. 2017. Multi-Task Learning with Labeled and Unlabeled Tasks. *arXiv:1602.06518 [cs, stat]*. ArXiv: 1602.06518 version: 2.

Pentina, A.; Sharmanska, V.; and Lampert, C. H. 2015. Curriculum learning of multiple tasks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5492–5500. Boston, MA, USA: IEEE. ISBN 978-1-4673-6964-0.

Ravi, S.; and Larochelle, H. 2017. OPTIMIZATION AS A MODEL FOR FEW-SHOT LEARNING. *ICLR*, 11.

Reichart, R.; Tomanek, K.; Hahn, U.; and Rappoport, A. 2008. Multi-Task Active Learning for Linguistic Annotations. *ACL*, 9.

Requeima, J.; Gordon, J.; Bronskill, J.; Nowozin, S.; and Turner, R. E. 2020. Fast and Flexible Multi-Task Classification Using Conditional Neural Adaptive Processes. *arXiv:1906.07697 [cs, stat]*. ArXiv: 1906.07697.

Ruvolo, P.; and Eaton, E. 2013. Active Task Selection for Lifelong Machine Learning. *AAAI*, 7.

Saha, A.; Rai, P.; Iii, H. D.; and Venkatasubramanian, S. 2011. Online Learning of Multiple Tasks and Their Relationships. *AISTATS*, 9.

Settles, B. 2010. Active Learning Literature Survey. *Technical Report*, 67.

Shekhar, S.; Javidi, T.; and Ghavamzadeh, M. 2020. Adaptive Sampling for Estimating Probability Distributions. *ICML*, 10.

Song, H.; Kim, M.; Park, D.; Shin, Y.; and Lee, J.-G. 2021. Learning from Noisy Labels with Deep Neural Networks: A Survey. *arXiv:2007.08199 [cs, stat]*. ArXiv: 2007.08199.

Sun, G.; Cong, Y.; and Xu, X. 2018. Active Lifelong Learning with “Watchdog”. *AAAI*, 8.

Tripuraneni, N.; Jin, C.; and Jordan, M. I. 2020. Provable Meta-Learning of Linear Representations. *arXiv:2002.11684 [cs, stat]*. ArXiv: 2002.11684.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2017. Matching Networks for One Shot Learning. *arXiv:1606.04080 [cs, stat]*. ArXiv: 1606.04080.

Waterhouse, W. C. 1983. Do Symmetric Problems Have Symmetric Solutions? *The American Mathematical Monthly*, 90(6): 378–387. Publisher: Mathematical Association of America.

Zhang, Y. 2010. Multi-Task Active Learning with Output Constraints. *AAAI*, 6.

Zhang, Z.; Jin, X.; Li, L.; Ding, G.; and Yang, Q. 2016. Multi-Domain Active Learning for Recommendation. *AAAI*, 7.