
Meta-learning for Mixed Linear Regression

Weihao Kong¹ Raghav Somani¹ Zhao Song² Sham Kakade¹ Sewoong Oh¹

Abstract

In modern supervised learning, there are a large number of tasks, but many of them are associated with only a small amount of labelled data. These include data from medical image processing and robotic interaction. Even though each individual task cannot be meaningfully trained in isolation, one seeks to meta-learn across the tasks from past experiences by exploiting some similarities. We study a fundamental question of interest: When can abundant tasks with small data compensate for lack of tasks with big data? We focus on a canonical scenario where each task is drawn from a mixture of k linear regressions, and identify sufficient conditions for such a graceful exchange to hold; there is little loss in sample complexity even when we only have access to small data tasks. To this end, we introduce a novel spectral approach and show that we can efficiently utilize small data tasks with the help of $\tilde{\Omega}(k^{3/2})$ medium data tasks each with $\tilde{\Omega}(k^{1/2})$ examples.

1. Introduction

Recent advances in machine learning highlight successes on a small set of tasks where a large number of labeled examples have been collected and exploited. These include image classification with 1.2 million labeled examples (Deng et al., 2009) and French-English machine translation with 40 million paired sentences (Bojar et al., 2014). For common tasks, however, collecting clean labels is costly, as they require human expertise (as in medical imaging) or physical interactions (as in robotics), for example. Thus collected real-world datasets follow a long-tailed distribution, in which a dominant set of tasks only have a small number of training

¹University of Washington, Seattle, Washington, USA

²Princeton University/Institute for Advanced Study. Correspondence to: Weihao Kong <kweihao@gmail.com>, Raghav Somani <raghavs@cs.washington.edu>, Zhao Song <zhaos@ias.edu>, Sham Kakade <sham@cs.washington.edu>, Sewoong Oh <sewoong@cs.washington.edu>.

examples (Wang et al., 2017).

Inspired by human ingenuity in quickly solving novel problems by leveraging prior experience, *meta-learning* approaches aim to jointly learn from past experience to quickly adapt to new tasks with little available data (Schmidhuber, 1987; Thrun & Pratt, 2012). This has had a significant impact in few-shot supervised learning, where each task is associated with only a few training examples. By leveraging structural similarities among those tasks, one can achieve accuracy far greater than what can be achieved for each task in isolation (Finn et al., 2017; Ravi & Larochelle, 2017; Koch et al., 2015; Oreshkin et al., 2018; Triantafillou et al., 2019; Rusu et al., 2018). The success of such approaches hinges on the following fundamental question: When can we jointly train small data tasks to achieve the accuracy of large data tasks?

We investigate this trade-off under a canonical scenario where the tasks are linear regressions in d -dimensions and the regression parameters are drawn i.i.d. from a discrete set of a support size k . Although widely studied, existing literature addresses the scenario where all tasks have the same fixed number of examples. We defer formal comparisons to Section 6.

On one extreme, when large training data of sample size $\Omega(d)$ is available, each task can easily be learned in isolation; here, $\Omega(k \log k)$ such tasks are sufficient to learn all k regression parameters. This is illustrated by a solid circle in Figure 1. On the other extreme, when each task has only one example, existing approaches require exponentially many tasks (see Table 1). This is illustrated by a solid square.

Several aspects of few-shot supervised learning makes training linear models challenging. The number of training examples varies significantly across tasks, all of which are significantly smaller than the dimension of the data d . The number of tasks are also limited, which restricts any algorithm with exponential sample complexity. An example distribution of such heterogeneous tasks is illustrated in Figure 1 with a bar graph in blue, where both the solid circle and square are far outside of the regime covered by the typical distribution of tasks.

In this data scarce regime, we show that we can still efficiently achieve any desired accuracy in estimating the

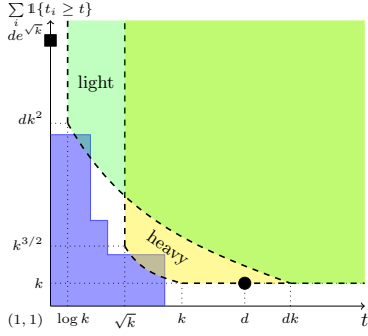


Figure 1. Realistic pool of meta-learning tasks do not include large data tasks (circle) or extremely large number of small data tasks (square), where existing approaches achieve high accuracy. The horizontal axis denotes the number of examples t per task, and the vertical axis denotes the number of tasks in the pool that have at least t examples. The proposed approach succeeds whenever any point in the light (green) region, and any point in the heavy (yellow) region are both covered by the blue bar graph, as is in this example. The blue graph summarizes the pool of tasks in hand, illustrating the cumulative count of tasks with more than t examples. We ignore constants and poly log factors.

meta-parameters defining the meta-learning problem. This is shown in the informal version of our main result in Corollary 1.1. As long as we have enough number of *light tasks* each with $t_L = \tilde{\Omega}(1)$ examples, we can achieve any accuracy with the help of a small number of *heavy tasks* each with $t_H = \tilde{\Omega}(\sqrt{k})$ examples. We only require the total number of examples that we have jointly across all light tasks to be of order $t_L n_L = \tilde{\Omega}(dk^2)$; the number of light tasks n_L and the number of examples per task t_L trade off gracefully. This is illustrated by the green region in Figure 1. Further, we only need a small number of heavy tasks with $t_H n_H = \tilde{\Omega}(k^{3/2})$, shown in the yellow region. As long as the cumulative count of tasks in blue graph intersects with the light (green) and heavy (yellow) regions, we can recover the meta-parameters accurately.

Corollary 1.1 (Special case of Theorem 1, informal).

Given two batches of samples, the first batch with

$$t_L = \tilde{\Omega}(1), t_L n_L = \tilde{\Omega}(dk^2),$$

and the second batch with

$$t_H = \tilde{\Omega}(\sqrt{k}), t_H n_H = \tilde{\Omega}(k^{3/2}),$$

Algorithm 1 estimates the meta-parameters up to any desired accuracy of $\mathcal{O}(1)$ with a high probability, under a certain assumptions on the meta-parameters.

We design a novel spectral approach inspired by (Vempala & Wang, 2004) that first learns a subspace using the light tasks, and then clusters the heavy tasks in the projected space. To get the desired tight bound on the sample complexity, we

improve upon a perturbation bound from (Li & Liang, 2018), and borrow techniques from recent advances in property testing in (Kong et al., 2019).

2. Problem formulation and notations

There are two perspectives on approaching meta-learning: optimization based (Li et al., 2017; Bertinetto et al., 2019; Zhou et al., 2018; Zintgraf et al., 2019; Rajeswaran et al., 2019), and probabilistic (Grant et al., 2018; Finn et al., 2018; Kim et al., 2018; Harrison et al., 2018). Our approach is motivated by the probabilistic view and we present a brief preliminary in Section 2.1. In Section 2.2, we present a simple but canonical scenario where the tasks are linear regressions, which is the focus of this paper.

2.1. Review of probabilistic view on meta-learning

A standard meta-training for few-shot supervised learning assumes that we are given a collection of n meta-training tasks $\{\mathcal{T}_i\}_{i=1}^n$ drawn from some distribution $\mathbb{P}(\mathcal{T})$. Each task is associated with a dataset of size t_i , collectively denoted as a *meta-training dataset* $\mathcal{D}_{\text{meta-train}} = \{ \{(\mathbf{x}_{i,j}, y_{i,j}) \in \mathbb{R}^d \times \mathbb{R}\}_{j \in [t_i]} \}_{i \in [n]}$. Exploiting some structural similarities in $\mathbb{P}(\mathcal{T})$, the goal is to train a model for a new task \mathcal{T}^{new} , coming from $\mathbb{P}(\mathcal{T})$, from a small amount of *training dataset* $\mathcal{D} = \{(\mathbf{x}_j^{\text{new}}, y_j^{\text{new}})\}_{j \in [\tau]}$.

Each task \mathcal{T}_i is associated with a *model parameter* ϕ_i , where the meta-training data is independently drawn from: $(\mathbf{x}_{i,j}, y_{i,j}) \sim \mathbb{P}_{\phi_i}(y|\mathbf{x})\mathbb{P}(\mathbf{x})$ for all $j \in [t_i]$. The prior distribution of the tasks, and hence the model parameters, is fully characterized by a *meta-parameter* θ such that $\phi_i \sim \mathbb{P}_{\theta}(\phi)$.

Following the definition from (Grant et al., 2018), the *meta-learning problem* is defined as estimating the most likely meta-parameter given meta-training data by solving

$$\theta^* \in \arg \max_{\theta} \log \mathbb{P}(\theta | \mathcal{D}_{\text{meta-train}}), \quad (1)$$

which is a special case of empirical Bayes methods for learning the prior distribution from data (Carlin & Louis, 2010). Once meta-learning is done, the model parameter of a newly arriving task can be estimated by a Maximum a Posteriori (MAP) estimator:

$$\hat{\phi} \in \arg \max_{\phi} \log \mathbb{P}(\phi | \mathcal{D}, \theta^*), \quad (2)$$

or a Bayes optimal estimator:

$$\hat{\phi} \in \arg \min_{\phi} \mathbb{E}_{\phi' \sim \mathbb{P}(\phi' | \mathcal{D}, \theta^*)} [\ell(\phi, \phi')], \quad (3)$$

for a choice of a loss function ℓ . This estimated parameter is then used for predicting the label of a new data point \mathbf{x} in task \mathcal{T}^{new} as

$$\hat{y} \in \arg \max_y \mathbb{P}_{\hat{\phi}}(y|\mathbf{x}). \quad (4)$$

General notations. We define $[n] := \{1, \dots, n\} \forall n \in \mathbb{N}$; $\|\mathbf{x}\|_p := (\sum_{x \in \mathbf{x}} |x|^p)^{1/p}$ as the standard ℓ_p -norm; and $B_{p,k}(\boldsymbol{\mu}, r) := \{\mathbf{x} \in \mathbb{R}^k \mid \|\mathbf{x} - \boldsymbol{\mu}\|_p = r\}$. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$, and $\mathbb{1}\{E\}$ denotes the indicator of an event E .

2.2. Linear regression with a discrete prior

In general, the meta-learning problem of (1) is computationally intractable and no statistical guarantees are known. To investigate the trade-offs involved, we assume a simple but canonical scenario where the tasks are linear regressions:

$$\mathbf{x}_{i,j} \sim \mathcal{P}_{\mathbf{x}}, \quad y_{i,j} = \beta_i^\top \mathbf{x}_{i,j} + \epsilon_{i,j}, \quad (5)$$

for the i -th task and j -th example. Each task is associated with a model parameter $\phi_i = (\beta_i \in \mathbb{R}^d, \sigma_i \in \mathbb{R}_+)$. The noise $\epsilon_{i,j}$ is i.i.d. as $\epsilon_{i,j} \sim \mathcal{P}_{\epsilon_i}$, and \mathcal{P}_{ϵ_i} is a centered sub-Gaussian distribution with parameter σ_i^2 . Without loss of generality, we assume that $\mathcal{P}_{\mathbf{x}}$ is an isotropic (i.e. $\mathbb{E}[\mathbf{x}_{i,j} \mathbf{x}_{i,j}^\top] = \mathbf{I}_d$) centered sub-Gaussian distribution. If $\mathcal{P}_{\mathbf{x}}$ is not isotropic, we assume there are large number of $\mathbf{x}_{i,j}$'s for whitening such that $\mathcal{P}_{\mathbf{x}}$ is sufficiently close to isotropic.

We do not make any assumption on the prior of ϕ_i 's other than that they come from a discrete distribution of a support size k . Concretely, the meta-parameter $\theta = (\mathbf{W} \in \mathbb{R}^{d \times k}, \mathbf{s} \in \mathbb{R}_+^k, \mathbf{p} \in \mathbb{R}_+^k \cap B_{1,k}(\mathbf{0}, 1))$ defines a discrete prior (which is also known as mixture of linear experts (Chaganty & Liang, 2013)) on ϕ_i 's, where $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k]$ are the k candidate model parameters, and $\mathbf{s} = [s_1, \dots, s_k]$ are the k candidate noise parameters. The i -th task is randomly chosen from one of the k components from distribution \mathbf{p} , denoted by $z_i \sim \text{multinomial}(\mathbf{p})$. The training data is independently drawn from (5) for each $j \in [t_i]$ with $\beta_i = \mathbf{w}_{z_i}$ and $\sigma_i = s_{z_i}$.

We want to characterize the sample complexity of this meta-learning. This depends on how complex the ground truths prior θ is. This can be measured by the number of components k , the separation between the parameters \mathbf{W} , the minimum mixing probability p_{\min} , and the minimum positive eigen-value λ_{\min} of the matrix $\sum_{j=1}^k p_j \mathbf{w}_j \mathbf{w}_j^\top$.

Notations. We define $\rho_i := \sqrt{s_{z_i}^2 + \|\mathbf{w}_{z_i}\|_2^2}$ as the sub-Gaussian norm of a label $y_{i,j}$ in the i -th task, and $\rho^2 := \max_i \rho_i^2$. Without loss of generality, we assume $\rho = 1$, which can be always achieved by scaling the meta-parameters appropriately. We also define $p_{\min} := \min_{j \in [k]} p_j$, and $\Delta := \min_{i,j \in [k], i \neq j} \|\mathbf{w}_i - \mathbf{w}_j\|_2$ and assume $p_{\min}, \Delta > 0$. $\omega \in \mathbb{R}_+$ is such that two $n \times n$ matrices can be multiplied in $\mathcal{O}(n^\omega)$ time.

3. Algorithm

We propose a novel spectral approach (Algorithm 1) to solve the meta-learning linear regression, consisting of three sub-algorithms: *subspace estimation*, *clustering*, and *classification*. These sub-algorithms require different types of tasks, depending on how many labelled examples are available.

Clustering requires *heavy tasks*, where each task is associated with many labelled examples, but we need a smaller number of such tasks. On the other hand, for subspace estimation and classification, *light tasks* are sufficient, where each task is associated with a few labelled examples. However, we need a large number of such tasks. In this section, we present the intuition behind our algorithm design, and the types of tasks required. Precisely analyzing these requirements is the main contribution of this paper, to be presented in Section 4.

3.1. Intuitions behind the algorithm design

We give a sketch of the algorithm below. Each step of meta-learning is spelled out in full detail in Section 5. This provides an estimated meta-parameter $\hat{\theta} = (\widehat{\mathbf{W}}, \widehat{\mathbf{s}}, \widehat{\mathbf{p}})$. When a new task arrives, this can be readily applied to solve for prediction, as defined in Definition 4.5.

Algorithm 1

Meta-learning

1. *Subspace estimation.* Compute subspace \mathbf{U} which approximates $\text{span}\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$, with singular value decomposition.
2. *Clustering.* Project the heavy tasks onto the subspace of \mathbf{U} , perform distance-based k clustering, and estimate $\tilde{\mathbf{w}}_i$ for each cluster.
3. *Classification.* Perform likelihood-based classification of the light tasks using $\tilde{\mathbf{w}}_i$ estimated from the *Clustering* step, and compute the more refined estimates $(\widehat{\mathbf{w}}_i, \widehat{s}_i, \widehat{p}_i)$ of (\mathbf{w}_i, s_i, p_i) for $i \in [k]$.

Prediction

4. *Prediction.* Perform MAP or Bayes optimal prediction using the estimated meta-parameter as a prior.
-

Subspace estimation. The subspace spanned by the regression vectors, $\text{span}\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$, can be easily estimated using data from the (possibly) light tasks with only $t_i \geq 2$. Using any two independent examples from the same task $(\mathbf{x}_{i,1}, y_{i,1}), (\mathbf{x}_{i,2}, y_{i,2})$, it holds that $\mathbb{E}[y_{i,1} y_{i,2} \mathbf{x}_{i,1} \mathbf{x}_{i,2}^\top] = \sum_{j=1}^k p_j \mathbf{w}_j \mathbf{w}_j^\top$. With a total of $\Omega(d \log d)$ such examples, the matrix $\sum_{j=1}^k p_j \mathbf{w}_j \mathbf{w}_j^\top$ can be accurately estimated under spectral norm, and so is the column space

$\text{span}\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$. We call this step *subspace estimation*.

Clustering. Given an accurate estimation of the subspace $\text{span}\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$, we can reduce the problem from a d -dimensional to a k -dimensional regression problem by projecting \mathbf{x} onto the subspace of \mathbf{U} . Tasks with $t_i = \Omega(k)$ examples can be individually trained as the unknown parameter is now in \mathbb{R}^k . The fundamental question we address is: *What can we do when $t_i = o(k)$?* We propose clustering such light tasks based on their estimates of the regression vector β_i 's, and jointly solve a single regression problem for each cluster.

To this end, we borrow techniques from recent advances in property estimation for linear regression. Recently, in the contextual bandit setting, Kong et al. (2019) proposed an estimator for the correlation between the linear regressors between a pair of datasets. Concretely, given two datasets $\{\mathbf{x}_{1,j}, y_{1,j}\}_{j \in [t]}$ and $\{\mathbf{x}_{2,j}, y_{2,j}\}_{j \in [t]}$ whose true (unknown) regression vectors are β_1 and β_2 , one can estimate $\|\beta_1\|_2^2$, $\|\beta_2\|_2^2$ and $\beta_1^\top \beta_2$ accurately with $t = \mathcal{O}(\sqrt{d})$. We use this technique to estimate $\|\beta_{i_2} - \beta_{i_1}\|_2^2$, whose value can be used to check if the two tasks are in the same clusters. We cluster the tasks with $t_i = \Omega(\sqrt{k})$ into k disjoint clusters. We call this step *clustering*.

After clustering, resulting estimated $\tilde{\mathbf{w}}_i$'s have two sources of error: the error in the subspace estimation, and the error in the parameter estimation for each cluster. If we cluster more heavy tasks, we can reduce the second error but not the first. We could increase the samples used in subspace estimation, but there is a more sample efficient way: classification.

Classification. We start the classification step, once each cluster has enough (i.e. $\Omega(k)$) datapoints to obtain a rough estimation of their corresponding regression vector. In this regime, we have $\mathcal{O}(1)$ error in the estimated $\tilde{\mathbf{w}}_i$'s. This is sufficient for us to add more datapoints to grow each of the clusters. When enough data points are accumulated (i.e. $\tilde{\Omega}(d)$ for each cluster), then we can achieve any desired accuracy with this larger set of accurately classified tasks. This separation of the roles of the three sub-algorithms is critical in achieving the tightest sample complexity.

In contrast to the necessary condition of $t_i = \Omega(\sqrt{k})$ for the clustering step, we show that one can accurately determine which cluster a new task belongs to with only $t_i = \Omega(\log k)$ examples once we have a rough initial estimation $\tilde{\mathbf{W}}$ of the parameter \mathbf{W} . We grow the clusters by adding tasks with a logarithmic number of examples until we have enough data points per cluster to achieve the desired accuracy. We call this step *classification*. This concludes our algorithm for the parameter estimation (i.e. meta-learning) phase.

4. Main results

Suppose we have n_H heavy tasks each with at least t_H training examples, and n_L light tasks each with at least t_L training examples. If heavy tasks are data rich ($t_H \gg d$), we can learn \mathbf{W} straightforwardly from a relatively small number, i.e. $n_H = \Omega(k \log k)$. If the light tasks are data rich ($t_L \gg k$), they can be straightforwardly clustered on the projected k -dimensional subspace. We therefore focus on the following challenging regime of data scarcity.

Assumption 1. *The heavy dataset \mathcal{D}_H consists of n_H heavy tasks, each with at least t_H samples. The first light dataset \mathcal{D}_{L1} consists of n_{L1} light tasks, each with at least t_{L1} samples. The second light dataset \mathcal{D}_{L2} consists of n_{L2} tasks, each with at least t_{L2} samples. We assume $t_{L1}, t_{L2} < k$, and $t_H < d$.*

To give more fine grained analyses on the sufficient conditions, we assume two types of light tasks are available with potentially differing sizes (Remark 4.3). In meta-learning step in Algorithm 1, subspace estimation uses \mathcal{D}_{L1} , clustering uses \mathcal{D}_H , and classification uses \mathcal{D}_{L2} . We provide proofs of the main results in Appendices A, B, and C.

4.1. Meta-learning

We characterize a sufficient condition to achieve a target accuracy ϵ in estimating the meta-parameters $\theta = (\mathbf{W}, \mathbf{s}, \mathbf{p})$.

Theorem 1 (Meta-learning). *For any failure probability $\delta \in (0, 1)$, and accuracy $\epsilon \in (0, 1)$, given three batches of samples under Assumption 1, meta-learning step of Algorithm 1 estimates the meta-parameters with accuracy*

$$\begin{aligned} \|\hat{\mathbf{w}}_i - \mathbf{w}_i\|_2 &\leq \epsilon s_i, \\ |\hat{s}_i^2 - s_i^2| &\leq \frac{\epsilon}{\sqrt{d}} s_i^2, \quad \text{and} \\ |\hat{p}_i - p_i| &\leq \epsilon \sqrt{\frac{t_{L2}}{d}} p_i, \end{aligned}$$

with probability at least $1 - \delta$, if the following holds. The numbers of tasks satisfy

$$\begin{aligned} n_{L1} &= \Omega \left(\frac{d \log^3 \left(\frac{d}{p_{\min} \Delta \delta} \right)}{t_{L1}} \cdot \min \{ \Delta^{-6} p_{\min}^{-2}, \Delta^{-2} \lambda_{\min}^{-2} \} \right), \\ n_H &= \Omega \left(\frac{\log(k/\delta)}{t_H p_{\min} \Delta^2} (k + \Delta^{-2}) \right), \\ n_{L2} &= \Omega \left(\frac{d \log^2(k/\delta)}{t_{L2} p_{\min} \epsilon^2} \right), \end{aligned}$$

and the numbers of samples per task satisfy $t_{L1} \geq 2$, $t_{L2} = \Omega(\log(kd/(p_{\min} \delta \epsilon)) / \Delta^4)$, and $t_H = \Omega(\Delta^{-2} \sqrt{k} \log(k/(p_{\min} \Delta \delta)))$, where λ_{\min} is the smallest non-zero eigen value of $\mathbf{M} := \sum_{j=1}^k p_j \mathbf{w}_j \mathbf{w}_j^\top \in \mathbb{R}^{d \times d}$.

In the following remarks, we explain each of the conditions.

Remark 4.1 (Dependency in \mathcal{D}_{L_1}). *The total number of samples used in subspace estimation is $n_{L_1}t_{L_1}$. The sufficient condition scales linearly in d which matches the information theoretically necessary condition up to logarithmic factors. If the matrix \mathbf{M} is well conditioned, for example when \mathbf{w}_i 's are all orthogonal to each other, subspace estimation is easy, and $n_{L_1}t_{L_1}$ scales as $\Delta^{-2}\lambda_{\min}^{-2}$. Otherwise, the problem gets harder, and we need $\Delta^{-6}p_{\min}^{-2}$ samples. Note that in this regime, tensor decomposition approaches often fails to provide any meaningful guarantee (see Table 1). In proving this result, we improve upon a matrix perturbation bound in (Li & Liang, 2018) to shave off a k^6 factor on n_{L_1} (see Lemma A.12).*

Remark 4.2 (Dependency in \mathcal{D}_H). *The clustering step requires $t_H = \Omega(\sqrt{k})$, which is necessary for distance-based clustering approaches such as single-linkage clustering. From (Kong & Valiant, 2018; Kong et al., 2019) we know that it is necessary (and sufficient) to have $t = \Theta(\sqrt{k})$, even for a simpler testing problem between $\beta_1 = \beta_2$ or $\|\beta_1 - \beta_2\|_2^2 \gg 0$, from two labelled datasets with two linear models β_1 and β_2 .*

Our clustering step is inspired by (Vempala & Wang, 2004) on clustering under Gaussian mixture models, where the algorithm succeeds if $t_H = \Omega(\Delta^{-2}\sqrt{k})$. Although a straightforward adaptation fails, we match the sufficient condition.

We only require the number of heavy samples $n_H t_H$ to be $\tilde{\Omega}(k/p_{\min})$ up to logarithmic factors, which is information theoretically necessary.

Remark 4.3 (Gain of using two types of light tasks). *To get the tightest guarantee, it is necessary to use a different set of light tasks to perform the final estimation step. First notice that the first light dataset \mathcal{D}_{L_1} does not cover the second light dataset since we need $t_{L_2} \geq \Omega(\log(kd))$ which does not need to hold for the first dataset \mathcal{D}_{L_1} . On the other hand, the second light dataset does not cover the first light dataset in the setting where Δ or p_{\min} is very small.*

Remark 4.4 (Dependency in \mathcal{D}_{L_2}). *Classification and prediction use the same routine to classify the given task. Hence, the $\log k$ requirement in t_{L_2} is tight, as it matches our lower bound in Proposition 4.6. The extra terms in the log factor come from the union bound over all n_{L_2} tasks to make sure all the tasks are correctly classified. It is possible to replace it by $\log(1/\epsilon)$ by showing that ϵ fraction of incorrectly classified tasks does not change the estimation by more than ϵ . We only require $n_{L_2}t_{L_2} = \Omega(d/p_{\min})$ up to logarithmic factors, which is information theoretically necessary.*

4.2. Prediction

Given an estimated meta-parameter $\hat{\theta} = (\widehat{\mathbf{W}}, \widehat{\mathbf{s}}, \widehat{\mathbf{p}})$, and a new dataset $\mathcal{D} = \{(\mathbf{x}_j^{\text{new}}, y_j^{\text{new}})\}_{j \in [\tau]}$, we make predictions on the new task with unknown parameters using two

estimators: MAP estimator and Bayes optimal estimator.

Definition 4.5. *Define the maximum a posterior (MAP) estimator as*

$$\widehat{\beta}_{\text{MAP}}(\mathcal{D}) := \widehat{\mathbf{w}}_{\hat{i}}, \quad \text{where } \hat{i} := \arg \max_{i \in [k]} \log \widehat{L}_i, \text{ and}$$

$$\widehat{L}_i := \exp \left(- \sum_{j=1}^{\tau} \frac{(y_j^{\text{new}} - \widehat{\mathbf{w}}_i^\top \mathbf{x}_j^{\text{new}})^2}{2\widehat{s}_i^2} - \tau \log \widehat{s}_i + \log \widehat{p}_i \right).$$

Define the posterior mean estimator as

$$\widehat{\beta}_{\text{Bayes}}(\mathcal{D}) := \frac{\sum_{i=1}^k \widehat{L}_i \widehat{\mathbf{w}}_i}{\sum_{i=1}^k \widehat{L}_i}.$$

If the true prior, $\{(\mathbf{w}_i, s_i, p_i)\}_{i \in [k]}$, is known. The posterior mean estimator achieves the smallest expected squared ℓ_2 error, $\mathbb{E}_{\mathcal{D}, \beta^{\text{new}}} \left[\left\| \widehat{\beta}(\mathcal{D}) - \beta^{\text{new}} \right\|_2^2 \right]$. Hence, we refer to it as Bayes optimal estimator. The MAP estimator maximizes the probability of exact recovery.

Theorem 2 (Prediction). *Under the hypotheses of Theorem 1 with $\epsilon \leq \min \left\{ \Delta/10, \Delta^2 \sqrt{d}/50 \right\}$, the expected prediction errors of both the MAP and Bayes optimal estimators $\widehat{\beta}(\mathcal{D})$ are bound as*

$$\mathbb{E} \left[\left(\mathbf{x}^\top \widehat{\beta}(\mathcal{D}) - y \right)^2 \right] \leq \delta + (1 + \epsilon^2) \sum_{i=1}^k p_i s_i^2, \quad (6)$$

if $\tau \geq \Theta(\log(k/\delta)/\Delta^4)$, where the true meta-parameter is $\theta = \{(\mathbf{w}_i, s_i, p_i)\}_{i=1}^k$, the expectation is over the new task with model parameter $\phi^{\text{new}} = (\beta^{\text{new}}, \sigma^{\text{new}}) \sim \mathbb{P}_\theta$, training dataset $\mathcal{D} \sim \mathbb{P}_{\phi^{\text{new}}}$, and test data $(\mathbf{x}, y) \sim \mathbb{P}_{\phi^{\text{new}}}$.

Note that the $\sum_{i=1}^k p_i s_i^2$ term in (6) is due to the noise in y , and can not be avoided by any estimator. With an accurate meta-learning, we can achieve a prediction error arbitrarily close to this statistical limit, with $\tau = \mathcal{O}(\log k)$. Although both predictors achieve the same guarantee, Bayes optimal estimator achieves smaller training and test errors in Figure 2, especially in challenging regimes with small data.

We show that $\tau = \Omega(\log k)$ training samples are necessary (even if the ground truths meta-parameter θ is known) to achieve error approaching this statistical limit. Let $\Theta_{k, \Delta, \sigma}$ denote the set of all meta-parameters with k components, satisfying $\|\mathbf{w}_i - \mathbf{w}_j\|_2 \geq \Delta$ for $i \neq j \in [k]$ and $s_i \leq \sigma$ for all $i \in [k]$. The following minimax lower bound shows that there exists a threshold scaling as $\mathcal{O}(\log k)$ below which no algorithm can achieve the fundamental limit of σ^2 , which is $\sum_{i=1}^k p_i s_i^2$ in this minimax setting.

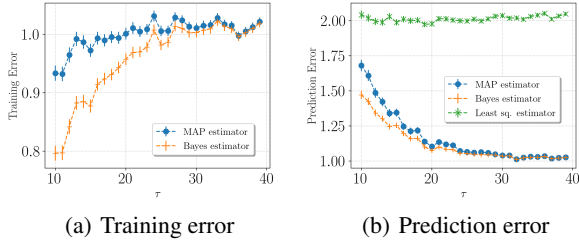


Figure 2. Bayes optimal estimator achieves smaller errors for an example. Here, $k = 32$, $d = 256$, $\mathbf{W}^\top \mathbf{W} = \mathbf{I}_k$, $\mathbf{s} = \mathbf{1}_k$, $\mathbf{p} = \mathbf{1}_k/k$, and $\mathcal{P}_\mathbf{x}$ and \mathcal{P}_ϵ are standard Gaussian distributions. The parameters were learnt using the Meta-learning part of Algorithm 1 as a continuation of simulations discussed in Appendix E, where we provide extensive experiments confirming our analyses.

Remark 4.6 (Lower bound for prediction). *For any $\sigma, \Delta > 0$, if $\tau = ((1 + \Delta^2)/\sigma^2)^{-1} \log(k - 1)$, then*

$$\inf_{\hat{y}} \sup_{\theta \in \Theta_{k, \Delta, \sigma}} \mathbb{E} \left[(\hat{y}(\mathcal{D}, \theta) - y)^2 \right] = \sigma^2 + \Omega(\Delta^2), \quad (7)$$

where the minimization is over all measurable functions of the meta-parameter θ and the training data \mathcal{D} of size τ .

5. Details of the algorithm and the analyses

We explain and analyze each step in Algorithm 1. These analyses imply our main result in meta-learning, which is explicitly written in Appendix A.

5.1. Subspace estimation

In the following, we use $k_SVD(\cdot, k)$ routine that outputs the top k -singular vectors. As $\mathbb{E}[\widehat{\mathbf{M}}] = \mathbf{M} := \sum_{j=1}^k p_j \mathbf{w}_j \mathbf{w}_j^\top$, this outputs an estimate of the subspace spanned by the true parameters. We show that as long as $t_{L1} \geq 2$, the accuracy only depends on the total number of examples, and it is sufficient to have $n_{L1} t_{L1} = \tilde{\Omega}(d)$.

Algorithm 2 Subspace estimation

Input: data $\mathcal{D}_{L1} = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{i \in [n_{L1}], j \in [t_{L1}]}$, $k \in \mathbb{N}$

compute for all $i \in [n_{L1}]$

$$\hat{\beta}_i^{(1)} \leftarrow \frac{2}{t_{L1}} \sum_{j=1}^{t_{L1}/2} y_{i,j} \mathbf{x}_{i,j}, \quad \hat{\beta}_i^{(2)} \leftarrow \frac{2}{t_{L1}} \sum_{j=t_{L1}/2+1}^{t_{L1}} y_{i,j} \mathbf{x}_{i,j}$$

$$\widehat{\mathbf{M}} \leftarrow (2n_{L1})^{-1} \sum_{i=1}^{n_{L1}} \left(\hat{\beta}_i^{(1)} \hat{\beta}_i^{(2)\top} + \hat{\beta}_i^{(2)} \hat{\beta}_i^{(1)\top} \right)$$

$$\mathbf{U} \leftarrow k_SVD(\widehat{\mathbf{M}}, k)$$

output \mathbf{U}

The dependency on the accuracy ϵ changes based on the ground truths meta-parameters. In an ideal case when \mathbf{W} is an orthonormal matrix (with condition number one), the

sample complexity is $\tilde{\mathcal{O}}(d/(p_{\min}^2 \epsilon^2))$. For the worst case \mathbf{W} , it is $\tilde{\mathcal{O}}(d/(p_{\min}^2 \epsilon^6))$.

Lemma 5.1 (Learning the subspace). *Suppose Assumption 1 holds, and let $\mathbf{U} \in \mathbb{R}^{d \times k}$ be the matrix with top k eigen vectors of matrix $\widehat{\mathbf{M}} \in \mathbb{R}^{d \times d}$. For any failure probability $\delta \in (0, 1)$ and accuracy $\epsilon \in (0, 1)$, if the sample size is large enough such that*

$$n_{L1} = \Omega(dt_{L1}^{-1} \cdot \min\{\epsilon^{-6} p_{\min}^{-2}, \epsilon^{-2} \lambda_{\min}^{-2}\} \cdot \log^3(n_{L1} d / \delta)),$$

and $2 \leq t_{L1} < d$, we have

$$\|(\mathbf{U}\mathbf{U}^\top - \mathbf{I})\mathbf{w}_i\|_2 \leq \epsilon, \quad (8)$$

for all $i \in [k]$ with probability at least $1 - \delta$, where λ_{\min} is the smallest non-zero eigen value of $\mathbf{M} := \sum_{i=1}^k p_i \mathbf{w}_i \mathbf{w}_i^\top$.

Time complexity: $\mathcal{O}((n_{L1}^{\omega-1} + n_{L1} t_{L1})d)$ for computing $\widehat{\mathbf{M}}$, and $\mathcal{O}(kd^2)$ for k_SVD (Allen-Zhu & Li, 2016).

5.2. Clustering

Once we have the subspace, we can efficiently cluster any task associated with $t_H = \tilde{\Omega}(\sqrt{k})$ samples. In the following, the matrix $\mathbf{H} \in \mathbb{R}^{n_H \times n_H}$ estimates the distance between the parameters in the projected k -dimensional space. If there is no error in \mathbf{U} , then $\mathbb{E}[\mathbf{H}_{i,j}] \geq \Omega(\Delta^2)$ if i and j are from different components, and zero otherwise. Any clustering algorithm can be applied treating \mathbf{H} as a distance matrix.

Algorithm 3 Clustering and estimation

Input: data $\mathcal{D}_H = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{i \in [n_H], j \in [t_H]}$, $2L \leq t_H$, $k \in \mathbb{N}$, $L \in \mathbb{N}$, $\mathbf{U} \in \mathbb{R}^{d \times k}$

compute for all $\ell \in [L]$ and $i \in [n_H]$

$$\beta_i^{(\ell)} \leftarrow (2L/t_H) \sum_{j=(\ell-1) \cdot (t_H/2L)+1}^{\ell \cdot (t_H/2L)} y_{i,j} \mathbf{x}_{i,j}$$

$$\beta_i^{(\ell+L)} \leftarrow (2L/t_H) \sum_{j=\ell \cdot (t_H/2L)+1}^{2\ell \cdot (t_H/2L)} y_{i,j} \mathbf{x}_{i,j}$$

compute for all $\ell \in [L]$ and $(i, j) \in [n_H] \times [n_H]$

$$\mathbf{H}_{i,j}^{(\ell)} \leftarrow \left(\hat{\beta}_i^{(\ell)} - \hat{\beta}_j^{(\ell)} \right)^\top \mathbf{U}\mathbf{U}^\top \left(\hat{\beta}_i^{(\ell+L)} - \hat{\beta}_j^{(\ell+L)} \right)$$

compute for all $(i, j) \in [n_H] \times [n_H]$

$$\mathbf{H}_{i,j} \leftarrow \text{median}(\{\mathbf{H}_{i,j}^{(\ell)}\}_{\ell \in [L]})$$

Cluster \mathcal{D}_H using \mathbf{H} and return its partition $\{\mathcal{C}_\ell\}_{\ell \in [k]}$

compute for all $\ell \in [k]$

$$\tilde{\mathbf{w}}_\ell \leftarrow (t_H |\mathcal{C}_\ell|)^{-1} \sum_{i \in \mathcal{C}_\ell, j \in [t_H]} y_{i,j} \mathbf{U}\mathbf{U}^\top \mathbf{x}_{i,j}$$

$$\tilde{r}_\ell^2 \leftarrow (t_H |\mathcal{C}_\ell|)^{-1} \sum_{i \in \mathcal{C}_\ell, j \in [t_H]} (y_{i,j} - \mathbf{x}_{i,j}^\top \tilde{\mathbf{w}}_\ell)^2$$

$$\tilde{p}_\ell \leftarrow |\mathcal{C}_\ell| / n_H$$

output $\{\mathcal{C}_\ell, \tilde{\mathbf{w}}_\ell, \tilde{r}_\ell^2, \tilde{p}_\ell\}_{\ell=1}^k$

This is inspired by (Vempala & Wang, 2004), where clustering mixture of Gaussians is studied. One might wonder if it is possible to apply their clustering approach to $\hat{\beta}_i$'s directly. This approach fails as it crucially relies on the

fact that $\|\mathbf{x} - \mu\|_2 = \sqrt{k} \pm \tilde{\mathcal{O}}(1)$ with high probability for $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$. Under our linear regression setting, $\|y\mathbf{x} - \beta\|_2$ does not concentrate. We instead propose median of estimates, to get the desired $t_H = \tilde{\Omega}(\sqrt{k})$ sufficient condition.

Lemma 5.2 (Clustering and initial parameter estimation). *Under Assumption 1, and given an orthonormal matrix $\mathbf{U} \in \mathbb{R}^{d \times k}$ satisfying (8) with any $\epsilon \in (0, \Delta/4)$, Algorithm 3 correctly clusters all tasks with $t_H = \Omega(\Delta^{-2}\sqrt{k} \log(n_H/\delta))$ with probability at least $1 - \delta$, $\forall \delta \in (0, 1)$. Further, if*

$$n_H = \Omega\left(\frac{k \log(k/\delta)}{t_H \tilde{\epsilon}^2 p_{\min}}\right), \quad (9)$$

for any $\tilde{\epsilon} > 0$, with probability at least $1 - \delta$,

$$\|\mathbf{U}^\top(\tilde{\mathbf{w}}_i - \mathbf{w}_i)\|_2^2 \leq \tilde{\epsilon} \quad (10a)$$

$$|\tilde{r}_i^2 - r_i^2| \leq \frac{\tilde{\epsilon}}{\sqrt{k}} r_i^2, \quad (10b)$$

where $r_i^2 := (s_i^2 + \|\tilde{\mathbf{w}}_i - \mathbf{w}_i\|_2^2)$ for all $i \in [k]$.

Time complexity: It takes $\mathcal{O}(n_H dt_H + n_H dk)$ time to compute $\{\mathbf{U}^\top \hat{\beta}_i^{(l)}\}_{i \in [n_H], l \in [L]}$. Then by using matrix multiplication, it takes $\mathcal{O}(n_H^2 k^{\omega-2})$ time to compute the matrix \mathbf{H} , and the single linkage clustering algorithm takes $\mathcal{O}(n_H^2)$ time (Sibson, 1973).

Experiments. We set $d = 8k$, $\mathbf{p} = \mathbf{1}_k/k$, $\mathbf{s} = \mathbf{1}_k$, and \mathcal{P}_x and \mathcal{P}_ϵ are standard Gaussian distributions. Given an estimated subspace with estimation error ~ 0.1 , the clustering step is performed with $n_H = \max\{k^{3/2}, 256\}$ tasks for various t_H . The minimum t_H such that the clustering accuracy is above 99% for at-least $1 - \delta$ fraction of 10 random trials is denoted by $t_{\min}(1 - \delta)$. Figure 3 and Table 3 illustrate the dependence of k on $t_{\min}(0.5)$, and $t_{\min}(0.9)$. More experimental results are provided in Appendix E.

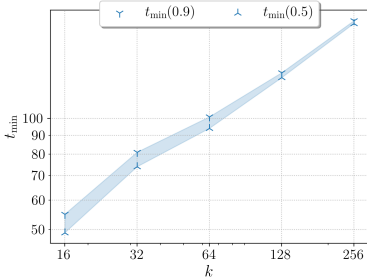


Figure 3. The trends of $t_{\min}(0.9)$ and $t_{\min}(0.5)$ for various k show the $t_H \simeq k^{1/2}$ dependence as predicted by Lemma 5.2 when $n_H = k^{3/2}$.

5.3. Classification

Once we have $\{\tilde{\mathbf{w}}_\ell\}_{\ell=1}^k$ from the clustering step, we can efficiently classify any task with $t_{L2} = \tilde{\Omega}(\log k)$ samples, and an extra $\log n_{L2}$ samples are necessary to apply the union bound. This allows us to use the light samples, in order to refine the clusters estimated with heavy samples. This separation allows us to achieve the desired sample complexity on light tasks ($t_{L2} = \Omega(\Delta^{-4} \log d)$, $n_{L2} t_{L2} p_{\min} = \tilde{\Omega}(\epsilon^{-2} d)$), and heavy tasks ($t_H = \tilde{\Omega}(\Delta^{-2}\sqrt{k})$, $n_H t_H p_{\min} = \tilde{\Omega}(\Delta^{-2} k)$).

In the following, we use Least_Squares(\cdot) routine that outputs the least-squares estimate of all the examples in each cluster. Once each cluster has $\mathcal{O}(d)$ samples, we can accurately estimate the meta-parameters.

Algorithm 4 Classification and estimation

Input: data $\mathcal{D}_{L2} = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{i \in [n_{L2}], j \in [t_{L2}]}$, $\{\mathcal{C}_\ell, \tilde{\mathbf{w}}_\ell, \tilde{r}_\ell^2\}_{\ell \in [k]}$
compute for all $i \in [n_{L2}]$

$$h_i \leftarrow \arg \min_{\ell \in [k]} \frac{1}{2\tilde{r}_\ell^2} \sum_{j \in [t_{L2}]} (y_{i,j} - \mathbf{x}_{i,j}^\top \tilde{\mathbf{w}}_\ell)^2 + t_{L2} \log \tilde{r}_\ell$$

$$\mathcal{C}_{h_i} \leftarrow \mathcal{C}_{h_i} \cup \{(\mathbf{x}_{i,j}, y_{i,j})\}_{j=1}^{t_{L2}}$$

compute for all $\ell \in [k]$,

$$\hat{\mathbf{w}}_\ell \leftarrow \text{Least.Squares}(\mathcal{C}_\ell)$$

$$\hat{s}_\ell^2 \leftarrow (t_{L2} |\mathcal{C}_\ell| - d)^{-1} \sum_{i \in \mathcal{C}_\ell, j \in [t_{L2}]} (y_{i,j} - \mathbf{x}_{i,j}^\top \hat{\mathbf{w}}_\ell)^2$$

$$\hat{p}_\ell \leftarrow |\mathcal{C}_\ell| / n_{L2}$$

output $\{\mathcal{C}_\ell, \hat{\mathbf{w}}_\ell, \hat{s}_\ell^2, \hat{p}_\ell\}_{\ell=1}^k$

Lemma 5.3 (Refined parameter estimation via classification). *Under Assumption 1 and given estimated parameters $\tilde{\mathbf{w}}_i, \tilde{r}_i$ satisfying $\|\tilde{\mathbf{w}}_i - \mathbf{w}_i\|_2 \leq \Delta/10$, $(1 - \Delta^2/50) \tilde{r}_i^2 \leq s_i^2 + \|\tilde{\mathbf{w}}_i - \mathbf{w}_i\|_2^2 \leq (1 + \Delta^2/50) \tilde{r}_i^2$ for all $i \in [k]$ and n_{L2} task with $t_{L2} = \Omega(\log(kn_{L2}/\delta)/\Delta^4)$ examples per task, with probability $1 - \delta$, Algorithm 4 correctly classifies all the n_{L2} tasks. Further, for any $0 < \epsilon \leq 1$ if*

$$n_{L2} = \Omega\left(\frac{d \log^2(k/\delta)}{t_{L2} p_{\min} \epsilon^2}\right), \quad (11)$$

the following holds for all $i \in [k]$,

$$\|\hat{\mathbf{w}}_i - \mathbf{w}_i\|_2 \leq \epsilon s_i, \quad (12a)$$

$$|\hat{s}_i^2 - s_i^2| \leq \frac{\epsilon}{\sqrt{d}} s_i^2, \quad \text{and} \quad (12b)$$

$$|\hat{p}_i - p_i| \leq \epsilon \sqrt{t_{L2}/d} p_i. \quad (12c)$$

Time complexity: Computing $\{h_i\}_{i \in [n_{L2}]}$ takes $\mathcal{O}(n_{L2} t_{L2} dk)$ time, and least square estimation takes $\mathcal{O}(n_{L2} t_{L2} d^{\omega-1})$ time.

Table 1. Sample complexity for previous work in MLR to achieve small constant error on parameters recovery of the mixed linear regression problem. We ignore the constants and poly log factors. Let n , d , and k denote the number of samples, the dimension of the data points, and the number of clusters, respectively. (Yi et al., 2016) and (Chaganty & Liang, 2013) requires σ_k , the k -th singular value of some moment matrix. (Sedghi et al., 2016) requires s_{\min} , the k -th singular value of the matrix of the regression vectors. Note that $1/s_{\min}$ and $1/\sigma_k$ can be infinite even when $\Delta > 0$. (Zhong et al., 2016) algorithm requires $\Delta_{\max}/\Delta_{\min} = \mathcal{O}(1)$ and some spectral properties.

| References | Noise | # Samples n |
|--------------------------|-------|--|
| (CHAGANTY & LIANG, 2013) | YES | $d^6 \cdot \text{poly}(k, 1/\sigma_k)$ |
| (YI ET AL., 2016) | NO | $d \cdot \text{poly}(k, 1/\Delta, 1/\sigma_k)$ |
| (ZHONG ET AL., 2016) | NO | $d \cdot \exp(k \log(k \log d))$ |
| (SEDGHI ET AL., 2016) | YES | $d^3 \cdot \text{poly}(k, 1/s_{\min})$ |
| (LI & LIANG, 2018) | NO | $d \cdot \text{poly}(k/\Delta) + \exp(k^2 \log(k/\Delta))$ |
| (CHEN ET AL., 2020) | NO | $d \cdot \exp(\sqrt{k}) \text{poly}(1/\Delta)$ |

6. Related Work

Meta-learning linear models have been studied in two contexts: mixed linear regression and multi-task learning.

Mixed Linear Regression (MLR). When each task has only one sample, (i.e. $t_i = 1$), the problem has been widely studied. Prior work in MLR are summarized in Table 1. We emphasize that the sample and time complexity of all the previous work either has a super polynomial dependency on k (specifically at least $\exp(\sqrt{k})$) as in (Zhong et al., 2016; Li & Liang, 2018; Chen et al., 2020)), or depends on the inverse of the k -th singular value of some moment matrix as in (Chaganty & Liang, 2013; Yi et al., 2016; Sedghi et al., 2016), which can be infinite. (Chen et al., 2020) cannot achieve vanishing error when there is noise.

Multi-task learning. (Baxter, 2000; Ando & Zhang, 2005; Rish et al., 2008; Orlitsky, 2005) address a similar problem of finding an unknown k -dimensional subspace, where all tasks can be accurately solved. The main difference is that all tasks have the same number of examples, and the performance is evaluated on the observed tasks used in training. Typical approaches use trace-norm to encourage low-rank solutions of the matrix $[\hat{\beta}_1, \dots, \hat{\beta}_n] \in \mathbb{R}^{d \times n}$. This is posed as a convex program (Argyriou et al., 2008; Harchaoui et al., 2012; Amit et al., 2007; Pontil & Maurer, 2013).

Closer to our work is the streaming setting where n tasks are arriving in an online fashion and one can choose how many examples to collect for each. (Balcan et al., 2015) provides an online algorithm using a memory of size only $\mathcal{O}(kn + kd)$, but requires some tasks to have $t_i = \Omega(dk/\epsilon^2)$ examples. In comparison, we only need $t_H = \tilde{\Omega}(\sqrt{k})$ but use $\mathcal{O}(d^2 + kn)$ memory. (Bullins et al., 2019) also use only small memory, but requires $\tilde{\Omega}(d^2)$ total samples to perform the *subspace estimation* under the setting studied in this paper.

Empirical Bayes/Population of parameters. A simple

canonical setting of probabilistic meta-learning is when \mathbb{P}_{ϕ_i} is a univariate distribution (e.g. Gaussian, Bernoulli) and ϕ_i is the parameter of the distribution (e.g. Gaussian mean, success probability). Several related questions have been studied. In some cases, one might be interested in just learning the prior distribution $\mathbb{P}_{\theta}(\phi)$ or the set of ϕ_i 's. For example, if we assume each student's score of one particular exam x_i is a binomial random variable with mean ϕ_i (true score), given the scores of the students in a class, an ETS statistician (Lord, 1969) might want to learn the distribution of their true score ϕ_i 's. Surprisingly, the minimax rate on estimating the prior distribution $\mathbb{P}_{\theta}(\phi)$ was not known until very recently (Tian et al., 2017; Vinayak et al., 2019) even in the most basic setting where $\mathbb{P}_{\phi_i}(x)$ is Binomial.

In some cases, similar to the goal of meta-learning, one might want to accurately estimate the parameter of the new task ϕ^{new} given the new data x^{new} , perhaps by leveraging an estimation of the prior $\mathbb{P}_{\theta}(\phi)$. This has been studied for decades under the *empirical bayes* framework in statistics (see, e.g. the book by Efron (Efron, 2012) for an introduction of the field).

7. Discussion

We investigate how we can meta-learn when we have multiple tasks but each with a small number of labelled examples. This is also known as a few-shot supervised learning setting. When each task is a linear regression, we propose a novel spectral approach and show that we can leverage past experience on small data tasks to accurately learn the meta-parameters and predict new tasks.

When each task is a logistic regression coming from a mixture model, then our algorithm can be applied seamlessly. However, the notion of separation $\Delta = \min_{i \neq j} \|\mathbf{w}_i - \mathbf{w}_j\|_2$ does not capture the dependence on the statistical complexity. Identifying the appropriate notion of complexity on the groundtruths meta-parameters is an interesting research question.

The subspace estimation algorithm requires a total number of $\tilde{\Omega}(dk^2)$ examples. It is worth understanding whether this is also necessary.

Compared to online algorithms for multi-task learning, whose focus is in using small memory, our approach achieves better sample complexity at the expense of requiring a larger memory scaling as $\mathcal{O}(d^2 + kn)$; this is larger than the $\mathcal{O}(kd + kn)$ memory required by (Balcan et al., 2015). An interesting research direction is to achieve the sample complexity of our approach with only $\mathcal{O}(kd + kn)$ memory.

Handling the setting where \mathcal{P}_x has different covariances in different tasks is a challenging problem. There does not seem to exist an unbiased estimator for \mathbf{W} . Nevertheless, (Li & Liang, 2018) study the $t = 1$ case in this setting and come up with an exponential time algorithm. Studying this general setting and coming up with a polynomial time algorithm for meta-learning in a data constrained setting is an interesting direction.

Our clustering algorithm requires the existence of medium data tasks with $t_H = \Omega(\sqrt{k})$ examples per task. It is worth investigating whether there exists a polynomial time and sample complexity algorithms that learns with $t_H = o(\sqrt{k})$. We conjecture that with the techniques developed in the robust clustering literature (Diakonikolas et al., 2018; Hopkins & Li, 2018; Kothari et al., 2018), it is possible to learn with $t_H = o(\sqrt{k})$ in the expense of larger n_H , and higher computation complexity. For a lower bound perspective, it is worth understanding the information theoretic trade-off between t_H and n_H when $t_H = o(\sqrt{k})$.

8. Acknowledgement

Sham Kakade acknowledges funding from the Washington Research Foundation for Innovation in Data-intensive Discovery, and the NSF Awards CCF-1637360, CCF-1703574, and CCF-1740551. Sewoong Oh acknowledges funding from the Google Faculty Research Award, Intel Future Wireless Systems Research Award, and NSF awards CCF-1927712, CCF-1705007, IIS-1929955, and CNS-2002664.

References

Allen-Zhu, Z. and Li, Y. Lazysvd: even faster svd decomposition yet without agonizing pain. In *NIPS*. arXiv:1607.03463, 2016.

Amit, Y., Fink, M., Srebro, N., and Ullman, S. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th international conference on Machine learning*, pp. 17–24, 2007.

Ando, R. K. and Zhang, T. A framework for learning pre-

dictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005.

Argyriou, A., Evgeniou, T., and Pontil, M. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

Balcan, M.-F., Blum, A., and Vempala, S. Efficient representations for lifelong learning and autoencoding. In *Conference on Learning Theory*, pp. 191–210, 2015.

Baxter, J. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

Bertinetto, L., Henriques, J. F., Torr, P. H., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In *ICLR*. arXiv preprint arXiv:1805.08136, 2019.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pp. 12–58, 2014.

Bullins, B., Hazan, E., Kalai, A., and Livni, R. Generalize across tasks: Efficient algorithms for linear representation learning. In *Algorithmic Learning Theory*, pp. 235–246, 2019.

Carlin, B. P. and Louis, T. A. *Bayes and empirical Bayes methods for data analysis*. Chapman and Hall/CRC, 2010.

Chaganty, A. T. and Liang, P. Spectral experts for estimating mixtures of linear regressions. In *International Conference on Machine Learning (ICML)*, pp. 1040–1048, 2013.

Chen, S., Li, J., and Song, Z. Learning mixtures of linear regressions in subexponential time via Fourier moments. In *STOC*. <https://arxiv.org/pdf/1912.07629.pdf>, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Diakonikolas, I., Kane, D. M., and Stewart, A. List-decodable robust mean estimation and learning mixtures of spherical gaussians. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1047–1060, 2018.

Efron, B. *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press, 2012.

- Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1126–1135, 2017.
- Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9516–9527, 2018.
- Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.
- Harchaoui, Z., Douze, M., Paulin, M., Dudik, M., and Malick, J. Large-scale image classification with trace-norm regularization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3386–3393. IEEE, 2012.
- Harrison, J., Sharma, A., and Pavone, M. Meta-learning priors for efficient online bayesian regression. *arXiv preprint arXiv:1807.08912*, 2018.
- Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- Hopkins, S. B. and Li, J. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1021–1034, 2018.
- Hsu, D., Kakade, S. M., and Zhang, T. Random design analysis of ridge regression. In *Conference on learning theory*, pp. 9–1, 2012.
- Kim, T., Yoon, J., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. In *NeurIPS*. *arXiv preprint arXiv:1806.03836*, 2018.
- Koch, G., Zemel, R., and Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015.
- Kong, W. and Valiant, G. Estimating learnability in the sublinear data regime. In *Advances in Neural Information Processing Systems*, pp. 5455–5464, 2018.
- Kong, W., Valiant, G., and Brunskill, E. Sublinear optimal policy value estimation in contextual bandits. *arXiv preprint arXiv:1912.06111*, 2019.
- Kothari, P. K., Steinhardt, J., and Steurer, D. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1035–1046, 2018.
- Li, Y. and Liang, Y. Learning mixtures of linear regressions with nearly optimal complexity. In *COLT*. *arXiv preprint arXiv:1802.07895*, 2018.
- Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.
- Lord, F. M. Estimating true-score distributions in psychological testing (an empirical bayes estimation problem). *Psychometrika*, 34(3):259–299, 1969.
- Oreshkin, B., López, P. R., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.
- Orlitsky, A. Supervised dimensionality reduction using mixture models. In *Proceedings of the 22nd international conference on Machine learning*, pp. 768–775, 2005.
- Pontil, M. and Maurer, A. Excess risk bounds for multitask learning with trace norm regularization. In *Conference on Learning Theory*, pp. 55–76, 2013.
- Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 113–124, 2019.
- Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *International Conference on Representation Learning*, 2017.
- Rish, I., Grabarnik, G., Cecchi, G., Pereira, F., and Gordon, G. J. Closed-form supervised dimensionality reduction with generalized linear models. In *Proceedings of the 25th international conference on Machine learning*, pp. 832–839, 2008.
- Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.
- Schmidhuber, J. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta... hook*. PhD thesis, Technische Universität München, 1987.
- Sedghi, H., Janzamin, M., and Anandkumar, A. Provable tensor methods for learning mixtures of generalized linear models. In *Artificial Intelligence and Statistics (AISTATS)*, pp. 1223–1231, 2016.
- Sibson, R. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1): 30–34, 1973.

- Thrun, S. and Pratt, L. *Learning to learn*. Springer Science & Business Media, 2012.
- Tian, K., Kong, W., and Valiant, G. Learning populations of parameters. In *Advances in Neural Information Processing Systems*, pp. 5778–5787, 2017.
- Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., and Larochelle, H. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.
- Tropp, J. A. et al. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.
- Vempala, S. and Wang, G. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.
- Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.
- Vinayak, R. K., Kong, W., Valiant, G., and Kakade, S. M. Maximum likelihood estimation for learning populations of parameters. *arXiv preprint arXiv:1902.04553*, 2019.
- Wang, Y.-X., Ramanan, D., and Hebert, M. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pp. 7029–7039, 2017.
- Yi, X., Caramanis, C., and Sanghavi, S. Solving a mixture of many random linear equations by tensor decomposition and alternating minimization. *arXiv preprint arXiv:1608.05749*, 2016.
- Zhong, K., Jain, P., and Dhillon, I. S. Mixed linear regression with multiple components. In *Advances in neural information processing systems (NIPS)*, pp. 2190–2198, 2016.
- Zhou, F., Wu, B., and Li, Z. Deep meta-learning: Learning to learn in the concept space. *arXiv preprint arXiv:1802.03596*, 2018.
- Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. In *International Conference on Machine Learning (ICML)*, pp. 7693–7702, 2019.