ISTITUTO ITALIANO DI TECNOLOGIA
Computational Statistics and Machine Learning Department

UNIVERSITÀ DEGLI STUDI DI GENOVA
Mathematics Department

PHD PROGRAM IN MATHEMATICS AND APPLICATIONS

Efficient Lifelong Learning Algorithms: Regret Bounds and Statistical Guarantees

by

Giulia Denevi

Thesis submitted for the degree of *Doctor of Philosophy* (32° cycle)

November 23, 2019

Prof. Massimiliano Pontil — Supervisor
Prof. Stefano Vigni — Head of the PhD program

Thesis Reviewers:
Prof. Nicolò Cesa-Bianchi, *Università degli Studi di Milano* — External examiner
Prof. Francesco Orabona, *Boston University* — External examiner

To my family

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and they have not been submitted, in whole or in part, for any other degree or qualification in this. This dissertation is my own work and it contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and in the Acknowledgments.

Giulia Denevi, November 2019

Acknowledgements

First, I would like to acknowledge my supervisor Massimiliano Pontil for his guidance and for the time he dedicated to me during the PhD. Then, I would like to thank Carlo Ciliberto, Dimitris Stamos and Riccardo Grazzi for their collaboration to the topics in this thesis and for their hospitality at the Imperial College and the University College in London. Finally, I would like to thank my brother Fabio and my parents, Cinzia and Italo, for their love and for their steady support.

Vorrei innanzitutto ringraziare il mio supervisore Massimiliano Pontil per la sua guida e per il tempo che mi ha dedicato durante il dottorato. Vorrei poi ringraziare Carlo Ciliberto, Dimitris Stamos e Riccardo Grazzi per la loro collaborazione agli argomenti in questa tesi e per la loro ospitalità presso Imperial College e University College a Londra. Infine, vorrei ringraziare il mio fratellone Fabio e i miei genitori, Cinzia e Italo, per il loro amore e per il loro costante supporto.

Abstract

We study the *Meta-Learning* paradigm where the goal is to select an algorithm in a prescribed family – usually denoted as *inner* or *within-task* algorithm – that is appropriate to address a class of learning problems (tasks), sharing specific similarities. More precisely, we aim at designing a procedure, called *meta-algorithm*, that is able to infer this tasks' relatedness from a sequence of observed tasks and to exploit such a knowledge in order to return a within-task algorithm in the class that is best suited to solve a *new* similar task.

We are interested in the *online* Meta-Learning setting, also known as *Lifelong Learning*. In this scenario the meta-algorithm receives the tasks sequentially and it incrementally adapts the inner algorithm on the fly as the tasks arrive. In particular, we refer to the framework in which also the within-task data are processed sequentially by the inner algorithm as *Online-Within-Online* (OWO) Meta-Learning, while, we use the term *Online-Within-Batch* (OWB) Meta-Learning to denote the setting in which the within-task data are processed in a single batch.

In this work we propose an OWO Meta-Learning method based on primal-dual Online Learning. Our method is theoretically grounded and it is able to cover various types of tasks' relatedness and learning algorithms. More precisely, we focus on the family of inner algorithms given by a parametrized variant of Follow The Regularized Leader (FTRL) aiming at minimizing the within-task regularized empirical risk. The inner algorithm in this class is incrementally adapted by a FTRL meta-algorithm using the within-task minimum regularized empirical risk as the meta-loss. In order to keep the process fully online, we use the online inner algorithm to approximate the subgradients used by the meta-algorithm and we show how to exploit an upper bound on this approximation error in order to derive a cumulative error bound for the proposed method. Our analysis can be adapted to the statistical setting by two nested online-to-batch conversion steps. We also show how the proposed OWO method can provide statistical guarantees comparable to its natural more expensive OWB variant, where the inner online algorithm is substituted by the batch minimizer of the regularized empirical risk. Finally, we apply our method to two important families of learning algorithms parametrized by a bias vector or a linear feature map.

Contents

List of Figures

List of Symbols

w.r.t.	with respect to
i.i.d.	independently identically distributed
OWO	Online-Within-Online
OWB	Online-Within-Batch
ITL	Independent-Task Learning
MTL	Multi-Task Learning
ERM	Empirical Risk Minimizer
RERM	Regularized Empirical Risk Minimizer
$\mathcal{O}(\cdot)$	limiting behavior when the argument (or some parts of it) tends to infinity
$\mathbb{R}^d$	the d dimensional Euclidean space
$\mathbb{R}^{d\times T}$	the set of the real $d \times T$ matrices
$\mathbb{S}^d$	the set of the real $d \times d$ symmetric matrices
$\mathbb{S}^d_+$	the set of the real $d \times d$ symmetric and positive semi-definite matrices
$\|\cdot\|_2$	the Euclidean or ℓ_2 norm, defined for any $w \in \mathbb{R}^d$ as $\|w\|_2 = \sqrt{\sum_{i=1}^d w_i^2}$
$\|\cdot\|_1$	the ℓ_1 norm, defined for any $w \in \mathbb{R}^d$ as $\|w\|_1 = \sum_{i=1}^d \|w_i\|$
$\|\cdot\|_\infty$	the ℓ_∞ norm, defined for any $w \in \mathbb{R}^d$ as $\|w\|_\infty = \max_{i=1,\dots,d} \|w_i\|$
$\mathrm{Tr}(\cdot)$	the trace operator
$\mathrm{Ran}(\cdot)$	the range operator
$\cdot^\top$	the transpose operator
$\cdot^*$	the conjugate or adjoint operator
$\cdot^\dagger$	the pseudo-inverse operator
$\cdot^{1/2}$	the square root operator
$\Sigma(W)$	the singular values' vector of the matrix $W \in \mathbb{R}^{d\times T}$
$\|\cdot\|_F$	the Frobenius norm, defined for any $W \in \mathbb{R}^{d\times T}$ as $\|W\|_F = \|\Sigma(W)\|_2$
$\|\cdot\|_{\mathrm{Tr}}$	the trace norm, defined for any $W \in \mathbb{R}^{d\times T}$ as $\|W\|_{\mathrm{Tr}} = \|\Sigma(W)\|_1$
$\|\cdot\|_\infty$	the operator norm, defined for any $W \in \mathbb{R}^{d\times T}$ as $\|W\|_\infty = \|\Sigma(W)\|_\infty$
$\iota_{\mathcal{S}}(\cdot)$	the indicator function of the set $\mathcal{S}$, taking value 0 over $\mathcal{S}$ and $+\infty$ otherwise
$\mathcal{V}$	generic Euclidean space, a finite dimensional real vector space

$\langle \cdot, \cdot \rangle$	generic scalar product over $\mathcal{V}$
$\| \cdot \|$	generic norm over $\mathcal{V}$
$\| \cdot \|_*$	the dual norm of $\| \cdot \|$, defined for any $\alpha \in \mathcal{V}$ as $\|\alpha\|_* = \sup_{v \in \mathcal{V}: \|v\| \leq 1} \langle v, \alpha \rangle$
$\mathrm{Dom} f$	the domain of a function f
$\Gamma_0(\mathcal{V})$	the set of the proper, closed and convex functions over $\mathcal{V}$
n	number of within-task points
$i \in \{1, \dots n\}$	within-task index
T	number of tasks
$t \in \{1, \dots, T\}$	outer-task index
$\mathcal{X} \subseteq \mathbb{R}^d$	the input (feature) space
$\mathcal{Y} \subseteq \mathbb{R}$	the output (label) space
$\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$	the data space
z	generic datapoint, $z = (x, y) \in \mathcal{Z}$
$Z = (z_i)_{i=1}^n$	generic dataset of n points, $z_i = (x_i, y_i) \in \mathcal{Z}$
$Z_t = (z_{t,i})_{i=1}^n$	dataset of n points for the task t, $z_{t,i} = (x_{t,i}, y_{t,i}) \in \mathcal{Z}$
$\mathbf{Z} = (Z_t)_{t=1}^T$	meta-dataset, collection of tasks' datasets
μ	probability distribution over $\mathcal{Z}$
$z \sim \mu$	a point z sampled from μ
$Z \sim \mu^n$	a collection of n points i.i.d. according to μ
ρ	meta-distribution over the space of the probability distributions over $\mathcal{Z}$
$\mathcal{M}$	Euclidean space of meta-parameters
$\Theta \subseteq \mathcal{M}$	subset of meta-parameters parametrizing the class of within-task algorithms
$\|\|\|\cdot\|\|\|$	generic norm over $\mathcal{M}$
$\|\|\|\cdot\|\|\|_*$	the dual norm of $\|\|\|\cdot\|\|\|$
A_θ	within-task algorithm associated to the meta-parameter $\theta \in \Theta$
$\mathbf{A}$	meta-algorithm working on the class of within-task algorithms $\{A_\theta : \theta \in \Theta\}$
ℓ	within-task loss function
λ	within-task regularization parameter
$f(\cdot, \theta)$	within-task regularizer parametrized by the meta-parameter $\theta \in \Theta$
$\| \cdot \|_\theta$	a norm over $\mathbb{R}^d$ parametrized by the meta-parameter $\theta \in \Theta$
$\| \cdot \|_{\theta,*}$	the dual norm of $\| \cdot \|_\theta$
$\mathcal{R}_Z$	empirical risk over Z by the loss ℓ, see Eq. (2.1)
$\hat{w}$	minimum norm minimizer of $\mathcal{R}_Z$ over $\mathbb{R}^d$, see below Eq. (2.1)
$\mathcal{R}_{\theta,Z}$	empirical risk $\mathcal{R}_Z$ regularized by $\lambda f(\cdot, \theta)$, see Eq. (2.23)
$\hat{w}_\theta$	minimizer of $\mathcal{R}_{\theta,Z}$ over $\mathbb{R}^d$, see Eq. (2.24)
$\mathcal{E}_Z(A)$	average regret of the within-task algorithm A over Z, see Eq. (2.2)

$\mathcal{E}_{\theta,Z}(A)$	average regret $\mathcal{E}_Z(A)$ regularized by $\lambda f(\cdot,\theta)$, see Eq. (3.5)
$\mathcal{R}_\mu$	(true) risk over μ by the loss ℓ, see Eq. (2.4)
w_μ	minimum norm minimizer of $\mathcal{R}_\mu$ over $\mathbb{R}^d$, see below Eq. (2.4)
$\mathcal{R}_{\theta,\mu}$	(true) risk $\mathcal{R}_\mu$ regularized by $\lambda f(\cdot,\theta)$, see Eq. (3.29)
$w_{\theta,\mu}$	minimizer of $\mathcal{R}_{\theta,\mu}$ over $\mathbb{R}^d$, see Eq. (3.30)
$\mathcal{E}_\mu(A)$	expected excess risk of the within-task algorithm A over μ, see Eq. (2.6)
$\mathcal{E}_{\theta,\mu}(A)$	expected excess risk $\mathcal{E}_\mu(A)$ regularized by $\lambda f(\cdot,\theta)$, see Eq. (3.29)
$(w_{\theta,i})_{i=1}^n$	iterates generated by the online within-task algorithm A_θ over Z
$\bar{w}_\theta$	average of the above iterates, see Eq. (2.5)
$\mathcal{L}_Z$	meta-objective over Z, see Eq. (2.23)
η	meta-regularization parameter
F	meta-regularizer
$(\theta_t)_{t=1}^T$	iterates generated by the online meta-algorithm $\mathbf{A}$ over $\mathbf{Z}$
$\bar{\theta}$	average of the above iterates, see Eq. (2.14)
$\hat{\theta}$	optimal meta-parameter in Θ in the non-statistical setting, see below Eq. (2.11)
θ_ρ	optimal meta-parameter in Θ in the statistical setting, see below Eq. (2.13)
θ_{ITL}	the meta-parameter in Θ corresponding to ITL

Chapter 1

Introduction

We start this introductory chapter by giving the motivation for Lifelong Learning in Sec. 1.1. After that, in Sec. 1.2 and in Sec. 1.3, we briefly summarize the contributions of this work and the list of the publications during the PhD, respectively. Finally, in Sec. 1.4, we describe how this thesis is organized.

1.1 Motivation

Given a collection of datapoints (dataset) Z deriving from a specific problem (task), classic learning systems usually apply a learning algorithm A over the dataset, in order to produce a model $f = A(Z)$ that is able to capture the underlying structure of the data for that problem. However, this procedure can require a large amount of data in order to get a model returning satisfactory performance. This aspect makes classic learning systems rather limited, especially, when it comes to tackle a sequence of learning problems, a situation naturally arising in many real-world scenarios. Overcoming this limitation can have a broad impact in artificial intelligence, as it can save the expensive preparation of large training samples, often humanly annotated, needed by current machine learning methods. In contrast, humans can quickly adapt knowledge gained when learning past tasks, in order to solve novel tasks more efficiently, from just few examples.

All these observations motivated the rise of the so-called *Meta-Learning* or *Learning-To-Learn* paradigm, which has received increasing attention, both from applied (Finn et al., 2017; Ravi and Larochelle, 2017; Thrun and Mitchell, 1995) and theoretical perspective (Alquier et al., 2017;

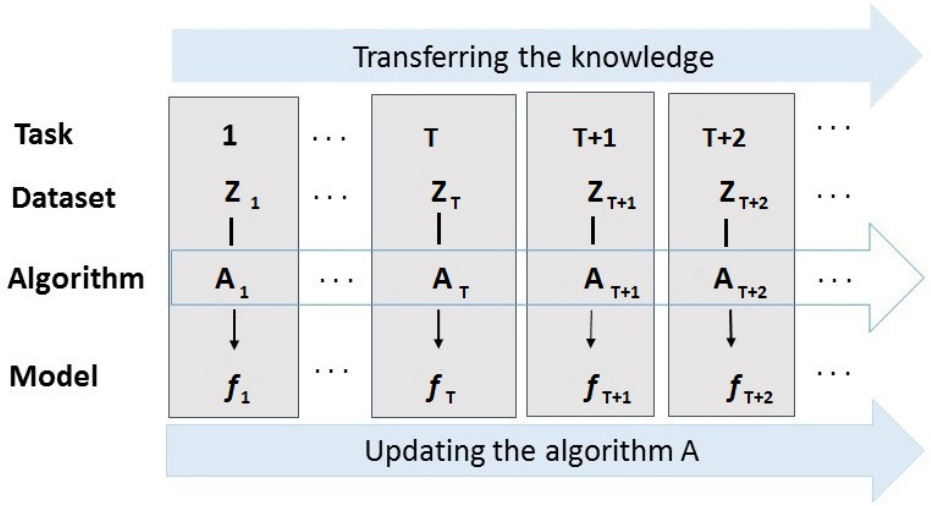

Figure 1.1 Representing Lifelong Learning by an image.

Baxter, 1998; Gupta and Roughgarden, 2017; Maurer, 2005; Maurer et al., 2016; Thrun and Pratt, 1998). In such a framework the aim is to design learning systems that are able to transfer information among several related problems in order to improve the overall performance, instead of building a new model from scratch for every new learning task.

More formally, Meta-Learning aims at designing a procedure, called *meta-algorithm*, able to infer the similarities shared among a class of related problems (tasks), which are only partially observed via a finite collection of training examples (datasets). These similarities are then exploited in order to select an algorithm in a prescribed family that is best suited to address a *new* similar task. Throughout this work, to highlight the difference between the meta-algorithm and an element of the prescribed family, we will refer to the latter as *inner* or *within-task* algorithm. Moreover, as we will see in the following, the choice of a specific class of inner algorithms naturally implies a corresponding type of tasks' relatedness.

The flavor is similar to the motivation behind *Multi-Task Learning* (MTL), see e.g. (Caruana, 1997). The difference is that, in MTL, the idea is to consider all given tasks jointly and transfer information among them in order to select a good algorithm making the learning process for those tasks more efficient, while, in Meta-Learning, the main goal of the learner is not to perform well on the observed tasks, but rather, to extract some information from them that would be useful for solving *new* tasks. Obviously, for this goal to make sense, as described above, one has to assume some relatedness between the observed tasks and the new ones. In other words, from a more practical point of view, in MTL, the performance of the selected inner algorithm is tested on the

same set of tasks used to select such an algorithm, while, in Meta-Learning, the performance is tested on a new yet-to-be-seen task.

A key aspect for the success of the information transfer process is the level of relatedness of the tasks. The strong connection between the two frameworks above have suggested to adapt many relatedness assumptions usually used in the multi-task literature to the Meta-Learning setting. In this work, for instance, we will particularly focus on two classic settings in which the tasks are assumed to have small variance, see (Evgeniou and Pontil, 2004), or to share a low-dimensional linear representation or feature map, see (Argyriou et al., 2008a).

In order to evaluate the effectiveness of a Meta-Learning approach different aspects must be taken into account. A good Meta-Learning approach should be memory efficient, e.g. it should not require to memorize the previous data, it should be time efficient, e.g. it should run in polynomial time with respect to (w.r.t.) the dimension of the problem and the number of data, and, finally, it should be also theoretically grounded. An effective Meta-Learning approach is also expected to bring substantial improvement over learning in isolation – also known as *Independent-Task Learning* (ITL) – when the tasks satisfy the similarity assumption the method is trying to infer from the data and the sample size per task is small, a setting which naturally arises in many applications. Furthermore, when the tasks are not similar as guessed, the method should be able to avoid the so-called *negative tansfer* effect, i.e. it should return performance that are not worse than the performance one would get by solving those tasks independently.

We think that one of the key aspect motivating the theoretical research should be always its applicability to real-world scenarios. In these cases, learning naturally becomes an ongoing (and possibly never ending) process. As a matter of fact, many applications deal with evolving settings, in which data naturally arrive sequentially. We can think for instance to recommendation systems, robotics, autonomous vehicles, financial and weather forecasting or, more generally, to applications stemming from models based on time series. Moreover, in other cases, even when data are available in one entire batch, we may want to process them only few at the time because of limited computational resources.

Even though dealing with learning in an online fashion becomes fundamental from a practical point of view, until very recently, Meta-Learning was mainly studied in the batch statistical setting, where data are assumed to be independently sampled from some distribution and they are processed in one entire batch, see for instance (Baxter, 2000; Maurer, 2009; Maurer et al., 2013, 2016; Pentina and Lampert, 2014). Only recently, a lot of interest raised in investigating more efficient methods, combining ideas from Online Learning and Meta-Learning, see (Alquier et al., 2017; Balcan et al., 2019; Bullins et al., 2019; Denevi et al., 2018a,b, 2019a; Finn et al.,

2019; Pentina and Urner, 2016). In this setting, which is sometimes referred to as *Lifelong Learning*, the tasks are observed sequentially – via corresponding sets of training examples – and the meta-algorithm incrementally updates the inner algorithm on the fly as the tasks arrive. In Fig. 1.1 we report a schematic representation of this process. Key to this setting is for the method to rapidly incorporate new observations into the model as they arrive, without keeping them in memory and without over-fitting them. In other words, the method must be able to avoid the so-called *catastrophic forgetting*, the phenomenon in which the inner algorithm is usually adapted to address only the most recent observations, loosing its effectiveness on the previous ones.

There are different ways to deal with Meta-Learning in an online framework: the so-called *Online-Within-Batch* (OWB) framework, where the tasks are processed online but the data within each task are processed in one batch – see e.g. (Alquier et al., 2017; Balcan et al., 2019; Bullins et al., 2019; Denevi et al., 2018a,b; Finn et al., 2019; Khodak et al., 2019) – or the so-called *Online-Within-Online* (OWO) framework, where data are processed sequentially both within and across the tasks – see e.g. (Alquier et al., 2017; Balcan et al., 2019; Denevi et al., 2019a; Finn et al., 2019; Khodak et al., 2019). In this dissertation we will mainly focus on the most appealing OWO setting, but sometimes we will also discuss about the OWB framework.

In the next section, we describe the main contributions of this work.

1.2 Contributions

Previous work on Meta-Learning mainly analyzed specific settings or gave only a partial study of their methods. The main goal of this work is to propose an OWO Meta-Learning approach that can be adapted to a broad family of standard learning algorithms and to provide a complete analysis for its computational and learning guarantees. We consider both the *non-statistical* setting, where we do not have further assumption on our data, and the *statistical* setting, where data are assumed to be sampled from some distribution. The generality, the provable guarantees and the computational and memory efficiency are strengths of our approach.

In our OWO method, we consider a parametrized class of inner algorithms based on primal-dual Online Learning. Specifically, we discuss in detail the case of Follow The Regularized Leader, where the regularizer belongs to a general family of strongly convex functions parametrized by a meta-parameter. The inner algorithm is adapted by a meta-algorithm, which also consists in applying Follow The Regularized Leader on the meta-objective given by the within-task minimum regularized empirical risk.

The interplay between the meta-algorithm and the inner algorithm plays a key role in our analysis. The latter is used to compute a good approximation of the meta-subgradient which is supplied to the former. A key novelty of our analysis is to show that, exploiting a closed form expression of the error on the meta-subgradients, we can automatically derive a cumulative error bound for the entire procedure in the non-statistical setting, without additional assumptions. Our analysis can be also adapted to the statistical setting by two nested online-to-batch conversions.

We also show how, in the statistical setting, the proposed OWO method can provide comparable guarantees as its more expensive OWB variant, where the inner online algorithm is substituted by the batch minimizer of the regularized empirical risk.

Finally, we show how our general method and the corresponding analysis can be directly applied to two important families of learning algorithms in which the meta-parameter is either a bias vector or a linear feature map shared across the tasks.

1.3 List of Publications

The following main publications were completed during the course of the PhD.

1. *Online-Within-Online Meta-Learning*. G. Denevi, D. Stamos, C. Ciliberto, M. Pontil. Conference on Neural Information Processing Systems (NeurIPS), 2019. See (Denevi et al., 2019b).

2. *Learning-To-Learn Stochastic Gradient Descent with Biased Regularization*. G. Denevi, C. Ciliberto, R. Grazzi, M. Pontil. International Conference on Machine Learning (ICML), 2019. See (Denevi et al., 2019a).

3. *Learning-To-Learn Around A Common Mean*. G. Denevi, C. Ciliberto, D. Stamos, M. Pontil. Conference on Neural Information Processing Systems (NeurIPS), 2018. See (Denevi et al., 2018b).

4. *Incremental Learning-To-Learn with Statistical Guarantees*. G. Denevi, C. Ciliberto, D. Stamos, M. Pontil. Conference on Uncertainty in Artificial Intelligence (UAI), 2018. See (Denevi et al., 2018a).

5. *Iterative Algorithms For a Non-Linear Inverse Problem in Atmospheric LiDAR*. G. Denevi, S. Garbarino, A. Sorrentino. Inverse Problems, 2017. See (Denevi et al., 2017).

This thesis is mainly based on the material taken from the first work above. In such a paper, we present a more general framework which can be specified in order to include also the setting presented in the second work in the list. We will discuss during the thesis how the third and the fourth work fit in the main framework described in this dissertation. Finally, the last paper is not related to the topic presented here, but it comes from the activity developed during my MSc thesis, where I mainly worked on Inverse Problems. For this reason, it will be not mentioned during this dissertation.

1.4 Outline

This dissertation is organized as follows. In Chpt. 2 we start by recalling some background on the standard Online Single-Task Learning setting which leads the basis for the formulation of the OWO Meta-Learning problem, described immediately after. We conclude the chapter by recalling the Multi-Task Learning framework, from which we take inspiration for designing our OWO Meta-Learning method, described and analyzed in the following Chpt. 3. The method is computationally appealing in that it processes the data sequentially both within and across the tasks, without requiring their memorization. At the same time, in the statistical setting, we are able to provide theoretical guarantees for our method which can be comparable to those of its more expensive variant described in Chpt. 4 in which the data within each task are processed in one batch. In Chpt. 5 and Chpt. 6 we show that specializing the above method and the corresponding analysis to two important examples in which the tasks are all close to a common bias vector or share a common simple linear representation, respectively, we get meaningful results, both from the theoretical and the experimental point of view. Finally, in Chpt. 7 we draw conclusion and we discuss future research directions.

The proofs omitted from the main body are postponed to the appendices. In particular, in App. A we provide some basic tools from convex analysis that we use in order to prove our statements and in App. B we describe the material from primal-dual Online Learning that is used to analyze the proposed method. Finally, in App. C we clarify some experimental details regarding the implementation of our method.

In the following chapter we provide the background material used during this dissertation.

Chapter 2

Background

In this chapter we first introduce the standard online Single-Task Learning problem in Sec. 2.1 and then, in parallel way, we formalize the Online-Within-Online Meta-Learning problem in Sec. 2.2. As described in the following, we consider both the non-statistical and the statistical setting. We conclude the chapter in Sec. 2.3 by recalling the Multi-Task Learning framework which inspires the design of our OWO Meta-Learning method in the next Chpt. 3.

2.1 Online Single-Task Learning

In many applications, data are naturally received sequentially and we are required to exploit the information collected up to that moment in order to return a response on the fly. Other times, even when data are available in one batch, we may want to process them only few at the time because of limited computational resources. All these reasons motivate the importance of the study of Online Learning, where the target is usually to design an online algorithm A that makes predictions through time from past information, processing the data sequentially. We refer to (Cesa-Bianchi and Lugosi, 2006; Hazan, 2016; Shalev-Shwartz, 2007; Shalev-Shwartz and Ben-David, 2014; Shalev-Shwartz et al., 2012) and references therein for a detailed discussion about the topic.

More formally, given an input space $\mathcal{X}$ and an output space $\mathcal{Y}$, we will consider the following online problem over the dataspace $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. At each step $i \in \{1, \dots, n\}$:

1. the learner A receives a datapoint $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$,
2. it outputs a label $\hat{y}_i$,

3. it incurs the error $\ell_i(\hat{y}_i) = \ell(\hat{y}_i, y_i)$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function.

To simplify our presentation, throughout this work we let $\mathcal{X} \subseteq \mathbb{R}^d$ (the features' space), $\mathcal{Y} \subseteq \mathbb{R}$ (the labels' space) and we consider algorithms that perform linear predictions of the form $\hat{y}_i = \langle x_i, w_i \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^d$ and $(w_i)_{i=1}^n$ is a sequence of weight vectors which is updated by the algorithm at each iteration during an additional final step:

4. a new weight vector w_{i+1} is computed by the learner.

Precise assumptions are made in the following. In our case, we are especially interested in efficient algorithms which can update the weight vectors $(w_i)_{i=1}^n$ online, with limited time and space capabilities. In the rest of this work, we will denote by $Z = (z_i)_{i=1}^n = (x_i, y_i)_{i=1}^n$ the collection of datapoints processed by our algorithm and, often, in order to simplify the notation, we will not emphasize the dependency of the vectors $(w_i)_{i=1}^n$ on the data Z. Regarding this point, we precise that, by construction, for any $i \in \{1, \dots, n\}$, the vector w_i generated by an online algorithm as above depends only on the datapoints $(z_j)_{j=1}^{i-1}$. We remark also that the process described above is a slightly different variant of the classic Online Learning paradigm, where usually the learner receives the true label y_i only after making the prediction $\hat{y}_i$.

In order to evaluate the performance of the learner above, we have, first of all, to clarify the target problem that we would like to solve. In order to do this, we have to make a distinction according to the nature of our data.

2.1.1 Non-Statistical Setting

In the non-statistical setting we do not have further assumptions on the data processed by the learned. In this case, we define as the problem that we would like to solve the one of minimizing the *empirical risk* associated to the entire batch dataset Z, i.e.

$$\min_{w \in \mathbb{R}^d} \mathcal{R}_Z(w) \qquad \mathcal{R}_Z(w) = \frac{1}{n} \sum_{i=1}^n \ell_i(\langle x_i, w \rangle). \tag{2.1}$$

In the following, we will always assume that the above minimum is in fact attained and we will denote by $\hat{w}$ or $\hat{w}(Z)$ (when we want to emphasize the dependency on Z) the empirical risk minimizer (ERM) with minimum norm. We remark that this choice is made to simplify the presentation, since the bounds we will give in the following will hold for a generic empirical risk minimizer.

In this case, we evaluate the performance of the online learner A described above over the data sequence Z by giving a bound on its *average regret*

$$\mathcal{E}_Z(A) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\langle x_i, w_i \rangle) - \mathcal{R}_Z(\hat{w}), \tag{2.2}$$

which corresponds to the difference between the cumulative error incurred by the vectors $(w_i)_{i=1}^n$ returned by the online learner and the batch minimum, normalized by the number of points n. In other words, from the optimization point of view, we are evaluating how much we are loosing because of processing the points sequentially instead of in one batch.

An online learner A is considered effective when it is able to approach the performance of the target vector $\hat{w}$, as the number of observed points n increases. We can equivalently express this condition by requiring that the average regret of the algorithm in Eq. (2.2) vanishes as the number of points n increases to $+\infty$. In particular, from the efficiency point of view, we are also interested at the speed at which such quantity vanishes. Regarding this aspect, it is well-known that, under appropriate Lipschitz assumptions of the loss function, standard rates for the above average regret are $\mathcal{O}(n^{-1/2})$ when the functions are convex – see e.g. (Shalev-Shwartz, 2007; Shalev-Shwartz and Singer, 2007a; Shalev-Shwartz et al., 2012; Zinkevich, 2003) – and (up to logarithmic factors) $\mathcal{O}(\sigma^{-1}n^{-1})$ when the functions are σ-strongly convex (with $\sigma > 0$) – see e.g. (Hazan et al., 2007; Shalev-Shwartz and Kakade, 2009; Shalev-Shwartz and Singer, 2007b).

Before proceeding, we make the following remark which will be useful for the statistical setting below.

Remark 1 (Weaker Regret)**.** *The above definition in Eq. (2.2) is sometimes called in literature worst case regret, in the sense that we are testing the performance of the online learner in the worst case scenario in which the competitor coincides with the best vector in hindsight $\hat{w}$. However, sometimes in the following, for our aim, it will be sufficient to consider a weaker notion of regret, in which the best vector $\hat{w}$ in Eq. (2.2) is substituted by another fixed vector $w \in \mathbb{R}^d$. Obviously, by definition, we have*

$$\mathcal{E}_Z(A) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\langle x_i, w_i \rangle) - \min_{w\in\mathbb{R}^d} \mathcal{R}_Z(w) = \max_{w\in\mathbb{R}^d} \left\{\frac{1}{n}\sum_{i=1}^{n} \ell_i(\langle x_i, w_i \rangle) - \mathcal{R}_Z(w)\right\}. \tag{2.3}$$

As a consequence, a worst case regret bound automatically translates into a regret bound w.r.t. any fixed competitor $w \in \mathbb{R}^d$.

2.1.2 Statistical Setting

Differently from above, in the statistical setting, the data Z processed by the online learner are assumed to be independently identically distributed (i.i.d.) according to a distribution (or task) μ over the dataspace $\mathcal{Z}$. In the following we will use the short-hand notation $Z \sim \mu^n$ to denote this sampling process. Such a distribution in practice is unknown and it is only partially observed by these i.i.d. samples. In this case, the problem that we would like to solve is to minimize the *(true) risk* associated to the task μ, namely

$$\min_{w \in \mathbb{R}^d} \mathcal{R}_\mu(w) \qquad \mathcal{R}_\mu(w) = \mathbb{E}_{(x,y)\sim\mu}\, \ell(\langle x, w\rangle, y). \tag{2.4}$$

Again, in the following, we will assume that the above minimum is in fact attained and we will denote by w_μ the minimizer with minimum norm.

In this case, given the online learner A described above, we evaluate its performance by giving a bound on the *expected excess risk* over the task μ of the average

$$\bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i \tag{2.5}$$

of the returned vectors $(w_i)_{i=1}^n$, namely,

$$\mathcal{E}_\mu(A) = \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(\bar{w}) - \mathcal{R}_\mu(w_\mu). \tag{2.6}$$

From the statistical point of view, we are evaluating the generalization (predictive) ability of the estimator $\bar{w}$, i.e. the error that we incur on a new point $z = (x, y)$ sampled from μ and independent form Z when we predict $\langle x, \bar{w}\rangle$ instead of the true label y. We stress again that, throughout this work, in order to simplify the notation, we usually do not emphasize the dependency of $\bar{w}$ on Z.

In this case, we say that the online learner A is effective when the average of its iterations is a consistent estimator approaching the performance of the target vector w_μ as the number of points increases. This condition can be equivalently formulated as requiring that the expected excess risk in Eq. (2.6) vanishes as the number of points n increases to $+\infty$. In particular, it is well-known that, under appropriate standard assumptions of the loss function, standard rates for the bound above are $\mathcal{O}(n^{-1/2})$, see e.g. (Shalev-Shwartz et al., 2009; Shamir and Zhang, 2013).

We remark that, obviously, the expected excess risk in Eq. (2.6) can be defined and evaluated at a generic vector $w \in \mathbb{R}^d$. At this point of the discussion the reader may wonder why, in order to evaluate the performance of the online algorithm, we do choose to analyze the expected

excess risk in Eq. (2.6) of the average of its iterations $\bar{w}$, instead of any other vector $w \in \mathbb{R}^d$. The motivation is essentially due to the following standard result linking the performance of the iterations generated by an online learner in the non-statistical setting to the performance of the corresponding average in the statistical setting. This allows us to automatically pass from the non-statistical to the statistical setting, without additional assumptions.

Proposition 1 (Online-To-Batch Conversion, see (Littlestone, 1989))**.** *Consider an online algorithm A that, when applied to a sequence Z of points, returns a sequence of vectors $(w_i)_{i=1}^n$ as described above and denote by $\mathcal{E}_Z(A)$ its average regret defined in Eq. (2.2), where, for any $y \in \mathcal{Y}$, the loss function $\ell(\cdot, y)$ is convex. Then, if the points processed by A are i.i.d. according to a distribution μ over the dataspace $\mathcal{Z}$, we have*

$$\mathcal{E}_\mu(A) \leq \mathbb{E}_{Z\sim\mu^n}\, \mathcal{E}_Z(A), \tag{2.7}$$

where $\mathcal{E}_\mu(A)$ denotes the expected excess risk of the vector $\bar{w}$, the average of the iterations in Eq. (2.6).

Proof. We start from observing that, by definition of $\mathcal{E}_\mu(A)$ and $\mathcal{E}_Z(A)$, the statement above is equivalent to the following

$$\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(\bar{w}) - \mathcal{R}_\mu(w_\mu) \leq \mathbb{E}_{Z\sim\mu^n} \left[\frac{1}{n} \sum_{i=1}^n \ell_i(\langle x_i, w_i \rangle) - \mathcal{R}_Z(\hat{w}) \right]. \tag{2.8}$$

We now observe that, the following relations hold

$$\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(\bar{w}) \leq \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(w_i) = \mathbb{E}_{Z\sim\mu^n} \left[\frac{1}{n} \sum_{i=1}^n \ell_i(\langle x_i, w_i \rangle) \right], \tag{2.9}$$

where, in the inequality we have applied Jensen's inequality (see Lemma 39 in App. A) to the convex function $\mathcal{R}_\mu$ and in the equality we have exploited the relation $\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(w_i) = \mathbb{E}_{Z\sim\mu^n}\, \ell_i(\langle x_i, w_i \rangle)$ (consequence of the fact that, by construction, w_i depends only on the points

$(z_j)_{j=1}^{i-1}$ and $Z \sim \mu^n$). The desired statement derives from the following steps:

$$\begin{aligned}\mathbb{E}_{Z\sim\mu^n}\,\mathcal{R}_\mu(\bar{w}) - \mathcal{R}_\mu(w_\mu) &\leq \mathbb{E}_{Z\sim\mu^n}\left[\frac{1}{n}\sum_{i=1}^n \ell_i(\langle x_i, w_i\rangle) - \mathcal{R}_\mu(w_\mu)\right] \\ &= \mathbb{E}_{Z\sim\mu^n}\left[\frac{1}{n}\sum_{i=1}^n \ell_i(\langle x_i, w_i\rangle) - \mathcal{R}_Z(w_\mu)\right] \\ &\leq \mathbb{E}_{Z\sim\mu^n}\left[\frac{1}{n}\sum_{i=1}^n \ell_i(\langle x_i, w_i\rangle) - \mathcal{R}_Z(\hat{w})\right],\end{aligned} \tag{2.10}$$

where, in the first inequality we have applied Eq. (2.9), the equality is due to the relation $\mathbb{E}_{Z\sim\mu^n}\,\mathcal{R}_Z(w_\mu) = \mathcal{R}_\mu(w_\mu)$ (consequence of the fact that w_μ does not depend on the data Z and $Z \sim \mu^n$), and, finally, the second inequality directly derives from the definition of $\hat{w}$, according to which $\mathcal{R}_Z(\hat{w}) \leq \mathcal{R}_Z(w_\mu)$. ■

The above result in Prop. 1 ensures that whenever we have an online algorithm with an average regret bound, the same bound in expectation w.r.t. the sampling of the data holds in the statistical setting for the expected excess risk of the average of the iterations. In particular, a standard average regret bound $\mathcal{O}(n^{-1/2})$ translates into an equivalent standard excess risk bound for the average in the convex statistical setting.

We observe that, in our case, we consider bounds in expectations. However, combining Martingales' arguments and concentration inequalities, it is possible to get more sophisticate variants of the above result in Prop. 1 that hold in high probability w.r.t. the sampling of the points, see e.g. (Cesa-Bianchi and Gentile, 2006; Cesa-Bianchi et al., 2004). Regarding this aspect, we remark that, for σ-strongly convex functions ($\sigma > 0$), in order to keep the faster rate $\mathcal{O}(\sigma^{-1}n^{-1})$ during the online-to-batch conversion in high probability, it is necessary to apply different concentration inequalities; we refer to (Kakade and Tewari, 2009) for more details about this. In expectation, this problem does not subsist, since the faster rate is automatically maintained, by simply keeping the expectation of the average regret bound.

We also remark that an online-to-batch conversion can be performed when convexity is missing. In such a case, one possible candidate vector in the statistical setting is no more the average, but a vector sampled uniformly from the whole pool of the iterations $(w_i)_{i=1}^n$ generated by the online algorithm. However, in this case, it is necessary to add randomness to the process and, when the number of points is not known in advance, it is also necessary to memorize all the previous vectors in order to perform the uniform sampling. On the contrary, in the convex case, the average of the iterations can be efficiently computed on the fly without memorizing the previous vectors.

Finally, we want to highlight the following fact about the statement in Prop. 1.

Remark 2 (Online-To-Batch Conversion by Weaker Regret)**.** *Looking above at the second row in Eq. (2.10), the reader can immediately note that, in order to have an expected excess risk bound $\mathcal{E}_\mu(A)$ on the average of the iterations, one does not necessarily use a worst case regret bound, but it is sufficient to take the expectation of a weaker regret bound w.r.t. the competitor vector w_μ (see Rem. 1). This fact will be used in the sequel of this work.*

After introducing the standard online Single-Task Learning setting, describing how we tackle the problem by an online algorithm and how we measure its performance, we now are ready to move to the Online-Within-Online Meta-Learning framework.

2.2 Online-Within-Online Meta-Learning

As anticipated in Chpt. 1, in the OWO Meta-Learning setting, we have a family of inner (within-task) online algorithms A_θ identified by a meta-parameter θ, belonging to a prescribed parameter set Θ, and the goal is to adapt the inner algorithm A_θ (i.e. the meta-parameter θ) to a sequence of T online related tasks, in an online fashion. More formally, throughout this work, Θ will be a closed, convex and non-empty subset of an Euclidean space $\mathcal{M}$. The broad goal here is to transfer the information gained when learning previous tasks, in order to help learning future similar tasks. For this purpose, we propose a Meta-Learning procedure $\mathbf{A}$, which will be denoted in the following by *meta-algorithm*, acting across the tasks and modifying the inner algorithm one task after another.

More formally, we assume that all the tasks share the same data space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}$. For each task $t \in \{1, \dots, T\}$, we *sequentially* observe a dataset $Z_t = (z_{t,i})_{i=1}^n = (x_{t,i}, y_{t,i})_{i=1}^n$ [1] where, to simplify the presentation, each dataset Z_t is composed by the same number of points n. At each step $t \in \{1, \dots, T\}$:

1. the meta-learner $\mathbf{A}$ incrementally receives a task dataset Z_t,
2. it runs the inner online algorithm A_{θ_t} with meta-parameter θ_t on Z_t, returning the vectors $(w_{\theta_t,i})_{i=1}^n$,
3. it incrementally incurs the errors $\ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i}\rangle) = \ell(\langle x_{t,i}, w_{\theta_t,i}\rangle, y_{t,i})$ measuring the performance on the task t,

[1] Throughout this work we will use the double subscript notation "$_{t,i}$", to denote the {outer, inner} task index.

4. the meta-parameter θ_t (and consequently the inner algorithm) is updated in θ_{t+1}.

In the rest of this work we will denote by $\mathbf{Z} = (Z_t)_{t=1}^T$ the meta-dataset, i.e. the collection of all the datasets processed by our meta-algorithm $\mathbf{A}$. Moreover, in order to simplify the notation, the dependencies of the meta-parameters $(\theta_t)_{t=1}^T$ on the data will be not made explicit. Regarding this point, we precise that, by construction, for any $t \in \{1, \ldots, T\}$, the meta-parameter θ_t generated by an online meta-algorithm as above depends only on the datasets $(Z_j)_{j=1}^{t-1}$.

In order to evaluate the performance of the meta-algorithm, we have to take into account the performance of the resulting inner algorithms and, as we have seen for the Single-Task Learning setting in Sec. 2.1, this requires to make a distinction according to the nature of our data.

2.2.1 Non-Statistical Setting

We recall that, in the non-statistical setting, as described in the previous Sec. 2.1.1, given an online within-task algorithm A, we evaluate its performance over a data sequence Z by giving a bound on its average regret, which is defined in Eq. (2.2) as

$$\mathcal{E}_Z(A) = \frac{1}{n} \sum_{i=1}^{n} \ell_i(\langle x_i, w_i \rangle) - \mathcal{R}_Z(\hat{w}). \tag{2.2}$$

This suggests to us a very natural way to define the problem we would like to solve as the one of minimizing, over the class of our inner algorithms, the above quantity accumulated by a fixed algorithm over all the datasets. Namely, assuming the existence of the minimum below, the problem we aim to solve in this case is the following

$$\min_{\theta \in \Theta} \frac{1}{T} \sum_{t=1}^{T} \mathcal{E}_{Z_t}(A_\theta). \tag{2.11}$$

This formulation allows us to formally define an optimal algorithm in our class as an algorithm associated to any meta-parameter minimizing the above quantity. Also in this case, when such a meta-parameter is not unique, we apply some choice rule to select one of them. In the following we will denote this representative optimal meta-parameter depending on the data $\mathbf{Z}$ by $\hat{\theta}$.

In this case, given the online meta-algorithm $\mathbf{A}$ described above, we evaluate its performance by analyzing the following quantity

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{Z_t}(A_{\theta_t}) - \min_{\theta\in\Theta}\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{Z_t}(A_\theta). \tag{2.12}$$

In the above equation, we are computing the difference between the average regrets accumulated by the within-task algorithms $(A_{\theta_t})_{t=1}^T$ returned by the meta-learner over the sequence of datasets $\mathbf{Z} = (Z_t)_{t=1}^T$ and the corresponding quantity accumulated by the optimal algorithm in the class. Then, everything is normalized by the number of observed tasks T. Throughout this work, we will refer to the first term above in Eq. (2.12) as *average meta-regret*, while we will denote the second term evaluated at any meta-parameter $\theta \in \Theta$ by *average across-tasks regret*.

An online meta-learner $\mathbf{A}$ is considered to be effective when the overall performance of the returned algorithms $(A_{\theta_t})_{t=1}^T$ is comparable to the performance of the best algorithm in the class $A_{\hat{\theta}}$ when $n \ll T$. This is in fact the effective setting for Meta-Learning. As a matter of fact, in such a case, due to the lack of within-task datapoints, solving each task in isolation is difficult but, on the other hand, the number of observed tasks is sufficiently large to allow our meta-algorithm to learn and to leverage an eventual relatedness among them in order to make the learning process easier for each similar task. The above definition can be roughly reformulated as requiring that, for a fixed value of within-task points n, the corresponding quantity in Eq. (2.12) vanishes as the number of tasks T increases to $+\infty$. However, differently from the Single-Task Learning setting described in Sec. 2.1.1, in this case, at this level of the presentation, we do not manage to establish a standard rate at which the above quantity should usually vanish. As we will see in the following, the rates will vary according to the specific characteristics of the problem. For instance, for the bias setting (see Ex. 1 in Chpt. 3 and Chpt. 5 below), the gap above will vanish as $\mathcal{O}(T^{-1/2})$, for the feature map one (see Ex. 2 in Chpt. 3 and Chpt. 6 below), as $\mathcal{O}(T^{-1/4})$.

One important point to remark is the following. The class of inner algorithms we will consider in our settings will contain also the algorithm that corresponds to solve each task independently (ITL). In the following we will denote by θ_{ITL} the corresponding meta-parameter. This means that, according to the degree of similarity of our tasks, the optimal strategy could be also to solve each task independently. As a consequence, if our meta-algorithm solves the problem according to the formulation above, the corresponding performance can not be worse (at least equivalent) than the one of the ITL algorithm in the class. This means that our meta-learner is not prone to negative transfer.

In the following section, we describe the statistical OWO Meta-Learning setting.

2.2.2 Statistical Setting

Following the framework outlined in (Baxter, 2000; Maurer, 2005), we assume that, for any $t \in \{1, \dots, T\}$, the within-task dataset Z_t is an i.i.d. sample from a distribution μ_t over the data space $\mathcal{Z}$, and, in turn, these distributions $(\mu_t)_{t=1}^T$ are an i.i.d. sample from a meta-distribution ρ which is often called in literature *environment*. Hence, the meta-sequence $\mathbf{Z}$ processed by our meta-algorithm is assumed to be generated by these two nested sampling processes.

In this case, as described in the previous Sec. 2.1.2, given the online within-task algorithm A processing a sequence of points $Z \sim \mu^n$, we evaluate its performance by giving a bound on the expected excess risk of the average $\bar{w}$ of its iterations over that distribution μ, which is defined in Eq. (2.6) as

$$\mathcal{E}_\mu(A) = \mathbb{E}_{Z \sim \mu^n} \mathcal{R}_\mu(\bar{w}) - \mathcal{R}_\mu(w_\mu). \tag{2.6}$$

This suggests to us a very natural way to define the problem we would like to solve as that one of minimizing, over the class of our inner algorithms, the expectation of the above quantity w.r.t. the sampling of the distribution μ from the meta-distribution ρ. Namely, assuming the existence of the minimum below, the problem we aim to solve in this case is the following

$$\min_{\theta \in \Theta} \mathbb{E}_{\mu \sim \rho} \mathcal{E}_\mu(A_\theta). \tag{2.13}$$

This formulation allows us, also in this statistical framework, to formally define an optimal algorithm in our class as an algorithm associated to any meta-parameter minimizing the quantity above. The choice of a single representative optimal parameter $\theta_\rho \in \Theta$ must be intended as described above in the non-statistical framework.

In this case, given the online meta-algorithm $\mathbf{A}$ described above, we consider

$$\bar{\theta} = \frac{1}{T} \sum_{t=1}^T \theta_t, \tag{2.14}$$

the average of its iterations, and we evaluate its performance by analyzing the following quantity

$$\mathbb{E}_{\mathbf{Z}} \mathbb{E}_{\mu \sim \rho} \mathcal{E}_\mu(A_{\bar{\theta}}) - \min_{\theta \in \Theta} \mathbb{E}_{\mu \sim \rho} \mathcal{E}_\mu(A_\theta), \tag{2.15}$$

where the external expectation is w.r.t. the nested sampling of the meta-dataset $\mathbf{Z}$ used by the meta-algorithm to compute the meta-parameter $\bar{\theta}$. In the equation above we are computing the difference between the expected excess risk of the algorithm associated to the meta-parameter

$\bar{\theta}$ estimated by the meta-algorithm and the corresponding quantity accumulated by the optimal algorithm in the class. Then, we take the expectation w.r.t. the sampling $\mu \sim \rho$ and w.r.t. the data used by the meta-algorithm. Throughout this work, we will refer to the first term above in Eq. (2.15) as *expected meta-excess risk*, while we will denote the second term evaluated at any meta-parameter $\theta \in \Theta$ by *expected across-tasks excess risk*.

In this case, we say that an online meta-learner $\mathbf{A}$ is effective when the resulting algorithm $A_{\bar{\theta}}$ provides performance which are comparable to those of the best algorithm in the class A_{θ_ρ}, when $n \ll T$. Again, this can be reformulated in an alternative way requiring that the corresponding quantity in Eq. (2.15) is consistent as $T \to +\infty$, for fixed values of n. The comments on the rates and the negative transfer effect made above for the non-statistical setting hold also for this framework.

The choice of the average of the meta-parameters is motivated by similar reasons as those described in Sec. 2.1.2 aiming at adapting the results from the non-statistical setting to the statistical one, by the application of two nested online-to-batch conversions, one within and one across the tasks. When the inner loss function satisfies the convexity assumption, the within-task conversion is direct, by taking the average of the iterations of the within-task algorithm and applying Jensen's inequality as described in the previous Prop. 1. The delicate step is usually the across-tasks conversion, where the application of Jensen's inequality is usually not allowed because of the lack of convexity w.r.t. the meta-parameter, even when the within-task loss is convex. However, in the following Prop. 10 (in Chpt. 3), we will see that, for our method, relying on appropriate surrogate functions, we will manage to guarantee the convexity w.r.t. the meta-parameter and, in fact, this will allow us to provide guarantees for the average of the meta-parameters, which has, as already observed at the end of Sec. 2.1.2, practical and theoretical advantages w.r.t. other estimators that are usually considered in the non-convex case.

We conclude this section with the following observation regarding the notation.

Remark 3 (Transfer Risk of an Algorithm)**.** *In literature (see e.g. (Maurer, 2005, 2009; Maurer et al., 2013, 2016)), for a generic algorithm A returning a vector $A(Z)$ over a dataset Z, the first term*

$$\mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(A(Z)) \tag{2.16}$$

appearing in the definition of the quantity $\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A)$ is often called transfer risk of the learning algorithm A. In Chpt. 4, we will consider this quantity also for a family of batch inner algorithms.

We now briefly recall the Multi-Task Learning framework which will inspire the design of our OWO Meta-Learning method in the next Chpt. 3.

2.3 Multi-Task Learning

As described above, in the statistical Meta-Learning framework, the goal is to select an algorithm that performs well on a new similar task sampled from the meta-distribution. This can be formally reformulated as solving the problem in Eq. (2.13):

$$\min_{\theta\in\Theta} \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta). \tag{2.13}$$

Instead, in the statistical Multi-Task Learning (MTL), given a prescribed (deterministic) set of tasks $(\mu_t)_{t=1}^T$ and the corresponding i.i.d. datasets $(Z_t)_{t=1}^T$ with $Z_t \sim \mu_t^n$, we aim at selecting an algorithm that is well suited to address such a set of tasks. This goal can be equivalently expressed as the one of minimizing over the class of algorithms the averaged expected excess risk associated to the prescribed set of tasks, i.e.

$$\min_{\theta\in\Theta} \frac{1}{T}\sum_{t=1}^T \mathcal{E}_{\mu_t}(A_\theta), \tag{2.17}$$

where, as usual, for simplicity, the minimum above is assumed to be attained. More explicitly, denoting by $(A_\theta(Z_t))_{t=1}^T$ the linear predictors generated by the algorithm A_θ over the tasks' datasets $(Z_t)_{t=1}^T$, the MTL problem above reads as follows

$$\min_{\theta\in\Theta} \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{Z_t\sim\mu_t^n}\, \mathcal{R}_{\mu_t}(A_\theta(Z_t)) - \mathcal{R}_{\mu_t}(w_{\mu_t}), \tag{2.18}$$

where w_{μ_t} denotes the minimum norm minimizer of the (true) risk $\mathcal{R}_{\mu_t}$ of the task μ_t.

In the discussion above, the predictor vectors for the tasks are generated by applying the same inner algorithm A_θ for each task over the corresponding dataset. We now show that this is not a restrictive assumption, since many Multi-Task Learning methods in literature naturally lead to such a formulation. For this target, we start from observing the following. In order to solve the target MTL problem

$$\min_{W\in\mathbb{R}^{d\times T}} \frac{1}{T}\sum_{t=1}^T \mathcal{R}_{\mu_t}(w_t), \tag{2.19}$$

where $w_t \in \mathbb{R}^d$ denotes the t–th column of the matrix W, a standard approach in literature consists in solving the surrogate empirical problem

$$\min_{W\in\mathbb{R}^{d\times T}} \frac{1}{T}\sum_{t=1}^T \mathcal{R}_{Z_t}(w_t) + \lambda\Omega(W), \tag{2.20}$$

where $\lambda > 0$ is a regularization parameter and Ω is a multi-task regularizer encoding specific similarity assumptions that the tasks are guessed to satisfy and providing the existence of a minimum above. We now observe that most of the widely used multi-task regularizers in literature, see e.g. (Argyriou et al., 2008a; Ciliberto et al., 2015; Jacob et al., 2009), can be reformulated in the following variational formulation

$$\Omega(W) = \min_{\theta \in \Theta} \frac{1}{T} \sum_{t=1}^{T} f(w_t, \theta), \tag{2.21}$$

where Θ is an appropriate set of meta-parameters and f is an appropriate function. As a consequence, exploiting Eq. (2.21), we can rewrite the surrogate empirical problem in Eq. (2.20) as follows

$$\min_{\theta \in \Theta} \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_{Z_t}(\theta), \tag{2.22}$$

where, for any dataset Z and meta-parameter $\theta \in \Theta$, we have introduced the within-task minimum regularized empirical risk

$$\mathcal{L}_Z(\theta) = \min_{w \in \mathbb{R}^d} \mathcal{R}_{\theta,Z}(w) \qquad \mathcal{R}_{\theta,Z}(w) = \mathcal{R}_Z(w) + \lambda f(w, \theta). \tag{2.23}$$

As we will see in the following, f will be an appropriate complexity term ensuring the existence and the uniqueness of the above regularized empirical risk minimizer (RERM)

$$\hat{w}_\theta = \operatorname*{argmin}_{w \in \mathbb{R}^d} \mathcal{R}_{\theta,Z}(w). \tag{2.24}$$

Throughout this work, we will use both the notation $\hat{w}_\theta$ or $\hat{w}_\theta(Z)$ (when we need to stress the dependency on the data). We observe that in Eq. (2.22) the tasks' vectors $(w_t)_{t=1}^T$ act separately and the similarity assumption among the tasks is encoded by the shared meta-parameter $\theta \in \Theta$. Thus, the last formulation in Eq. (2.22) tells us that, in fact, in many multi-task frameworks, the weight predictors of all the tasks are estimated by an appropriate RERM algorithm $\hat{w}_\theta$ as above, applied to the corresponding dataset. This immediately suggests a Meta-Learning framework in which the family of inner algorithms is the one described above associated to the set of meta-parameters Θ and the meta-algorithm attempts to solve the problem in Eq. (2.22), in order to infer a good similarity parameter θ from the data.

One naive approach used in literature to follow this intuition has been the one of casting the observed tasks at every time step T as a multi-task problem. In such a case, the meta-algorithm aims at optimizing Eq. (2.22) on the data observed up to the time T, it uses the inferred meta-parameter θ for solving new tasks and it repeats the process whenever new data arrive. However,

the main drawback of this approach is the fact that it requires re-training a meta-parameter from scratch whenever every new task arrives. In addition, in order to perform the optimization step, it requires memorizing all the encountered datasets. This is obviously prohibitive when we have to face with a long (potentially never ending) stream of tasks. In such cases, one needs to efficiently update the underlying inner algorithm (i.e. the corresponding meta-parameter), in incremental way, as new data arrive, without memorizing the previous data.

In the following Chpt. 3 we will described how to convert this approach into a more efficient OWO Meta-Learning framework in which the data are processed sequentially both within and across the tasks. In particular, we will design a framework in which the meta-algorithm processes the tasks once at the time, using the functions $(\mathcal{L}_{Z_t})_{t=1}^T$ as meta-objectives and the batch family of inner RERM algorithms is substituted by an online counterpart, able to process the data inside each task sequentially.

Before describing in detail our method, we conclude this chapter by giving some well-known examples included in the formulation above. In order to do this, we require some additional notation. We let $\|\cdot\|_2$, $\|\cdot\|_F$, $\|\cdot\|_{\mathrm{Tr}}$, $\|\cdot\|_\infty$, be the Euclidean, Frobenius, trace, and operator norm, respectively. We also let $\cdot^\dagger$ be the pseudo-inverse, $\mathrm{Tr}(\cdot)$ be the trace, $\mathrm{Ran}(\cdot)$ be the range and $\mathbb{S}^d$ (resp. $\mathbb{S}^d_+$) be the set of symmetric (resp. positive semi-definite) matrices in $\mathbb{R}^{d\times d}$. Finally, throughout this work, we will denote by $\langle\cdot,\cdot\rangle$ the standard inner product in $\mathbb{R}^d$ or $\mathbb{S}^d$ and we will consider extended real-valued functions, such as the indicator function $\iota_{\mathcal{S}}$ of a set $\mathcal{S}$, taking value 0 when the argument belongs to $\mathcal{S}$ and $+\infty$ otherwise.

Various assumptions on tasks' relatedness have been exploited in the multi-task literature. One of the simplest assumption is that the target vectors corresponding to the tasks have small variance. This idea was introduced in (Evgeniou and Pontil, 2004) with application to the Support Vector Machines (SVM) problem and, then, adapted also to the online perceptron algorithm in (Cavallanti et al., 2010). In this case, the regularizer enforcing such a relation is the so-called variance regularizer, which can be expressed as in Eq. (2.21) in the following way

$$\Omega(W) = \min_{\theta\in\mathbb{R}^d} \frac{1}{T}\sum_{t=1}^T f(w_t,\theta) \qquad f(w_t,\theta) = \frac{1}{2}\|w_t-\theta\|_2^2. \tag{2.25}$$

The above regularizer enforces the tasks' vectors (i.e. the columns of the matrix W) to stay all closed to a common bias vector $\theta\in\mathbb{R}^d$. We highlight that such a restriction could not be appropriate for scenarios in which the tasks are separate in disjoint groups far away one each other. In such cases more flexible models should be considered, such as the graph regularizer

proposed in (Evgeniou et al., 2005), which requires, however, prior knowledge about the level of similarity between the tasks.

Another widely used assumption on tasks' relatedness is that the predictors for all tasks lay in a low-dimensional subspace, see (Argyriou et al., 2007, 2008a). In this case, the regularizer used to enforce such a relation is the well-known (square) trace norm regularizer, which can be expressed according to the formulation in Eq. (2.21) – see (Argyriou et al., 2008a, Eq. (13)) – as follows

$$\begin{aligned} &\Omega(W) = \|W\|_{\mathrm{Tr}}^2 = \min_{\theta \in \mathcal{S}} \frac{1}{T} \sum_{t=1}^{T} f(w_t, \theta) \\ &f(w_t, \theta) = \frac{1}{2} \langle w_t, \theta^\dagger w_t \rangle + \iota_{\mathrm{Ran}(\theta)}(w_t) + \iota_{\mathcal{S}}(\theta) \qquad \mathcal{S} = \left\{ \theta \in \mathbb{S}_+^d : \mathrm{Tr}(\theta) \leq 1 \right\}. \end{aligned} \tag{2.26}$$

The square root of the above regularizer coincides with the ℓ_1 norm of the singular values' vector of the matrix W and it enforces all the tasks' vectors to stay in the range of a low-rank (more precisely a 1-trace norm) matrix $\theta \in \mathbb{S}_+^d$. In the rest of this work we will refer to the matrix θ as *representation* or *feature map*, because by the change of variable $w_t \mapsto \theta^{1/2} v_t$, with $v_t \in \mathbb{R}^d$, its action can be interpreted as that of a linear feature map acting over the predictors $(w_t)_{t=1}^T$.

A slightly different variant of the relatedness described above is the one according to which all the tasks share a common small subset of significant features, see (Argyriou et al., 2007, 2008a). In this case, the appropriate regularizer to use is the so-called $\ell_{2,1}$ norm

$$\|W\|_{2,1} = \sum_{i=1}^{d} \|w^i\|_2, \tag{2.27}$$

where w^i denotes the i–th row of the matrix W. This regularizer enforces the matrix W to have many rows (those associated to non-significant features) equal to zero and its square can be rewritten in the same variational formulation of $\|W\|_{\mathrm{Tr}}^2$ in Eq. (2.26), by adding to the set $\mathcal{S}$ also the diagonal constraint:

$$\mathcal{S}^{\mathrm{diag}} = \left\{ \theta \in \mathbb{S}_+^d : \mathrm{Tr}(\theta) \leq 1, \theta \text{ is diagonal} \right\}. \tag{2.28}$$

These regularization methods were further extended in order to capture more general types of tasks' relatedness, such as regularizers enforcing disjoint (Argyriou et al., 2008b) or overlapping (Kumar and Daumé III, 2012) grouping effects among the tasks. We mention also the wide family of structured sparsity regularizers (Argyriou et al., 2008c; McDonald et al., 2016; Micchelli et al., 2013), enforcing particular structures on the subspace containing the predictors or on the sparsity

pattern of the significant features. Such regularizers can be expressed by the above variational formulation in Eq. (2.26), by choosing in an appropriate way the set $\mathcal{S}$.

The theoretically grounded success of the above regularizers in the multi-task and transfer learning framework, see e.g. (Kuzborskij and Orabona, 2013, 2017; Maurer, 2006; Maurer et al., 2014; Pontil and Maurer, 2013), motivated the development and the study of Meta-Learning approaches aiming at inferring these various types of tasks' relatedness directly from the data, in a batch or in an online framework. We mention for instance (Balcan et al., 2015; Bullins et al., 2019; Denevi et al., 2018a; Maurer, 2009; Maurer et al., 2013, 2016; Pentina and Lampert, 2014; Ruvolo and Eaton, 2013, 2014), where the authors attempt to infer a low-dimensional representation shared among the tasks, or (Denevi et al., 2018b, 2019a; Pentina and Lampert, 2014), where the goal is to estimate a common bias vector closed to all tasks' predictors.

After this introductory chapter, we now have all the ingredients necessary to describe our OWO Meta-Learning method and to analyze its performance. This is done in the following chapter.

Chapter 3

The Proposed Online-Within-Online Meta-Learning Method

In this chapter we present and analyze our OWO Meta-Learning method. Specifically, after describing the setting in Sec. 3.1, in Sec. 3.2 we give some tools from primal-dual Online Learning which will be necessary for the analysis of the method. After that, in Sec. 3.3 and Sec. 3.4 we analyze the proposed method in the non-statistical and statistical setting, respectively. We conclude the chapter in Sec. 3.5 and Sec. 3.6, where we present a discussion about previous work and our method, respectively. The material of this chapter is taken from our paper (Denevi et al., 2019b).

3.1 Setting

Our OWO Meta-Learning method takes inspiration from the Multi-Task Learning framework described in the former Sec. 2.3. As we have described there, assuming to have all the datasets $\mathbf{Z} = (Z_t)_{t=1}^T$ in hindsight, many MTL methods can be written as in Eq. (2.22):

$$\min_{\theta \in \Theta} \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_{Z_t}(\theta), \tag{2.22}$$

where we recall that, according to Eq. (2.23), for any dataset Z and meta-parameter $\theta \in \Theta$,

$$\mathcal{L}_Z(\theta) = \min_{w \in \mathbb{R}^d} \mathcal{R}_{\theta,Z}(w) \qquad \mathcal{R}_{\theta,Z}(w) = \mathcal{R}_Z(w) + \lambda f(w, \theta), \tag{2.23}$$

with $\lambda > 0$ a regularization parameter and f an appropriate complexity term ensuring the existence and the uniqueness of the above minimizer introduced in Eq. (2.24):

$$\hat{w}_\theta = \operatorname*{argmin}_{w \in \mathbb{R}^d} \mathcal{R}_{\theta,Z}(w). \tag{2.24}$$

In order to deduce our OWO Meta-Learning method, we will consider the following regularized variant of the problem in Eq. (2.22) over the entire meta-parameter space $\mathcal{M}$

$$\min_{\theta \in \mathcal{M}} \sum_{t=1}^{T} \mathcal{L}_{Z_t}(\theta) + \eta F(\theta), \tag{3.1}$$

in which $\eta > 0$ is a meta-regularization parameter and F is an appropriate meta-regularizer ensuring, also in this case, the existence and the uniqueness of the above minimizer. Throughout this work, we will use the short-hand notation $\mathcal{L}_t = \mathcal{L}_{Z_t}$.

We stress again that, in the OWO setting, data are received sequentially, both within and across the tasks. The above formulation inspires us to take a within-task online algorithm that mimics well the batch objective $\mathcal{L}_Z$ in Eq. (2.23) and defining as meta-objectives for the online meta-algorithm the functions $(\mathcal{L}_t)_{t=1}^T$. Obviously, in this setting, the meta-objectives (and consequently their subgradients used by the meta-algorithm) are computed only up to an approximation error, depending on the specific properties of the inner algorithm we are using. In the following, we will show how to control and exploit this approximation error in the analysis.

In the sequel, for an Euclidean space $\mathcal{V}$, we let $\Gamma_0(\mathcal{V})$ be the set of proper, closed and convex functions over $\mathcal{V}$ and, for any $f \in \Gamma_0(\mathcal{V})$, we denote by $\mathrm{Dom} f$ its domain (see App. A and references therein for basic notions on convex analysis). In this work, we make the following standard assumptions about the within-task problem in Eq. (2.23) and the outer-tasks problem in Eq. (3.1). In order to state the following assumptions, we introduce two abstract norms $\|\cdot\|_\theta$ and $|||\cdot|||$ that will be made explicit in two relevant applications below.

Assumption 1 (Loss and Regularizer)**.** *Let $\ell(\cdot, y)$ be a convex and closed real-valued function for any $y \in \mathcal{Y}$ and let $f \in \Gamma_0(\mathbb{R}^d \times \mathcal{M})$ be such that, for any $\theta \in \Theta$, $f(\cdot, \theta)$ is 1-strongly convex w.r.t. a norm $\|\cdot\|_\theta$ over $\mathbb{R}^d$, $\inf_{w \in \mathbb{R}^d} f(w, \theta) = 0$ and, for any $\theta \notin \Theta$, $\mathrm{Dom} f(\cdot, \theta) = \emptyset$.*

Assumption 2 (Meta-Regularizer)**.** *Let F be a closed and 1-strongly convex function w.r.t. a norm $|||\cdot|||$ over $\mathcal{M}$ such that $\inf_{\theta \in \mathcal{M}} F(\theta) = 0$ and $\mathrm{Dom} F = \Theta$.*

Notice that the norm w.r.t. which the function $f(\cdot, \theta)$ is assumed to be strongly convex may vary with θ. Moreover, it is immediate to see that, under Asm. 1, for any $\theta \in \Theta$, the function $\mathcal{R}_{\theta,Z}$

in Eq. (2.23) is proper, closed and λ-strongly convex w.r.t. the norm $\|\cdot\|_\theta$. As a consequence, by Lemma 54 in App. A, we can in fact ensure the existence and the uniqueness of the RERM introduced in Eq. (2.24), for any $\theta \in \Theta$. We also observe that, thanks to Asm. 1, the function $\mathcal{L}_Z$ results to be defined as the partial minimum (w.r.t. the variable w) of a joint function in the variables (w, θ) belonging to $\Gamma_0(\mathbb{R}^d \times \mathcal{M})$. This implies the property $\mathcal{L}_Z \in \Gamma_0(\mathcal{M})$ (see e.g. (Bauschke and Combettes, 2011, Lemma 1.29, Prop. 8.26)), supporting the choice of this function as the meta-objective for our meta-algorithm, and it guarantees the existence and the uniqueness of the minimizer in Eq. (3.1), by similar arguments as the ones made before for $\mathcal{R}_{\theta,Z}$. Furthermore, according to the observations made in the following Rem. 4, the assumptions $\inf_{w\in\mathbb{R}^d} f(w, \theta) = 0$ and $\inf_{\theta\in\mathcal{M}} F(\theta) = 0$ can be made without loss of generality: when these assumptions do not hold, it is sufficient to work with the translated regularizers $f - \inf_{w\in\mathbb{R}^d} f(w, \theta)$ and $F - \inf_{\theta\in\mathcal{M}} F(\theta)$. Finally, we observe that the last requirement in Asm. 1 implies $\Theta = \mathrm{Dom}\mathcal{L}_Z$. This last equivalence is not strictly necessary (to get our results it is sufficient to have $\Theta \subseteq \mathrm{Dom}\mathcal{L}_Z$), but it will simplify the presentation.

We conclude this section by describing in detail two examples included in the framework above and already announced in Sec. 2.3. The first one is inspired by the MTL variance regularizer, see Eq. (2.25) and (Evgeniou et al., 2005), while the second example, which, as already observed, can be easily extended to more general MTL regularizers such as in (Argyriou et al., 2008c; McDonald et al., 2016; Micchelli et al., 2013), relates to the MTL trace norm regularizer, see Eq. (2.26) and (Argyriou et al., 2007, 2008a). As we will see in the following, in the first example the tasks' predictors are encouraged to stay close to a common bias vector, in the second example they are encouraged to lie in the range of a low-rank feature map.

Example 1 (Bias)**.** *We choose* $\mathcal{M} = \Theta = \mathbb{R}^d$, $F(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, *satisfying Asm. 2 with* $|||\cdot||| = \|\cdot\|_2$ *and* $f(\cdot, \theta) = \frac{1}{2}\|\cdot - \theta\|_2^2$, *satisfying Asm. 1 with* $\|\cdot\|_\theta = \|\cdot\|_2$ *for every* $\theta \in \mathbb{R}^d$.

Example 2 (Feature Map)**.** *We choose* $\mathcal{M} = \mathbb{S}^d$, $\Theta = \mathcal{S}$ *where* $\mathcal{S} = \{\theta \in \mathbb{S}^d_+ : \mathrm{Tr}(\theta) \leq 1\}$. *For a fixed* $\theta_0 \in \mathcal{S}$, *we set* $F(\cdot) = \frac{1}{2}\|\cdot - \theta_0\|_F^2 + \iota_{\mathcal{S}}(\cdot)$, *satisfying Asm. 2 with* $|||\cdot||| = \|\cdot\|_F$ *and* $f(\cdot, \theta) = \frac{1}{2}\langle \cdot, \theta^\dagger \cdot\rangle + \iota_{\mathrm{Ran}(\theta)}(\cdot) + \iota_{\mathcal{S}}(\theta)$, *satisfying Asm. 1 with* $\|\cdot\|_\theta = \sqrt{\langle \cdot, \theta^\dagger \cdot\rangle}$ *for any* $\theta \in \mathcal{S}$.

We will return to these examples in Chpt. 5 and Chpt. 6, specializing our method and our analysis to these settings. Before describing in detail the proposed OWO Meta-Learning method, in the next section, we provide some material from primal-dual Online Learning that will be necessary to introduce and analyze the method.

Algorithm 1 Primal-Dual Online Algorithm – Linearized Follow The Regularized Leader

Input $(g_m)_{m=1}^M$, $(A_m)_{m=1}^M$, $(c_m)_{m=1}^M$, $(\epsilon_m)_{m=1}^M$, r as described in the text

Initialization $\alpha_1 = ()$, $v_1 = \nabla r^*(0) \in \text{Dom } r$

For $m = 1$ to M

Receive g_m, A_m, c_{m+1}, ϵ_m

Suffer $g_m(A_m v_m)$ and compute $\alpha'_m \in \partial_{\epsilon_m} g_m(A_m v_m)$

Update $\alpha_{m+1} = (\alpha_m, \alpha'_m)$

Define $v_{m+1} = \nabla r^*\Big(-1/c_{m+1} \sum_{j=1}^m A_j^* \alpha_{m+1,j} \Big) \in \text{Dom } r$

Return $(\alpha_m)_{m=1}^{M+1}$, $(v_m)_{m=1}^{M+1}$

3.2 Preliminaries: Primal-Dual Online Learning

Our OWO Meta-Learning method consists in the application of two nested primal-dual online algorithms, one operating within the tasks and one operating across the tasks. More precisely, in this work, we consider Follow The Regularized Leader algorithm. In this section we briefly recall some material from the primal-dual interpretation of the linearized variant of this algorithm that will be used in our subsequent analysis. The material from this section is an adaptation from (Shalev-Shwartz and Kakade, 2009; Shalev-Shwartz and Singer, 2007a,b; Shalev-Shwartz et al., 2012); we refer to App. B for a more detailed presentation dealing also with the non-linearized variant of the algorithm.

The linearized variant of Follow The Regularized Leader algorithm on a (primal) problem can be derived from the following primal-dual framework in which we introduce an appropriate dual algorithm. Specifically, at each iteration $m \in \{1, \dots, M\}$, we consider the following instantaneous primal optimization problem

$$\hat{P}_{m+1} = \inf_{v \in \mathcal{V}} P_{m+1}(v) \qquad P_{m+1}(v) = \sum_{j=1}^m g_j(A_j v) + c_m r(v) \tag{3.2}$$

where $\mathcal{V}$ is an Euclidean space, $c_m > 0$, $r \in \Gamma_0(\mathcal{V})$ is a 1-strongly convex function w.r.t. a norm $\|\cdot\|$ over $\mathcal{V}$ (with dual norm $\|\cdot\|_*$) such that $\inf_{v \in \mathcal{V}} r(v) = 0$, for any $j \in \{1, \dots, M\}$, letting $\mathcal{V}_j$ an Euclidean space, $g_j \in \Gamma_0(\mathcal{V}_j)$ and $A_j : \mathcal{V} \to \mathcal{V}_j$ is a linear operator with adjoint A_j^*. As explained in App. B, the corresponding dual problem is given by

$$\hat{D}_{m+1} = \inf_{\alpha \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m} D_{m+1}(\alpha) \qquad D_{m+1}(\alpha) = \sum_{j=1}^m g_j^*(\alpha_j) + c_m r^*\Big(-\frac{1}{c_m} \sum_{j=1}^m A_j^* \alpha_j \Big), \tag{3.3}$$

where, g_j^* and r^* are respectively the conjugate functions of g_j and r. After this, we define the dual scheme in which the dual variable α_{m+1} is updated by a greedy coordinate descent approach on the dual, setting $\alpha_{m+1} = (\alpha_m, \alpha'_m)$, where $\alpha'_m \in \partial_{\epsilon_m} g_m(A_m v_m)$ is an ϵ_m-subgradient of g_m at $A_m v_m$ and v_m is the current primal iteration. The primal variable is then updated from the dual one by a variant of the Karush–Kuhn–Tucker (KKT) or optimality conditions, see Alg. 1. It is possible to show that the corresponding primal iterates generated in this way belong to Dom r and they coincide with Follow The Regularized Leader applied to the linearized loss functions $v \mapsto \langle v, A_m^* \alpha'_m \rangle$. We recall that such a scheme includes many well-known algorithms, when one specifies the complexity term r in an appropriate way, see (Shalev-Shwartz et al., 2012) and references therein. We also point out that, in the original papers mentioned above, the author refers to such a scheme using the name *lazy online Mirror Descent*, however, throughout this dissertation, we choose to use the term *linearized Follow The Regularized Leader*, which is historically more accurate.

The behavior of Alg. 1 is analyzed in the next result which will be a key tool for our analysis. The statement is a collection and slightly different adaptation of results from (Shalev-Shwartz and Kakade, 2009; Shalev-Shwartz and Singer, 2007a,b; Shalev-Shwartz et al., 2012). We state it here, since we did not find it in literature in this form. More precisely, the first point of the statement below is an adaptation of (Shalev-Shwartz, 2007, Lemma 1), while, for the second point, we refer to (Shalev-Shwartz and Kakade, 2009, Lemma 5). For the interested reader, we report the proof in App. B.

Theorem 2 (Dual Optimality Gap for Alg. 1)**.** *Let $(v_m)_{m=1}^M$ be the primal iterates returned by the primal-dual online Alg. 1 when applied to the generic problem in Eq. (3.2) and let*

$$\Delta_{\text{Dual}} = D_{M+1}(\alpha_{M+1}) - \hat{D}_{M+1} \tag{3.4}$$

be the corresponding (non-negative) dual optimality gap at the last dual iterate α_{M+1}.

1. *If, for any $m \in \{1, \dots, M\}$, $c_{m+1} \geq c_m$, then,*

$$\Delta_{\text{Dual}} \leq -\sum_{m=1}^{M} g_m(A_m v_m) + \hat{P}_{M+1} + \frac{1}{2} \sum_{m=1}^{M} \frac{1}{c_m} \left\| A_m^* \alpha'_m \right\|_*^2 + \sum_{m=1}^{M} \epsilon_m.$$

2. *If, for any $m \in \{1, \dots, M\}$, $c_m = \sum_{j=1}^m \lambda_j$ for some $\lambda_j > 0$, then,*

$$\Delta_{\text{Dual}} \leq -\sum_{m=1}^{M} \left\{ g_m(A_m v_m) + \lambda_m r(v_m) \right\} + \hat{P}_{M+1} + \frac{1}{2} \sum_{m=1}^{M} \frac{1}{c_m} \left\| A_m^* \alpha'_m \right\|_*^2 + \sum_{m=1}^{M} \epsilon_m.$$

The first (resp. second) inequality in Thm. 2 links the dual optimality gap of the last dual iterate generated by Alg. 1, with the (resp. regularized) cumulative error of the corresponding primal iterates. Note that, as we will stress in the following, this result can be readily used to bound the cumulative error (or its regularized version) of Alg. 1 by the regularized error of the optimal batch comparative $\hat{P}_{M+1}$ and additional terms.

Remark 4 (Non-negativity of the Regularizer)**.** *We point out that, in the setting above, the assumption* $\inf_{v\in\mathcal{V}} r(v) = 0$ *can be made without loss of generality. As a matter of fact, when such assumption does not hold, the framework above applies to the regularizer* $r - \inf_{v\in\mathcal{V}} r(v)$*. In this case, the algorithm is the same reported in Alg. 1 and the bounds given in Thm. 2 must be adapted accordingly, substituting the regularizer* r *with its translated version* $r - \inf_{v\in\mathcal{V}} r(v)$*.*

We now have all the ingredients necessary to describe and analyze our OWO Meta-Learning method. This is done in the following section.

3.3 Method and Analysis in the Non-Statistical Setting

In this section we present the proposed OWO Meta-Learning method and we theoretically analyze its performance in the non-statistical setting. As anticipated in Sec. 3.1, the method consists in the application of Alg. 1 both to the within-task problem in Eq. (2.23) and to the across-tasks problem in Eq. (3.1), corresponding, as we will show in the following, to Alg. 2 and Alg. 3, respectively.

As described in Sec. 2.2.1, we measure the performance of our OWO Meta-Learning method by analyzing the gap in Eq. (2.12)

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{Z_t}(A_{\theta_t}) - \min_{\theta\in\Theta}\ \frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{Z_t}(A_\theta), \tag{2.12}$$

where $(\theta_t)_{t=1}^T$ is the sequence of meta-parameters returned by the meta-learner in Alg. 3 and, for any $\theta \in \Theta$, A_θ is the corresponding inner Alg. 2. We recall that, for an online within-task algorithm A returning the sequence $(w_i)_{i=1}^n$ when applied to a dataset Z, $\mathcal{E}_Z(A)$ is the average regret of its iterations, which is defined in Eq. (2.2) as

$$\mathcal{E}_Z(A) = \frac{1}{n}\sum_{i=1}^{n}\ell_i(\langle x_i, w_i\rangle) - \mathcal{R}_Z(\hat{w}). \tag{2.2}$$

Algorithm 2 Within-Task Algorithm

Input $\lambda > 0$, $\theta \in \Theta$, $Z = (z_i)_{i=1}^n$

Initialization $s_{\theta,1} = ()$, $w_{\theta,1} = \nabla f(\cdot,\theta)^*(0)$

For $i = 1$ to n

Receive the datapoint $z_i = (x_i, y_i)$

Compute $s'_{\theta,i} \in \partial \ell_i(\langle x_i, w_{\theta,i}\rangle) \subseteq \mathbb{R}$

Define $(s_{\theta,i+1})_i = s'_{\theta,i}$, $\gamma_i = \lambda(i+1)$

Update $w_{\theta,i+1} = \nabla f(\cdot,\theta)^*\Big(-1/\gamma_i \sum_{j=1}^{i} x_j s'_{\theta,j}\Big)$

Return $(w_{\theta,i})_{i=1}^{n+1}$, $\bar{w}_\theta = \dfrac{1}{n}\sum_{i=1}^{n} w_{\theta,i}$, $s_{\theta,n+1}$

Algorithm 3 Meta-Algorithm

Input $\eta > 0$, $\mathbf{Z} = (Z_t)_{t=1}^T$

Initialization $\theta_1 = \nabla F^*(0)$

For $t = 1$ to T

Receive incrementally the dataset Z_t

Run Alg. 2 with θ_t over Z_t

Compute $s_{\theta_t,n+1}$

Compute ∇'_{θ_t} as in Prop. 6 using $s_{\theta_t,n+1}$

Update $\theta_{t+1} = \nabla F^*\Big(-1/\eta \sum_{j=1}^{t} \nabla'_{\theta_j}\Big)$

Return $(\theta_t)_{t=1}^{T+1}$, $\bar{\theta} = \dfrac{1}{T}\sum_{t=1}^{T} \theta_t$

In the subsequent analysis, we will investigate about the performance of our method, by comparing two upper bounds for the two terms in Eq. (2.12). More precisely, we will give two stronger (because of the non-negativity of f by Asm. 1) upper bounds holding for the regularized versions of the above quantities, in which, for any inner algorithm A, the average regret $\mathcal{E}_Z(A)$ is substituted by the corresponding regularized version $\mathcal{E}_{\theta,Z}(A)$, defined as

$$\mathcal{E}_{\theta,Z}(A) = \frac{1}{n}\sum_{i=1}^{n}\Big\{\ell_i(\langle x_i, w_i\rangle) + \lambda f(w_i,\theta)\Big\} - \mathcal{R}_Z(\hat{w}). \tag{3.5}$$

In order to give these upper bounds, we start from explaining the origin of the inner Alg. 2 and analyzing its behavior on a single task.

Proposition 3 (Dual Optimality Gap for Alg. 2)**.** *Let Asm. 1 hold. Then, Alg. 2 coincides with the primal-dual online Alg. 1 applied to the non-normalized within-task problem in Eq. (2.23). Let now $(w_{\theta,i})_{i=1}^n$ be the iterates computed by Alg. 2 with meta-parameter $\theta \in \Theta$ over the data sequence $Z = (x_i, y_i)_{i=1}^n$, by means of the subgradients $(s'_{\theta,i})_{i=1}^n$, with $s'_{\theta,i} \in \partial\ell_i(\langle x_i, w_{\theta,i}\rangle)$. Then, $w_{\theta,i} \in \mathrm{Dom} f(\cdot,\theta)$ for any $i \in \{1,\dots,n\}$ and the following upper bound for the associated dual optimality gap Δ_{Dual} introduced in Thm. 2 holds*

$$\Delta_{\mathrm{Dual}} \le \epsilon_\theta \tag{3.6}$$

$$\epsilon_\theta = -n\left(\frac{1}{n}\sum_{i=1}^{n}\Big\{\ell_i(\langle x_i, w_{\theta,i}\rangle) + \lambda f(w_{\theta,i},\theta)\Big\} - \mathcal{L}_Z(\theta)\right) + \frac{1}{2\lambda}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\theta,i}\right\|^2_{\theta,*}. \tag{3.7}$$

As a direct consequence, since by definition Δ_{Dual} is non-negative, the following upper bound holds on the (regularized) average cumulative error of the iterates

$$\frac{1}{n}\sum_{i=1}^{n}\Big\{\ell_i(\langle x_i, w_{\theta,i}\rangle) + \lambda f(w_{\theta,i},\theta)\Big\} \leq \mathcal{L}_Z(\theta) + \frac{1}{2\lambda n}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\theta,i}\right\|^2_{\theta,*}. \tag{3.8}$$

Proof. We first note that the non-normalized within-task problem in Eq. (2.23) is of the form in Eq. (3.2) with $m = M$, once we make the identifications

$$j \rightsquigarrow i,\;\; M \rightsquigarrow n,\;\; v \rightsquigarrow w,\;\; \mathcal{V} \rightsquigarrow \mathbb{R}^d,\;\; g_j \rightsquigarrow \ell_i,\;\; A_j \rightsquigarrow x_i^\top,\;\; c_m \rightsquigarrow n\lambda,\;\; r(\cdot) \rightsquigarrow f(\cdot,\theta).$$

Specifically, adding to the notation the further dependency on $\theta \in \Theta$, we can rewrite the corresponding primal problem in Eq. (3.2) as follows

$$n\,\mathcal{L}_Z(\theta) = \min_{w\in\mathbb{R}^d} P_{n+1}(w,\theta) \qquad P_{n+1}(w,\theta) = \sum_{i=1}^{n}\ell_i(\langle x_i, w\rangle) + \lambda n f(w,\theta). \tag{3.9}$$

Moreover, by the identification $\alpha \rightsquigarrow s$ for the dual variable, the associated dual problem introduced in Eq. (3.3) reads as follows

$$\inf_{s\in\mathbb{R}^n} D_{n+1}(s,\theta) \qquad D_{n+1}(s,\theta) = \sum_{i=1}^{n}\ell_i^*(s_i) + \lambda n f(\cdot,\theta)^*\Big(-\frac{1}{\lambda n}\sum_{i=1}^{n}x_i s_i\Big), \tag{3.10}$$

where ℓ_i^* and $f(\cdot,\theta)^*$ denote the conjugate function of ℓ_i and $f(\cdot,\theta)$, respectively. Thus, exploiting these observations, it is immediate to see that the inner Alg. 2 coincides with Alg. 1 applied to the non-normalized within-task problem in Eq. (2.23), once one makes the identification $\alpha'_m \rightsquigarrow s'_{\theta,i}$ for the (exact) subgradients used by the algorithm. As a consequence, for any $i \in \{1,\dots,n\}$, $w_{\theta,i} \in \text{Dom} f(\cdot,\theta)$ and, in order to get the bound in Eq. (3.6), it is sufficient to combine the second point of Thm. 2 with these observations. Finally, as explained in the statement, the deduction of Eq. (3.8) is direct, by moving the terms and normalizing by the number of points n. ■

Before proceeding, we want to bring to the attention of the reader the following aspect.

Remark 5 (Surrogate Function, Non-Statistical Setting)**.** *Looking at the non-statistical target problem in Eq. (2.11):*

$$\min_{\theta\in\Theta}\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{Z_t}(A_\theta), \tag{2.11}$$

we observe that, for any dataset Z, the dependency on θ in the function $\mathcal{E}_Z(A_\theta)$ is contained only in the first term

$$\frac{1}{n}\sum_{i=1}^{n}\ell_i(\langle x_i, w_{\theta,i}\rangle) \tag{3.11}$$

coinciding with the average cumulative error of the iterates $(w_{\theta,i})_{i=1}^n$ generated by the algorithm A_θ over the dataset Z. At this point of the presentation, the reader may wonder why we do not choose directly such a function as meta-objective for our meta-algorithm. From a practical point of view, the main issue in using the above quantity in Eq. (3.11) as meta-objective is the fact that, very often, such a quantity is non-convex w.r.t. the meta-parameter θ. The above result in Eq. (3.8) tells us that, for a fixed value of λ, when the last term is bounded (corresponding, as we will see in the following, to a Lipschitz assumption for the loss and to a boundedness assumption for the inputs), the inner algorithm attempts to mimic the minimum regularized empirical risk $\mathcal{L}_Z$ in Eq. (2.23), as the number of points n increases. As a consequence, in such a case, the choice of the function $\mathcal{L}_Z$ as meta-objective is reasonable in that such a function can be interpreted as a convex upper bound for the average cumulative error of our inner algorithm.

As explained in the following corollary, from Eq. (3.8) we can immediately deduce the following (regularized) average single-task regret bound for the inner Alg. 2 with appropriate values of $\theta \in \Theta$.

Corollary 4 (Single-Task Regret Bound for Alg. 2)**.** *Let Asm. 1 hold. Fix a dataset $Z = (x_i, y_i)_{i=1}^n$ and consider the minimum norm empirical risk minimizer $\hat{w}$ associated to the dataset Z. For any $\theta \in \Theta$, let A_θ be the corresponding Alg. 2 and let $(w_{\theta,i})_{i=1}^n$ be the iterates computed by A_θ over the data sequence Z, by means of the subgradients $(s'_{\theta,i})_{i=1}^n$, with $s'_{\theta,i} \in \partial\ell_i(\langle x_i, w_{\theta,i}\rangle)$. Then, for any $\theta \in \Theta$ such that $f(\hat{w}, \theta) < +\infty$, the following (regularized) average regret bound holds for the above iterates*

$$\mathcal{E}_{\theta,Z}(A_\theta) \leq \lambda f(\hat{w}, \theta) + \frac{1}{2\lambda n}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\theta,i}\right\|_{\theta,*}^2. \tag{3.12}$$

Proof. We start from observing that, for any $\theta \in \Theta$ such that $f(\hat{w}, \theta) < +\infty$, the following inequalities hold

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\left\{\ell_i(\langle x_i, w_{\theta,i}\rangle) + \lambda f(w_{\theta,i}, \theta)\right\} &\leq \mathcal{L}_Z(\theta) + \frac{1}{2\lambda n}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\theta,i}\right\|_{\theta,*}^2 \\ &\leq \mathcal{R}_Z(\hat{w}) + \lambda f(\hat{w}, \theta) + \frac{1}{2\lambda n}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\theta,i}\right\|_{\theta,*}^2,\end{aligned} \tag{3.13}$$

where in the first inequality we have applied Eq. (3.8) and in the second inequality we have exploited the definition of $\mathcal{L}_Z$ in Eq. (2.23) as the minimum of the regularized empirical risk. The desired statement derives from the last inequality above by moving the term $\mathcal{R}_Z(\hat{w})$ on the left. ■

In the statistical setting described in the subsequent Sec. 3.4, we will exploit the following observation.

Remark 6 (Weaker Single-Task Regret Bound for Alg. 2)**.** *Looking at Eq. (3.13) in the proof above, the reader can immediately see that the worst case regret bound given in Cor. 4 can be immediately stated in the following weaker form (see Rem. 1) for a generic competitor vector* $w \in \mathbb{R}^d$*:*

$$\frac{1}{n}\sum_{i=1}^{n}\left\{\ell_i(\langle x_i, w_{\theta,i}\rangle) + \lambda f(w_{\theta,i}, \theta)\right\} - \mathcal{R}_Z(w) \leq \lambda f(w,\theta) + \frac{1}{2\lambda n}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\theta,i}\right\|^2_{\theta,*}, \tag{3.14}$$

where, in this case, $\theta \in \Theta$ *is such that* $f(w,\theta) < +\infty$*.*

The method we propose in this work relies on the application of Alg. 1 also to the meta-problem in Eq. (3.1) as the tasks are sequentially observed, using the functions $(\mathcal{L}_t)_{t=1}^T$ as meta-objectives. A key difficulty here is that the meta-objective is defined via the inner batch problem in Eq. (2.23), hence, it is not available exactly, but, as observed in Rem. 5, it is only approximately approached by the within-task online algorithm. From a practical point of view, this means that the application of Alg. 1 to the meta-problem in Eq. (3.1), differently from the inner algorithm, has to deal with an error on the meta-subgradients of the meta-objectives at each iteration. In order to describe in detail this aspect and the meta-algorithm, we need the following technical lemma.

Lemma 5 (Strong Duality for the Within-Task Problem)**.** *Let Asm. 1 hold. For any dataset* $Z = (x_i, y_i)_{i=1}^n$ *and any meta-parameter* $\theta \in \Theta$*, consider the primal and the dual within-task problems in Eq. (3.9) and Eq. (3.10), respectively. Then, the dual problem in Eq. (3.10) admits a solution*

$$\hat{s}_\theta \in \operatorname*{argmin}_{s\in\mathbb{R}^n} D_{n+1}(s,\theta). \tag{3.15}$$

Moreover, the following statements hold.

1. *Strong duality holds, namely, we have*

$$n\,\mathcal{L}_Z(\theta) = -\min_{s\in\mathbb{R}^n} D_{n+1}(s,\theta). \tag{3.16}$$

2. *The KKT conditions read as follows*

$$\hat{w}_\theta = \nabla f(\cdot,\theta)^*\Big(-\frac{1}{\lambda n}\sum_{i=1}^n x_i \hat{s}_{\theta,i}\Big) \qquad \hat{s}_\theta \in \partial\Big(\sum_{i=1}^n \ell_i\Big)\big(\langle x_1, \hat{w}_\theta\rangle, \dots, \langle x_n, \hat{w}_\theta\rangle\big), \quad (3.17)$$

where, we recall that $\hat{w}_\theta$ is the RERM in Eq. (2.24), coinciding with the solution of the primal problem in Eq. (3.9).

Proof. We rely on the standard result reported in Prop. 62 in App. A.1 according to which, the desired statements hold for the couples of within-task primal-dual problems above if, for any $\theta \in \Theta$, 1) the primal problem admits a solution and 2) there exist a point in $\mathrm{Dom} f(\cdot,\theta)$ where the function $\sum_{i=1}^n \ell_i(\langle x_i, \cdot\rangle)$ is continuous. Regarding the point 1), as already observed, the existence of the primal solution $\hat{w}_\theta$ is ensured by Asm. 1. Regarding the point 2), we observe that, thanks to Asm. 1, the function $\sum_{i=1}^n \ell_i(\langle x_i, \cdot\rangle)$ is real-valued. As a consequence, since a convex real-valued function is continuous over the entire space (see Lemma 40 in App. A), also the continuity requirement above is satisfied. Hence, the desired statement directly derives from specializing Prop. 62 in App. A to our context, observing that the strong convexity of $f(\cdot,\theta)$ is equivalent to the Lipschitz-smoothness of $f(\cdot,\theta)^*$ (see Lemma 57 in App. A). ■

Our next result describes how, leveraging on the dual optimality gap for the inner Alg. 2, we can compute an ϵ-subgradient of the meta-objective $\mathcal{L}_Z$ in Eq. (2.23), where ϵ is (up to normalization by n) the value stated in Prop. 3. This will allow us to develop an efficient method which is computationally appealing and fully online.

Proposition 6 (Computation of an ϵ-Subgradient of $\mathcal{L}_Z$)**.** *Let Asm. 1 hold and let $s_{\theta,n+1}$ be the output of Alg. 2 with $\theta \in \Theta$ over the dataset Z. Consider $\nabla_\theta \in \partial\{-D_{n+1}(s_{\theta,n+1},\cdot)\}(\theta)$, where D_{n+1} is the function in Eq. (3.10). Then, $\nabla'_\theta = \nabla_\theta/n \in \partial_{\epsilon_\theta/n}\mathcal{L}_Z(\theta)$, where ϵ_θ is the value in Prop. 3.*

Proof. We start from recalling that, for any $\theta \in \Theta$, the function $D_{n+1}(\cdot,\theta)$ reported in Eq. (3.10) is the objective of the dual problem associated to the non-normalized within-task problem in Eq. (3.9). Since in our assumptions, strong duality holds for this couple of problems (see Lemma 5 above), we can rewrite

$$\mathcal{L}_Z(\theta) = \max_{s\in\mathbb{R}^n} \tilde{D}_{n+1}(s,\theta) \qquad \tilde{D}_{n+1}(s,\theta) = -\frac{1}{n} D_{n+1}(s,\theta). \quad (3.18)$$

Thanks to Prop. 3, we know that the dual vector $s_{\theta,n+1}$ returned by Alg. 2 is an ϵ_θ-minimizer of the dual objective $D_{n+1}(\cdot,\theta)$, where ϵ_θ is given in Prop. 3. Consequently, $s_{\theta,n+1}$ is an (ϵ_θ/n)-maximizer of the function $\tilde{D}_{n+1}(\cdot,\theta)$ defined above. We now observe that, for any $\theta' \in \Theta$, we have

$$\begin{aligned}\mathcal{L}_Z(\theta') &= \max_{s\in\mathbb{R}^n} \tilde{D}_{n+1}(s,\theta') \geq \tilde{D}_{n+1}(s_{\theta,n+1},\theta') \\ &\geq \tilde{D}_{n+1}(s_{\theta,n+1},\theta) + \left\langle \frac{\nabla_\theta}{n}, \theta' - \theta \right\rangle \geq \mathcal{L}_Z(\theta) - \frac{\epsilon_\theta}{n} + \left\langle \frac{\nabla_\theta}{n}, \theta' - \theta \right\rangle,\end{aligned}$$

where, in the second inequality we have exploited the assumption $\nabla_\theta \in \partial\Big\{-D_{n+1}(s_{\theta,n+1},\cdot)\Big\}(\theta)$, implying $\nabla_\theta/n \in \partial\tilde{D}_{n+1}(s_{\theta,n+1},\cdot)(\theta)$, and in the last inequality we have used the fact that $s_{\theta,n+1}$ is an (ϵ_θ/n)-maximizer of the function $\tilde{D}_{n+1}(\cdot,\theta)$ as explained above and strong duality again. By definition of ϵ-subgradients, the above inequality proves the desired statement. ■

We remark that the procedure described above to compute an ϵ-subgradient has been also used in (Denevi et al., 2019a) for the statistical setting in Ex. 1. Here, with a different proof technique, we show that it can be extended also to more general inner regularizers.

Leveraging on the closed form of the error on the meta-subgradients computed as described in Prop. 6, we now show how we can deduce in a natural way an upper bound on the regularized variant of the first term in Eq. (2.12), without any additional assumptions. This automatically translates into a (regularized) average meta-regret bound for the OWO Meta-Learning procedure we are proposing.

Theorem 7 (Meta-Regret Bound)**.** *Let Asm. 1 and Asm. 2 hold. Then, Alg. 3 coincides with the generic primal-dual Alg. 1 applied to the meta-problem in Eq. (3.1). Fix now a meta-data sequence* $\mathbf{Z} = (Z_t)_{t=1}^T$*, where* $Z_t = (x_{t,i}, y_{t,i})_{i=1}^n$ *and denote by* $(\hat{w}_t)_{t=1}^T$ *the minimum norm empirical risk minimizers associated to the datasets* $(Z_t)_{t=1}^T$*. Denote by* $(\theta_t)_{t=1}^T$ *the iterates computed by Alg. 3 over* $\mathbf{Z}$*, by means of the approximated meta-subgradients* $(\nabla'_{\theta_t})_{t=1}^T$ *computed as described in Prop. 6. For any* $\theta \in \Theta$*, let* A_θ *be the corresponding Alg. 2 and let* $(w_{\theta_t,i})_{i=1}^n$ *be the iterates returned by the inner algorithm* A_{θ_t} *over the dataset* Z_t*, by means of the subgradients* $(s'_{\theta_t,i})_{i=1}^n$*, with* $s'_{\theta_t,i} \in \partial\ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i}\rangle)$*. Then, for any* $t \in \{1,\dots,T\}$*,* $\theta_t \in \Theta$ *and, for any* $\theta \in \Theta$ *such that* $f(\hat{w}_t,\theta) < +\infty$ *for any* $t \in \{1,\dots,T\}$*, the following (regularized) average meta-regret bound holds for the iterates above*

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^T \mathcal{E}_{\theta_t,Z_t}(A_{\theta_t}) \leq \frac{\lambda}{T}\sum_{t=1}^T f(\hat{w}_t,\theta) + \frac{1}{2\lambda nT}\sum_{t=1}^T\sum_{i=1}^n \frac{1}{i}\left\|x_{t,i}s'_{\theta_t,i}\right\|^2_{\theta_t,*} \\ + \frac{\eta F(\theta)}{T} + \frac{1}{2\eta T}\sum_{t=1}^T \left\|\nabla'_{\theta_t}\right\|^2_*.\end{aligned} \tag{3.19}$$

Proof. We first note that the outer-task problem in Eq. (3.1) is of the form in Eq. (3.2) with $m = M$, once we make the identifications

$$j \rightsquigarrow t, \;\; M \rightsquigarrow T, \;\; v \rightsquigarrow \theta, \;\; \mathcal{V} \rightsquigarrow \mathcal{M}, \;\; g_j \rightsquigarrow \mathcal{L}_t, \;\; A_j \rightsquigarrow I, \;\; c_m \rightsquigarrow \eta, \;\; r \rightsquigarrow F, \tag{3.20}$$

where I denotes the identity operator. Thus, exploiting these observations, it is immediate to see that the meta-algorithm in Alg. 3 coincides with Alg. 1 applied to the outer-tasks problem in Eq. (3.1), once one makes the identification $\alpha'_m \rightsquigarrow \nabla'_{\theta_t}$ for the (approximated) meta-subgradients used by the algorithm. As a consequence, $\theta_t \in \mathrm{Dom} F = \Theta$ for any $t \in \{1, \dots, T\}$. Moreover, denoting by Δ_{Dual} the associated dual optimality gap introduced in Thm. 2, specializing the first point of Thm. 2 to this setting and exploiting the non-negativity of Δ_{Dual}, we can write

$$0 \leq -\sum_{t=1}^{T} \mathcal{L}_t(\theta_t) + \min_{\theta \in \mathcal{M}} \left\{ \sum_{t=1}^{T} \mathcal{L}_t(\theta) + \eta F(\theta) \right\} + \frac{1}{2\eta} \sum_{t=1}^{T} \left\| \nabla'_{\theta_t} \right\|_*^2 + \sum_{t=1}^{T} \frac{\epsilon_{\theta_t}}{n}, \tag{3.21}$$

where, according to Prop. 3 (applied to the task t),

$$\frac{\epsilon_{\theta_t}}{n} = -\left(\frac{1}{n} \sum_{i=1}^{n} \left\{ \ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i} \rangle) + \lambda f(w_{\theta_t,i}, \theta_t) \right\} - \mathcal{L}_t(\theta_t) \right) + \frac{1}{2\lambda n} \sum_{i=1}^{n} \frac{1}{i} \left\| x_{t,i} s'_{\theta_t,i} \right\|_{\theta_t,*}^2. \tag{3.22}$$

Substituting this closed form into Eq. (3.21), one immediately observes that the term $\sum_{t=1}^{T} \mathcal{L}_t(\theta_t)$ erases, coming to

$$\begin{aligned} \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} \left\{ \ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i} \rangle) + \lambda f(w_{\theta_t,i}, \theta_t) \right\} \leq \min_{\theta \in \mathcal{M}} \left\{ \sum_{t=1}^{T} \mathcal{L}_t(\theta) + \eta F(\theta) \right\} + \frac{1}{2\eta} \sum_{t=1}^{T} \left\| \nabla'_{\theta_t} \right\|_*^2 \\ + \frac{1}{2\lambda n} \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1}{i} \left\| x_{t,i} s'_{\theta_t,i} \right\|_{\theta_t,*}^2. \end{aligned}$$

Now we observe that, by definition of $\mathcal{L}_t$ as minimum of the regularized empirical risk associated to the dataset Z_t, for any $\theta \in \Theta$ such that $f(\hat{w}_t, \theta) < +\infty$ for any $t \in \{1, \dots, T\}$, we can write

$$\sum_{t=1}^{T} \mathcal{L}_t(\theta) \leq \sum_{t=1}^{T} \mathcal{R}_{Z_t}(\hat{w}_t) + \lambda \sum_{t=1}^{T} f(\hat{w}_t, \theta). \tag{3.23}$$

Combining the two last inequalities above, moving the terms and normalizing by the number of tasks T, we get the desired statement. ∎

As we will see later, the term $\left\| \nabla'_{\theta_t} \right\|_*^2$ in Thm. 7 may hide a dependency w.r.t. λ or n and, in these cases, the resulting bound must be analyzed according to the specific setting at hand. On contrary, when the terms $\left\| \nabla'_{\theta_t} \right\|_*^2$ and $\left\| x_{t,i} s'_{\theta_t,i} \right\|_{\theta_t,*}^2$ can be upper bounded by a constant (not depending on

λ or n), using the inequality $\sum_{i=1}^{n} 1/i \leq \log(n) + 1$ and assuming the complexity terms f and F bounded, the bound above can be majorized as follows

$$\frac{1}{T}\sum_{t=1}^{T} \mathcal{E}_{\theta_t, Z_t}(A_{\theta_t}) \leq \mathcal{O}\left(\lambda + \frac{\log(n)+1}{\lambda n} + \frac{\eta}{T} + \frac{1}{\eta}\right). \tag{3.24}$$

Hence, for the choice of hyper-parameters $\lambda \approx n^{-1/2}$ and $\eta \approx T^{1/2}$, we recover a reasonable rate

$$\mathcal{O}\left(\sqrt{\frac{\log(n)+1}{n}}\right) + \mathcal{O}\left(\sqrt{\frac{1}{T}}\right). \tag{3.25}$$

Obviously, the above reasoning is rough, since, as we will see in the sequel, the constants in the bound above will play a key role in our analysis.

Similarly to what observed in Rem. 6, in the statistical setting described in the following Sec. 3.4, we will exploit the observation below.

Remark 7 (Weaker Meta-Regret Bound)**.** *Looking at Eq. (3.23) in the proof above, the reader can immediately see that the worst case meta-regret bound given in Thm. 7 can be immediately stated in the following weaker form for a generic sequence of competitor vectors* $(w_t)_{t=1}^T$*:*

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\left\{\frac{1}{n}\sum_{i=1}^{n}\left\{\ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i}\rangle) + \lambda f(w_{\theta_t,i}, \theta_t)\right\} - \mathcal{R}_{Z_t}(w_t)\right\} \\ \leq \frac{\lambda}{T}\sum_{t=1}^{T} f(w_t, \theta) + \frac{1}{2\lambda n T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|x_{t,i}s'_{\theta_t,i}\right\|^2_{\theta_t,*} \\ + \frac{\eta F(\theta)}{T} + \frac{1}{2\eta T}\sum_{t=1}^{T}\left|\!\left|\!\left|\nabla'_{\theta_t}\right|\!\right|\!\right|^2_*,\end{aligned} \tag{3.26}$$

where, in this case, $\theta \in \Theta$ *is such that* $f(w_t, \theta) < +\infty$ *for any* $t \in \{1, \ldots, T\}$*.*

In the next subsection, we introduce a way to measure the performance of our OWO Meta-Learning approach. More formally, we give an upper bound on the second term in Eq. (2.12).

3.3.1 The Benchmark for the Method

As described in Sec. 2.2, we would like to compare the performance of our method with the performance of the best algorithm in our class, solving the problem in Eq. (2.11):

$$\min_{\theta \in \Theta} \frac{1}{T} \sum_{t=1}^{T} \mathcal{E}_{Z_t}(A_\theta), \tag{2.11}$$

where A_θ denotes the inner Alg. 2 and $\mathcal{E}_{Z_t}(A_\theta)$ represents its average regret over the dataset Z_t. In the following result we give an upper bound for the regularized variant of the quantity above for a meta-parameter θ in a subset of Θ containing, as we will see in the following, the optimal meta-parameter $\hat{\theta}$. This bound automatically translates into a (regularized) average across-tasks regret bound deriving from the application of Alg. 2 with an appropriate meta-parameter θ fixed in hindsight for any tasks. Such a bound will be used as benchmark for the corresponding bound we have obtained in Thm. 7 for our Meta-Learning procedure.

Theorem 8 (Across-Tasks Regret Bound for Alg. 2)**.** *Let Asm. 1 hold. Fix a meta-data sequence* $\mathbf{Z} = (Z_t)_{t=1}^T$*, where* $Z_t = (x_{t,i}, y_{t,i})_{i=1}^n$ *and denote by* $(\hat{w}_t)_{t=1}^T$ *the minimum norm empirical risk minimizers associated to the datasets* $(Z_t)_{t=1}^T$*. For any* $\theta \in \Theta$*, let* A_θ *be the corresponding Alg. 2 and let* $(w_{t,i})_{i=1}^n$ *be the iterates returned by the inner algorithm* A_θ *over the dataset* Z_t*, by means of the subgradients* $(s'_{t,i})_{i=1}^n$*, with* $s'_{t,i} \in \partial \ell_{t,i}(\langle x_{t,i}, w_{t,i} \rangle)$*. Then, for any* $\theta \in \Theta$ *such that* $f(\hat{w}_t, \theta) < +\infty$ *for any* $t \in \{1, \dots, T\}$*, the following (regularized) average across-tasks regret bound holds for the above iterates*

$$\frac{1}{T} \sum_{t=1}^{T} \mathcal{E}_{\theta, Z_t}(A_\theta) \leq \frac{\lambda}{T} \sum_{t=1}^{T} f(\hat{w}_t, \theta) + \frac{1}{2\lambda n T} \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1}{i} \left\| x_{t,i} s'_{t,i} \right\|_{\theta,*}^2. \tag{3.27}$$

Proof. The statement directly derives from summing over the tasks the single-task regret bound in Eq. (3.12) and normalizing by the number of tasks T. ∎

We observe that the bound for our method in Eq. (3.19) is composed by two main parts: one part (the first row) is similar to the benchmark bound in Eq. (3.27), the other part (the second row) can be considered as the additional effort due to the estimation of the meta-parameter from the data. As we will see in the following Chpt. 5 and Chpt. 6 for the specific settings in Ex. 1 and Ex. 2, this last part of the bound can be made decreasing w.r.t. T, by choosing in an appropriate way the hyper-parameter η. However, since, as already observed, the bounds may hide delicate dependencies, at this point of the presentation, it is difficult to make a detailed and fair comparison between the bounds. We postpone this discussion later to more explicit settings.

We conclude this section with the following observation which will be used in the statistical setting below.

Remark 8 (Weaker Across-Tasks Regret Bound for Alg. 2)**.** *Summing over the tasks the weaker regret bound given in Rem. 6 with a generic sequence of competitor vectors $(w_t)_{t=1}^T$ and normalizing by the number of tasks T, we get the weaker form of the result in Thm. 8 above:*

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\left\{\frac{1}{n}\sum_{i=1}^{n}\left\{\ell_{t,i}(\langle x_{t,i}, w_{t,i}\rangle)+\lambda f(w_{t,i},\theta)\right\}-\mathcal{R}_{Z_t}(w_t)\right\}\\ \leq \frac{\lambda}{T}\sum_{t=1}^{T} f(w_t,\theta)+\frac{1}{2\lambda nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|x_{t,i}s'_{t,i}\right\|_{\theta,*}^2,\end{aligned} \tag{3.28}$$

where, in this case, $\theta \in \Theta$ is such that $f(w_t,\theta) < +\infty$ for any $t \in \{1,\dots,T\}$.

In the next section we provide theoretical guarantees for our OWO Meta-Learning method in the statistical setting.

3.4 Method and Analysis in the Statistical Setting

As described in Sec. 2.2.2, letting A_θ the inner Alg. 2, in the statistical setting, we measure the performance of our OWO Meta-Learning procedure by analyzing the gap in Eq. (2.15):

$$\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_\mu(A_{\bar\theta})-\min_{\theta\in\Theta}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_\mu(A_\theta), \tag{2.15}$$

where $\bar\theta$ is the average of the meta-parameters returned by the meta-learner in Alg. 3 and, for any $\theta \in \Theta$, A_θ is the corresponding inner Alg. 2. We recall that, for an online within-task algorithm A returning the sequence $(w_i)_{i=1}^n$ when applied to a dataset $Z \sim \mu^n$, $\mathcal{E}_\mu(A)$ is the expected excess risk of the average $\bar w$ of its iterations, which is defined in Eq. (2.6) as

$$\mathcal{E}_\mu(A)=\mathbb{E}_{Z\sim\mu^n}\,\mathcal{R}_\mu(\bar w)-\mathcal{R}_\mu(w_\mu). \tag{2.6}$$

In the subsequent analysis, we will investigate about the performance of our method, by comparing two upper bounds for the two terms above in Eq. (2.15). More precisely, also in this case, we will give two stronger upper bounds holding for the regularized versions of the above quantities, in which the expected excess risk $\mathcal{E}_\mu(A)$ of the inner algorithm A is substituted by the corresponding

regularized version $\mathcal{E}_{\theta,\mu}(A)$, defined as

$$\mathcal{E}_{\theta,\mu}(A) = \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta,\mu}(\bar{w}) - \mathcal{R}_\mu(w_\mu) \qquad \mathcal{R}_{\theta,\mu}(w) = \mathcal{R}_\mu(w) + \lambda f(w,\theta) \quad w \in \mathbb{R}^d. \tag{3.29}$$

Also in this case, by arguments similar to the ones made for the existence of the RERM in Eq. (2.24), exploiting Asm. 1, we manage to ensure the existence and the uniqueness of the regularized (true) risk minimizer above

$$w_{\theta,\mu} = \operatorname*{argmin}_{w\in\mathbb{R}^d} \mathcal{R}_{\theta,\mu}(w) \tag{3.30}$$

for any $\theta \in \Theta$. Before proceeding, we observe the following.

Remark 9 (Surrogate Function, Statistical Setting)**.** *Looking at the statistical target problem in Eq. (2.13):*

$$\min_{\theta\in\Theta} \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta), \tag{2.13}$$

we observe that for any task μ, the dependency on θ in the function $\mathcal{E}_\mu(A_\theta)$ is contained only in the first term

$$\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(\bar{w}_\theta) \tag{3.31}$$

coinciding with the expected (true) risk of the average $\bar{w}_\theta$ of the iterates $(w_{\theta,i})_{i=1}^n$ generated by the algorithm A_θ over the dataset Z. As observed in Rem. 5, also in this case, very often, such a quantity is non-convex w.r.t. the meta-parameter θ and for this reason it is not well-suited as meta-objective for a stochastic meta-algorithm. This issue is overcome, by considering the function $\mathbb{E}_{Z\sim\mu^n}\, \mathcal{L}_Z(\theta)$ as alternative meta-objective. This choice is reasonable in that, by similar arguments as those made in Rem. 9, such a function can represent a convex upper bound for the expected (true) risk of our inner algorithm. As a matter of fact, taking the expectation of Eq. (3.8) w.r.t. the sampling of $Z \sim \mu^n$ and combining with Eq. (2.9), we immediately get

$$\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(\bar{w}_\theta) \le \mathbb{E}_{Z\sim\mu^n}\, \mathcal{L}_Z(\theta) + \mathbb{E}_{Z\sim\mu^n}\, \frac{1}{2\lambda n} \sum_{i=1}^n \frac{1}{i} \left\| x_i s'_{\theta,i} \right\|_{\theta,*}^2. \tag{3.32}$$

In the following result we give an upper bound on the regularized variant of the first term in Eq. (2.15) which automatically translates into an (regularized) expected meta-excess risk bound for $\bar{w}_{\bar{\theta}}$, the average of the estimators generated from the combination of the inner Alg. 2 with the meta-algorithm in Alg. 3.

Theorem 9 (OWO Meta-Excess Risk Bound)**.** *Consider the statistical setting. Let Asm. 1 and Asm. 2 hold. Let $A_{\bar{\theta}}$ be the inner Alg. 2 with meta-parameter $\bar{\theta}$, the average of the meta-parameters returned by the meta-algorithm in Alg. 3 using the data $\mathbf{Z} = (Z_t)_{t=1}^T$ with $Z_t = (x_{t,i}, y_{t,i})_{i=1}^n$, by*

means of the approximated meta-subgradients $(\nabla'_{\theta_t})_{t=1}^T$*, computed as described above in Prop. 6. For a distribution* $\mu \sim \rho$*, fix a dataset* $Z = (x_i, y_i)_{i=1}^n \sim \mu^n$ *independent from* $\mathbf{Z}$*. Let* $\bar{w}_{\bar{\theta}}$ *be the average of the iterates* $(w_{\bar{\theta},i})_{i=1}^n$ *generated by applying* $A_{\bar{\theta}}$ *to the dataset* Z *by means of the subgradients* $(s'_{\bar{\theta},i})_{i=1}^n$*, with* $s'_{\bar{\theta},i} \in \partial \ell_i(\langle x_i, w_{\bar{\theta},i} \rangle)$*. Then, in expectation w.r.t. the sampling of the data* $\mathbf{Z}$*, recalling the minimum norm (true) risk minimizer* w_μ *associated to a distribution* $\mu \sim \rho$*, for any* $\theta \in \Theta$ *such that* $\mathbb{E}_{\mu\sim\rho}\, f(w_\mu, \theta) < +\infty$*, the following (regularized) expected meta-excess risk bound holds for* $\bar{w}_{\bar{\theta}}$

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_{\bar{\theta},\mu}(A_{\bar{\theta}}) \leq \lambda\, \mathbb{E}_{\mu\sim\rho}\, f(w_\mu, \theta) + \frac{1}{2\lambda n T}\, \mathbb{E}_{\mathbf{Z}} \sum_{t=1}^T \sum_{i=1}^n \frac{1}{i} \left\| x_{t,i} s'_{\theta_t,i} \right\|_{\theta_t,*}^2 \\ + \frac{\eta F(\theta)}{T} + \frac{1}{2\eta T}\, \mathbb{E}_{\mathbf{Z}} \sum_{t=1}^T \left\| \nabla'_{\theta_t} \right\|_*^2 \\ + \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \frac{1}{2\lambda n} \sum_{i=1}^n \frac{1}{i} \left\| x_i s'_{\bar{\theta},i} \right\|_{\bar{\theta},*}^2 .\end{aligned} \tag{3.33}$$

The proof of the statement above exploits the (regularized) average meta-regret bound for our OWO Meta-Learning procedure and two nested online-to-batch conversion steps, see e.g. (Cesa-Bianchi and Gentile, 2006; Cesa-Bianchi et al., 2004; Littlestone, 1989), one within-task and one across-tasks. Specifically, looking at the proof below, the reader can note that the first two rows of the bound above coincide with the expectation of the (regularized) average meta-regret bound for our method, while the additional term in the last row derives from the online-to-batch conversions. As we will see in the following, this further term can be treated in a similar way as the last term in the first row and, as a consequence, it will not affect the overall behavior of the final bound. We now provide the online-to-batch conversions necessary for the proof of Thm. 9.

Proposition 10 (Online-To-Batch Meta-Conversion)**.** *Under the same assumptions of Thm. 9, in expectation w.r.t. the sampling of the data* $\mathbf{Z} = (Z_t)_{t=1}^T$*, it holds*

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_{\bar{\theta},\mu}(A_{\bar{\theta}}) \leq \mathbb{E}_{\mathbf{Z}}\, \frac{1}{T} \sum_{t=1}^T \mathcal{E}_{\theta_t, Z_t}(A_{\theta_t}) + \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \frac{1}{2\lambda n} \sum_{i=1}^n \frac{1}{i} \left\| x_i s'_{\bar{\theta},i} \right\|_{\bar{\theta},*}^2 .$$

Proof. Throughout this proof, for any $\theta \in \Theta$, we will need to make explicit the dependency w.r.t. the dataset in the RERM $\hat{w}_\theta$ in Eq. (2.24), in the iterations $(w_{\theta,i})_{i=1}^n$ generated by Alg. 2 and in their average $\bar{w}_\theta$. We will also use the short-hand notation

$$C = \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \frac{1}{2\lambda n} \sum_{i=1}^n \frac{1}{i} \left\| x_i s'_{\bar{\theta},i} \right\|_{\bar{\theta},*}^2 . \tag{3.34}$$

Recalling that $\hat{w}_t$ denotes the minimum norm minimizer of the empirical risk associated to the dataset Z_t, the desired statement can be written more explicitly as follows

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\bar{\theta},\mu}(\bar{w}_{\bar{\theta}}(Z)) - \mathbb{E}_{\mu\sim\rho}\, \mathcal{R}_\mu(w_\mu) \leq \\ \mathbb{E}_{\mathbf{Z}}\, \frac{1}{T}\sum_{t=1}^{T} \frac{1}{n}\sum_{i=1}^{n} \left\{ \ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i}(Z_t)\rangle) + \lambda f(w_{\theta_t,i}(Z_t), \theta_t) \right\} - \mathbb{E}_{\mathbf{Z}}\, \frac{1}{T}\, \sum_{t=1}^{T} \mathcal{R}_{Z_t}(\hat{w}_t) + C. \tag{3.35}$$

In order to prove the statement above, we will prove the following two partial results.

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\bar{\theta},\mu}(\bar{w}_{\bar{\theta}}(Z)) \leq \mathbb{E}_{\mathbf{Z}}\, \frac{1}{T}\sum_{t=1}^{T} \frac{1}{n}\sum_{i=1}^{n} \left\{ \ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i}(Z_t)\rangle) + \lambda f(w_{\theta_t,i}(Z_t), \theta_t) \right\} + C \tag{3.36}$$

and

$$\mathbb{E}_{\mathbf{Z}}\, \frac{1}{T}\, \sum_{t=1}^{T} \mathcal{R}_{Z_t}(\hat{w}_t) \leq \mathbb{E}_{\mu\sim\rho}\, \mathcal{R}_\mu(w_\mu). \tag{3.37}$$

In the following, we will explicitly write the expectation $\mathbb{E}_{\mathbf{Z}}$ in the statements above as

$$\mathbb{E}_{\mu_1,\ldots,\mu_T\sim\rho^T}\, \mathbb{E}_{Z_1\sim\mu_1^n,\ldots,Z_T\sim\mu_T^n}. \tag{3.38}$$

We now prove Eq. (3.36). We start from observing that, for any dataset $Z \sim \mu^n$ and for any $\theta \in \Theta$ not depending on Z, recalling the subgradients $(s'_{\theta,i})_{i=1}^n$, $s'_{\theta,i} \in \partial\ell_i(\langle x_i, w_{\theta,i}(Z)\rangle)$, used by Alg. 2 over Z, we can write

$$\begin{aligned} \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta,\mu}(\bar{w}_\theta(Z)) &\leq \mathbb{E}_{Z\sim\mu^n}\, \frac{1}{n}\sum_{i=1}^{n} \mathcal{R}_{\theta,\mu}(w_{\theta,i}(Z)) \\ &= \mathbb{E}_{Z\sim\mu^n}\, \frac{1}{n}\sum_{i=1}^{n} \left\{ \ell_i(\langle x_i, w_{\theta,i}(Z)\rangle) + \lambda f(w_{\theta,i}(Z), \theta) \right\} \\ &\leq \mathbb{E}_{Z\sim\mu^n}\, \left[\mathcal{L}_Z(\theta) + \frac{1}{2\lambda n}\, \sum_{i=1}^{n} \frac{1}{i} \left\| x_i s'_{\theta,i} \right\|_{\theta,*}^2 \right]. \end{aligned} \tag{3.39}$$

In the first inequality above we have applied Jensen's inequality (see Lemma 39 in App. A) to the convex function $\mathcal{R}_{\theta,\mu}$, the equality holds by standard online-to-batch arguments, more precisely, since $w_{\theta,i}(Z)$ depends only on the points $(z_j)_{j=1}^{i-1}$, thanks to the fact $Z \sim \mu^n$, we have

$$\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta,\mu}(w_{\theta,i}(Z)) = \mathbb{E}_{Z\sim\mu^n}\, \left[\ell_i(\langle x_i, w_{\theta,i}(Z)\rangle) + \lambda f(w_{\theta,i}(Z), \theta) \right], \tag{3.40}$$

and, finally, the last inequality derives from Eq. (3.8). Hence, rewriting $\mathcal{L}_Z(\theta) = \mathcal{R}_{\theta,Z}(\hat{w}_\theta(Z))$ for any $\theta \in \Theta$, we can write the following

$$\begin{aligned}
&\mathbb{E}_{\mu_1,\dots,\mu_T\sim\rho^T}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_T\sim\mu_T^n}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\bar{\theta},\mu}(\bar{w}_{\bar{\theta}}(Z)) \\
&\qquad \leq \mathbb{E}_{\mu_1,\dots,\mu_T\sim\rho^T}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_T\sim\mu_T^n}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\bar{\theta},Z}(\hat{w}_{\bar{\theta}}(Z)) + C \\
&\qquad \leq \mathbb{E}_{\mu_1,\dots,\mu_T\sim\rho^T}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_T\sim\mu_T^n}\, \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mu\sim\rho}\, \underbrace{\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta_t,Z}(\hat{w}_{\theta_t}(Z))} + C,
\end{aligned} \tag{3.41}$$

where, in the first inequality we have applied Eq. (3.39) with $\theta = \bar{\theta}$ and in the second inequality we have applied Jensen's inequality (see Lemma 39 in App. A) to the convex function $\mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{L}_Z$. We now observe that, by definition of $\hat{w}_{\theta_t}(Z)$ and $w_{\theta_t,\mu}$, we can write the following

$$\begin{aligned}
\underbrace{\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta_t,Z}(\hat{w}_{\theta_t}(Z))} &\leq \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta_t,Z}(w_{\theta_t,\mu}) = \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta_t,\mu}(w_{\theta_t,\mu}) \\
&\leq \underbrace{\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta_t,\mu}(\bar{w}_{\theta_t}(Z))}\,.
\end{aligned} \tag{3.42}$$

Substituting in Eq. (3.41), we can write the following

$$\begin{aligned}
&\mathbb{E}_{\mu_1,\dots,\mu_T\sim\rho^T}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_T\sim\mu_T^n}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\bar{\theta},\mu}(\bar{w}_{\bar{\theta}}(Z)) \\
&\leq \mathbb{E}_{\mu_1,\dots,\mu_T\sim\rho^T}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_T\sim\mu_T^n}\, \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mu\sim\rho}\, \underbrace{\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta_t,Z}(\hat{w}_{\theta_t}(Z))} + C \\
&\leq \mathbb{E}_{\mu_1,\dots,\mu_T\sim\rho^T}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_T\sim\mu_T^n}\, \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mu\sim\rho}\, \underbrace{\mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta_t,\mu}(\bar{w}_{\theta_t}(Z))} + C \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mu_1,\dots,\mu_{t-1}\sim\rho^{t-1}}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_{t-1}\sim\mu_{t-1}^n}\, \mathbb{E}_{\mu_t\sim\rho}\, \mathbb{E}_{Z_t\sim\mu_t^n}\, \mathcal{R}_{\theta_t,\mu_t}(\bar{w}_{\theta_t}(Z_t)) + C \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mu_1,\dots,\mu_{t-1}\sim\rho^{t-1}}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_{t-1}\sim\mu_{t-1}^n}\, \mathbb{E}_{\mu_t\sim\rho}\, \mathbb{E}_{Z_t\sim\mu_t^n}\, \frac{1}{n}\sum_{i=1}^{n} \mathcal{R}_{\theta_t,\mu_t}(w_{\theta_t,i}(Z_t)) + C \\
&= \mathbb{E}_{\mu_1,\dots,\mu_T\sim\rho^T}\, \mathbb{E}_{Z_1\sim\mu_1^n,\dots,Z_T\sim\mu_T^n}\, \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n} \left\{ \ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i}(Z_t)\rangle) + \lambda f(w_{\theta_t,i}(Z_t),\theta_t) \right\} + C
\end{aligned}$$

where, in the first equality we have exploited the fact that θ_t depends only on the datasets $(Z_j)_{j=1}^{t-1}$ and the i.i.d. sampling of the datasets, in the third inequality we have applied Jensen's inequality (see Lemma 39 in App. A) to the convex function $\mathcal{R}_{\theta_t,\mu_t}$ and, finally, in the second equality we have exploited the fact that $w_{\theta_t,i}(Z_t)$ depends only on the points $(z_{t,j})_{j=1}^{i-1}$ and, consequently,

thanks to the fact $Z_t \sim \mu_t^n$, as already observed in Eq. (3.40),

$$\mathbb{E}_{Z_t \sim \mu_t^n} \mathcal{R}_{\theta_t, \mu_t}(w_{\theta_t,i}(Z_t)) = \mathbb{E}_{Z_t \sim \mu_t^n} \Big[\ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i}(Z_t) \rangle) + \lambda f(w_{\theta_t,i}(Z_t), \theta_t) \Big]. \tag{3.43}$$

This provides the first necessary statement in Eq. (3.36). The second statement in Eq. (3.37) is a direct consequence of the following steps

$$\begin{aligned} \mathbb{E}_{\mu_1, \dots, \mu_T \sim \rho^T} \mathbb{E}_{Z_1 \sim \mu_1^n, \dots, Z_T \sim \mu_T^n} \frac{1}{T} \sum_{t=1}^T \mathcal{R}_{Z_t}(\hat{w}_t) &\leq \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{\mu_t \sim \rho} \mathbb{E}_{Z_t \sim \mu_t^n} \mathcal{R}_{Z_t}(w_{\mu_t}) \\ &= \mathbb{E}_{\mu \sim \rho} \mathbb{E}_{Z \sim \mu^n} \mathcal{R}_Z(w_\mu) \\ &= \mathbb{E}_{\mu \sim \rho} \mathcal{R}_\mu(w_\mu), \end{aligned} \tag{3.44}$$

where in the inequality we have exploited the definition of the vectors $\hat{w}_t$, in the first equality we have leveraged the i.i.d. sampling of the datasets and in the second equality we have used the fact that $Z \sim \mu$ and the independence of w_μ on the data Z. Combining Eq. (3.36) and Eq. (3.37) we get Eq. (3.35) which is the desired statement. ■

The above result in Prop. 10 is a different version of (Alquier et al., 2017, Thm. 6.1) and (Balcan et al., 2019, Thm. 3.3), where the authors give statistical guarantees for the meta-parameter defined by sampling uniformly from the whole pool of the meta-parameters $(\theta_t)_{t=1}^T$ returned by their method. Their result is consequently in expectation w.r.t. the data and w.r.t. this uniform sampling. On the contrary, in our case, leveraging on the convexity of our surrogate functions and the fact that we derived a *regularized* average cumulative error bound for the inner algorithm (see Prop. 3), we have been able to obtain statistical guarantees for the average of the meta-parameters, without adding randomness and without the need of memorizing the previous meta-parameters.

Similarly to what already observed in Rem. 2 for the Single-Task Learning setting, in order to prove Thm. 9, we will use the following observation.

Remark 10 (Online-To-Batch Meta-Conversion by Weaker Meta-Regret)**.** *Looking above at the second row in Eq. (3.44), the reader can note that, in order to have a meta-excess risk bound, one does not necessarily use a worst case meta-regret bound as the one in Thm. 7, but it is sufficient to take the expectation of a weaker meta-regret bound as the one in Rem. 7 w.r.t. the sequence of the competitor vectors* $(w_{\mu_t})_{t=1}^T$.

We now have all the ingredients necessary to prove Thm. 9.

Proof of Thm. 9. The statement derives from applying Prop. 10 as described in Rem. 10 and observing that, when we take the expectation w.r.t. the i.i.d. sampling of the data $\mathbf{Z}$ in the weaker (regularized) average meta-regret bound in Rem. 7, we have

$$\mathbb{E}_{\mu_1,\dots,\mu_T\sim\rho^T} \frac{1}{T} \sum_{t=1}^{T} f(w_{\mu_t}, \theta) = \mathbb{E}_{\mu\sim\rho} f(w_\mu, \theta). \tag{3.45}$$

■

In the next subsection, we introduce a way to measure the performance of our OWO Meta-Learning approach in the statistical setting. More formally, we give an upper bound on the regularized variant of the second term in Eq. (2.15).

3.4.1 The Benchmark for the Method

As described in Sec. 2.2, we would like to compare the performance of our method with the performance of the best algorithm in our class, solving the problem in Eq. (2.13):

$$\min_{\theta\in\Theta} \mathbb{E}_{\mu\sim\rho} \mathcal{E}_\mu(A_\theta) \tag{2.13}$$

where A_θ denotes the inner Alg. 2 and $\mathcal{E}_\mu(A_\theta)$ represents the expected excess risk of the average of its iterations over the task $\mu \sim \rho$. In the following result we give an upper bound for the regularized variant of the quantity above for a meta-parameter θ in a subset of Θ containing, as we will see in the following, the optimal meta-parameter θ_ρ. This bound automatically translates into an (regularized) expected across-tasks excess risk bound for $\bar{w}_\theta$, the average of the iterations generated by Alg. 2 with an appropriate meta-parameter θ fixed in hindsight for any task. Such a bound will be used as benchmark for the corresponding bound we have obtained in Thm. 9 for the meta-parameter estimated by our Meta-Learning procedure.

Theorem 11 (Across-Tasks Excess Risk Bound for Alg. 2)**.** *Consider the statistical setting and let Asm. 1 hold. For a distribution $\mu \sim \rho$, fix a dataset $Z = (x_i, y_i)_{i=1}^n \sim \mu^n$. For any $\theta \in \Theta$, let A_θ be the corresponding inner Alg. 2 and let $(w_{\theta,i})_{i=1}^n$ be the iterates returned by A_θ over the dataset Z, by means of the subgradients $(s'_{\theta,i})_{i=1}^n$, with $s'_{\theta,i} \in \partial\ell_i(\langle x_i, w_{\theta,i}\rangle)$. Then, recalling the minimum norm (true) risk minimizer w_μ associated to a distribution $\mu \sim \rho$, for any $\theta \in \Theta$ such that $\mathbb{E}_{\mu\sim\rho} f(w_\mu, \theta) < +\infty$, the following (regularized) expected across-tasks excess risk bound*

holds for $\bar{w}_\theta$

$$\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_{\theta,\mu}(A_\theta) \leq \lambda\,\mathbb{E}_{\mu\sim\rho}\,f(w_\mu,\theta) + \frac{1}{2\lambda n}\,\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\,\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\theta,i}\right\|^2_{\theta,*}. \tag{3.46}$$

Proof. We start from observing that, applying the weaker regret bound in Rem. 6 with the competitor vector $w = w_\mu$, we can write

$$\frac{1}{n}\sum_{i=1}^{n}\left\{\ell_i(\langle x_i, w_{\theta,i}\rangle) + \lambda f(w_{\theta,i},\theta)\right\} \leq \mathcal{R}_Z(w_\mu) + \lambda f(w_\mu,\theta) + \frac{1}{2\lambda n}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\theta,i}\right\|^2_{\theta,*}.$$

Taking the expectation of the above bound w.r.t. $\mu\sim\rho$ and $Z\sim\mu^n$, recalling that, as already observed in Eq. (3.39),

$$\mathbb{E}_{Z\sim\mu^n}\,\mathcal{R}_{\theta,\mu}(\bar{w}_\theta) \leq \mathbb{E}_{Z\sim\mu^n}\,\frac{1}{n}\sum_{i=1}^{n}\left\{\ell_i(\langle x_i, w_{\theta,i}\rangle) + \lambda f(w_{\theta,i},\theta)\right\} \tag{3.47}$$

and recalling that $\mathbb{E}_{Z\sim\mu^n}\,\mathcal{R}_Z(w_\mu) = \mathcal{R}_\mu(w_\mu)$, we obtain the desired statement. ■

Looking at the bound in Eq. (3.33) for our method and the benchmark performance in Eq. (3.46), the conclusions and the comments we can derive are similar to the ones we have given above for the performance of our method in the non-statistical setting.

In the next section, we describe how our OWO Meta-Learning method fits in the recent literature.

3.5 Related Work

One of the first OWO Meta-Learning framework has been presented in (Alquier et al., 2017). In that case, the proposed setting can cover a quite broad family of inner algorithms and, as observed before, it can be adapted by online-to-batch arguments to the statistical framework. However, the main drawback of that work is the fact that the proposed meta-algorithm is not efficient, since it requires memorizing the entire data sequence.

In the more recent work (Finn et al., 2019), the authors consider under the Meta-Learning perspective the problem of the so-called *fine tuning*, in which the goal is to estimate a good starting point for a prescribed iterative inner algorithm. Specifically, they consider as inner algorithm one step of gradient descent from the point θ, namely, for an appropriate step size

$\gamma > 0$, $w_\theta = \theta - \gamma \nabla \hat{f}(\theta)$, where $\hat{f}$ is some function, for instance an approximation of the (true) risk. Then, in order to estimate the initial point θ, they consider a meta-objective of the form $\mathcal{L}(\theta) = f(\theta - \gamma \nabla \hat{f}(\theta))$, where f is another function with the same intuition of $\hat{f}$. The main result in (Finn et al., 2019) is to show that, under strong assumptions on the functions f and $\hat{f}$, such meta-objective is (strongly) convex in the meta-parameter θ. Once proven this, they propose to estimate the starting point applying as meta-algorithm Follow-The-Leader on the sequence of these functions and, relying on the well-known analysis for this algorithm, they state a regret bound for it.

Perhaps closer in spirit to our work is (Balcan et al., 2019). In that work, the authors consider as inner algorithm linearized Follow The Regularized Leader with constant step size and a penalty term given by a Bregman divergence parametrized by a meta-parameter. On the contrary, our inner algorithm corresponds to fixing the step size as $1/(\lambda(i+1))$ at each iteration and this allows us to derive a *regularized* regret bound. This, as already observed, brings benefits in the statistical setting. Furthermore, the proposed meta-algorithm here is different from the one in (Balcan et al., 2019), in that it works on different objective functions. In their case, as meta-objectives, they consider the sequence of Bregman divergences evaluated at the empirical risk minimizer of the corresponding task, while in our case, we consider the minimum of the entire regularized empirical risk. Such a choice, combined with the primal-dual interpretation of Follow The Regularized Leader and the concept of approximated subgradients, allows us to develop an OWO method without needing to add further assumptions. On the other hand, in (Balcan et al., 2019), in order to extend their work to the fully online setting, the authors need additional assumptions (specifically a growth condition on the empirical error). We also mention the very recent paper (Khodak et al., 2019), which is a sequel of the work mentioned above. In (Khodak et al., 2019), the authors consider a setting similar to the one described in (Balcan et al., 2019) and they propose a Meta-Learning approach to estimate also the step size of the inner algorithm. However, also in this case, the basic version of their method requires to compute a batch within-task empirical risk minimizer and, in order to extend their framework to the fully online setting, they need to introduce additional assumptions on the loss functions.

At last, we briefly discuss the framework analyzed in the other paper (Denevi et al., 2019a) published during the PhD. As we will see in Chpt. 5, the method and the analysis proposed there can be recovered from the OWO framework described above for the specific case of Ex. 1 in the statistical setting. In the framework described in the previous section, we develop a different analysis which allows us to extend the study to more general family of learning algorithms, also in the non-statistical setting.

3.6 Discussion

In this chapter we have presented our general OWO Meta-Learning method. We have analyzed its performance in the non-statistical setting and we have shown how these guarantees can be adapted to the statistical setting by online-to-batch arguments. The results in this chapter are not jet ready to be interpreted, since they could hide some delicate dependencies, but, as we will see in the following Chpt. 5 and Chpt. 6, when specified to the settings in Ex. 1 and Ex. 2, they will return interesting results, showing that the general analysis performed in this chapter is meaningful.

Chapter 4

An Online-Within-Batch Variant of the Method in the Statistical Setting

The method we have proposed in the previous Chpt. 3 is an Online-Within-Online method, i.e. it processes the data sequentially both within and across the tasks. Looking at many work in literature dealing with the statistical Online-Within-Batch (OWB) Meta-Learning framework – see e.g. (Balcan et al., 2019; Bullins et al., 2019; Denevi et al., 2018a,b; Khodak et al., 2019; Maurer et al., 2013, 2016) – one natural question arising is how much we are loosing in considering inner algorithms processing the data sequentially instead of in one batch. In this chapter, we investigate this aspect. Specifically, in Sec. 4.1, we introduce the natural OWB variant of our method and we study its guarantees in the statistical setting. We then discuss the most related work to this framework in Sec. 4.2 and, finally, we summarize the content of this chapter in Sec. 4.3.

4.1 Method and Analysis in the Statistical Setting

In this section, we introduce the OWB variant of our OWO method, where we substitute the family of inner algorithms in Alg. 2, given by (linearized) Follow The Regularized Leader applied to the within-task problem in Eq. (2.23), with the family of the batch minimizers of the same problem. Namely, for any meta-parameter $\theta \in \Theta$ and any dataset Z, we consider the batch inner algorithm given by the regularized empirical risk minimizer (RERM) introduced in Eq. (2.24). More precisely, we assume to be able to compute an exact solution $\hat{s}_\theta$ of the corresponding dual problem in Eq. (3.15). We recall that the existence of such a dual solution is guaranteed by

Lemma 5 and such a dual solution uniquely determines the corresponding (primal) RERM by the KKT conditions in Eq. (3.17).

In order to select a good algorithm in this new RERM family, i.e. in order to estimate a good meta-parameter from the data, we apply again the meta-algorithm outlined in Alg. 3, working this time with exact meta-subgradients. More precisely, the meta-subgradients are computed as described in the following proposition which differs from Prop. 6 only in the dual variable used. In Prop. 6 the dual variable is the last dual iteration of Alg. 2, corresponding to an *approximated* solution of the dual problem in Eq. (3.10), while, in the following proposition the dual variable is an *exact* solution of the dual problem in Eq. (3.10).

Proposition 12 (Computation of a Subgradient of $\mathcal{L}_Z$)**.** *Let Asm. 1 hold and let $\hat{s}_\theta$ be a solution of the within-task dual problem in Eq. (3.10) with $\theta \in \Theta$ on the dataset Z. Consider $\hat{\nabla}_\theta \in \partial\{-D_{n+1}(\hat{s}_\theta, \cdot)\}(\theta)$, where D_{n+1} is the function in Eq. (3.10). Then, $\hat{\nabla}'_\theta = \hat{\nabla}_\theta / n \in \partial \mathcal{L}_Z(\theta)$.*

Proof. The proof proceeds as in Prop. 6. We recall that, the function $D_{n+1}(\cdot, \theta)$ in Eq. (3.10) is the objective of the dual problem associated to the non-normalized within-task problem in Eq. (3.9) and, by strong duality (see Lemma 5), for any $\theta \in \Theta$,

$$\mathcal{L}_Z(\theta) = \max_{s \in \mathbb{R}^n} \tilde{D}_{n+1}(s, \theta) \qquad \tilde{D}_{n+1}(s, \theta) = -\frac{1}{n} D_{n+1}(s, \theta). \tag{4.1}$$

We also stress that, by definition, $\hat{s}_\theta$ is a maximizer of the function $\tilde{D}_{n+1}$. We now observe that, for any $\theta' \in \Theta$, we can write

$$\begin{aligned}\mathcal{L}_Z(\theta') &= \max_{s \in \mathbb{R}^n} \tilde{D}_{n+1}(s, \theta') \geq \tilde{D}_{n+1}(\hat{s}_\theta, \theta') \\ &\geq \tilde{D}_{n+1}(\hat{s}_\theta, \theta) + \left\langle \frac{\hat{\nabla}_\theta}{n}, \theta' - \theta \right\rangle = \mathcal{L}_Z(\theta) + \left\langle \frac{\hat{\nabla}_\theta}{n}, \theta' - \theta \right\rangle,\end{aligned}$$

where, in the first equality we have exploited strong duality, in the second inequality we have exploited the assumption $\hat{\nabla}_\theta \in \partial\{-D_{n+1}(\hat{s}_\theta, \cdot)\}(\theta)$, implying that $\hat{\nabla}_\theta / n \in \partial \tilde{D}_{n+1}(\hat{s}_\theta, \cdot)(\theta)$, and in the last equality we have used the fact that $\hat{s}_\theta$ is by definition a maximizer of the function $\tilde{D}_{n+1}(\cdot, \theta)$ defined above and strong duality again. By definition of subgradients, the above inequality proves the desired statement. ■

In the rest of this section, we study the guarantees for the variant of the method described above in the statistical setting. Similarly to what described in Sec. 3.4 for our OWO Meta-Learning method, also in this OWB variant, we measure the performance of our procedure by analyzing

the gap

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_{\bar\theta}) - \min_{\theta\in\Theta}\, \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta), \tag{2.15}$$

where, this time, $\bar\theta$ is the average of the meta-parameters returned by the meta-learner in Alg. 3 working with exact meta-subgradients computed as described in Prop. 12 and, for any $\theta \in \Theta$, A_θ is the corresponding inner RERM in Eq. (2.24) and $\mathcal{E}_\mu(A_\theta)$ is its expected excess risk over the task $\mu \sim \rho$:

$$\mathcal{E}_\mu(A_\theta) = \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_\mu(\hat w_\theta) - \mathcal{R}_\mu(w_\mu). \tag{4.2}$$

Differently to what we have done before, in the subsequent analysis, we will give two upper bounds for the two terms above in Eq. (2.15), without additional regularization terms. These upper bounds will be then compared to the ones we have obtained in Chpt. 3 for the OWO variant of our method.

We start from giving an upper bound for the first term in Eq. (2.15). In order to do this, we first need to study the generalization error of the batch RERM algorithm in Eq. (2.24), i.e. the discrepancy between the (true) risk and the empirical risk of the corresponding estimator. This is done in the following result where we exploit stability arguments, more precisely the so-called hypothesis stability, see (Bousquet and Elisseeff, 2002, Def. 3).

Proposition 13 (Generalization Error of the RERM Algorithm in Eq. (2.24))**.** *Consider the statistical setting and let Asm. 1 hold. For a distribution $\mu \sim \rho$, fix a dataset $Z = (x_i, y_i)_{i=1}^n \sim \mu^n$ and, for any $i \in \{1, \dots, n\}$, fix a datapoint $z_i' = (x_i', y_i') \sim \mu$ independent from Z. For any $\theta \in \Theta$, let $\hat w_\theta(Z)$ be the corresponding RERM in Eq. (2.24) over Z and let $s'_{\theta,i} \in \partial\ell(\cdot, y_i')(\langle x_i', \hat w_\theta(Z)\rangle)$. Then, the following generalization error bound holds for $\hat w_\theta(Z)$*

$$\mathbb{E}_{Z\sim\mu^n}\left[\mathcal{R}_\mu(\hat w_\theta(Z)) - \mathcal{R}_Z(\hat w_\theta(Z))\right] \le \frac{2}{\lambda n}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z_i'\sim\mu} \left\| x_i' s'_{\theta,i} \right\|_{\theta,*}^2. \tag{4.3}$$

Proof. During this proof, we need to make explicit the dependency of the RERM $\hat w_\theta$ in Eq. (2.24) w.r.t. the dataset Z. For any $i \in \{1, \dots, n\}$, consider the dataset $Z^{(i)}$, a copy of the original dataset Z in which we exchange the point $z_i = (x_i, y_i)$ with the new i.i.d. point $z_i' = (x_i', y_i')$. For a fixed $\theta \in \Theta$, we analyze how much this perturbation affects the outputs of the RERM algorithm in Eq. (2.24). In other words, we study the discrepancy between $\hat w_\theta(Z)$ and $\hat w_\theta(Z^{(i)})$. We start from observing that, since by Asm. 1 $\mathcal{R}_{\theta,Z}$ is λ-strongly convex w.r.t. $\|\cdot\|_\theta$, by growth condition

(see Lemma 56 in App. A) and the definition of the RERM algorithm, we can write the following

$$\begin{aligned}\frac{\lambda}{2}\left\|\hat{w}_\theta(Z^{(i)})-\hat{w}_\theta(Z)\right\|_\theta^2 &\leq \mathcal{R}_{\theta,Z}(\hat{w}_\theta(Z^{(i)}))-\mathcal{R}_{\theta,Z}(\hat{w}_\theta(Z))\\ \frac{\lambda}{2}\left\|\hat{w}_\theta(Z^{(i)})-\hat{w}_\theta(Z)\right\|_\theta^2 &\leq \mathcal{R}_{\theta,Z^{(i)}}(\hat{w}_\theta(Z))-\mathcal{R}_{\theta,Z^{(i)}}(\hat{w}_\theta(Z^{(i)})).\end{aligned} \tag{4.4}$$

Hence, summing the two inequalities above, we get

$$\begin{aligned}&\lambda\left\|\hat{w}_\theta(Z^{(i)})-\hat{w}_\theta(Z)\right\|_\theta^2\\ &\quad\leq \mathcal{R}_{\theta,Z}(\hat{w}_\theta(Z^{(i)}))-\mathcal{R}_{\theta,Z^{(i)}}(\hat{w}_\theta(Z^{(i)}))+\mathcal{R}_{\theta,Z^{(i)}}(\hat{w}_\theta(Z))-\mathcal{R}_{\theta,Z}(\hat{w}_\theta(Z))\\ &\quad=\frac{\mathrm{A}+\mathrm{B}}{n},\end{aligned} \tag{4.5}$$

where we have introduced the terms

$$\begin{aligned}\mathrm{A}&=\ell(\langle x_i',\hat{w}_\theta(Z)\rangle,y_i')-\ell(\langle x_i',\hat{w}_\theta(Z^{(i)})\rangle,y_i')\\ \mathrm{B}&=\ell(\langle x_i,\hat{w}_\theta(Z^{(i)})\rangle,y_i)-\ell(\langle x_i,\hat{w}_\theta(Z)\rangle,y_i).\end{aligned} \tag{4.6}$$

Now, exploiting the assumption $s_{\theta,i}'\in\partial\ell(\cdot,y_i')(\langle x_i',\hat{w}_\theta(Z)\rangle)$, applying Holder's inequality (see Lemma 32 in App. A) and introducing a subgradient $s_{\theta,i}\in\partial\ell(\cdot,y_i)(\langle x_i,\hat{w}_\theta(Z^{(i)})\rangle)$, we can write

$$\begin{aligned}\mathrm{A}&\leq\left\langle x_i's_{\theta,i}',\hat{w}_\theta(Z)-\hat{w}_\theta(Z^{(i)})\right\rangle\leq\left\|x_i's_{\theta,i}'\right\|_{\theta,*}\left\|\hat{w}_\theta(Z^{(i)})-\hat{w}_\theta(Z)\right\|_\theta\\ \mathrm{B}&\leq\left\langle x_is_{\theta,i},\hat{w}_\theta(Z^{(i)})-\hat{w}_\theta(Z)\right\rangle\leq\left\|x_is_{\theta,i}\right\|_{\theta,*}\left\|\hat{w}_\theta(Z^{(i)})-\hat{w}_\theta(Z)\right\|_\theta.\end{aligned} \tag{4.7}$$

Combining these last two inequalities with Eq. (4.5) and simplifying, we get the following

$$\left\|\hat{w}_\theta(Z^{(i)})-\hat{w}_\theta(Z)\right\|_\theta\leq\frac{1}{\lambda n}\left(\left\|x_i's_{\theta,i}'\right\|_{\theta,*}+\left\|x_is_{\theta,i}\right\|_{\theta,*}\right). \tag{4.8}$$

Hence, combining the first row in Eq. (4.7) with Eq. (4.8), we can write

$$\ell(\langle x_i',\hat{w}_\theta(Z)\rangle,y_i')-\ell(\langle x_i',\hat{w}_\theta(Z^{(i)})\rangle,y_i')\leq\frac{1}{\lambda n}\left(\left\|x_i's_{\theta,i}'\right\|_{\theta,*}^2+\left\|x_i's_{\theta,i}'\right\|_{\theta,*}\left\|x_is_{\theta,i}\right\|_{\theta,*}\right). \tag{4.9}$$

Now, taking the expectation w.r.t. $Z\sim\mu^n$ and $z_i'\sim\mu$ of the left side member above, according to (Bousquet and Elisseeff, 2002, Lemma 7), we get

$$\mathbb{E}_{Z\sim\mu^n}\,\mathbb{E}_{z_i'\sim\mu}\left[\ell(\langle x_i',\hat{w}_\theta(Z)\rangle,y_i')-\ell(\langle x_i',\hat{w}_\theta(Z^{(i)})\rangle,y_i')\right]=\mathbb{E}_{Z\sim\mu^n}\left[\mathcal{R}_\mu(\hat{w}_\theta(Z))-\mathcal{R}_Z(\hat{w}_\theta(Z))\right].$$

Finally, taking the expectation of the right side member, exploiting the fact that the points are i.i.d. according μ, we get

$$\mathbb{E}_{Z\sim\mu^n}\,\mathbb{E}_{z_i'\sim\mu}\,\frac{1}{\lambda n}\left(\left\|x_i' s_{\theta,i}'\right\|_{\theta,*}^2+\left\|x_i' s_{\theta,i}'\right\|_{\theta,*}\left\|x_i s_{\theta,i}\right\|_{\theta,*}\right)\leq\frac{2}{\lambda n}\,\mathbb{E}_{Z\sim\mu^n}\,\mathbb{E}_{z_i'\sim\mu}\left\|x_i' s_{\theta,i}'\right\|_{\theta,*}^2, \quad (4.10)$$

where we recall that $s_{\theta,i}'\in\partial\ell(\cdot,y_i')(\langle x_i',\hat{w}_\theta(Z)\rangle)$. The statement derives from combining the two last statements above with the expectation w.r.t. $Z\sim\mu^n$ and $z_i'\sim\mu$ of Eq. (4.9). ■

We now have all the ingredients necessary to give an upper bound for the first term in Eq. (2.15), which is equivalent to give an expected meta-excess risk bound for $\hat{w}_{\bar{\theta}}$, the RERM algorithm in Eq. (2.24) with the meta-parameter $\bar{\theta}$, the average of the meta-parameters returned by the variant of our Alg. 3 working with exact meta-subgradients computed ad described above in Prop. 12.

Theorem 14 (OWB Meta-Excess Risk Bound)**.** *Consider the statistical setting. Let Asm. 1 and Asm. 2 hold. Let $A_{\bar{\theta}}$ be the inner RERM algorithm in Eq. (2.24) with meta-parameter $\bar{\theta}$, the average of the meta-parameters $(\theta_t)_{t=1}^T$ returned by the meta-algorithm in Alg. 3 using the data $\mathbf{Z}=(Z_t)_{t=1}^T$ with $Z_t=(x_{t,i},y_{t,i})_{i=1}^n$, by means of the exact meta-subgradients $(\hat{\nabla}_{\theta_t}')_{t=1}^T$, computed as described above in Prop. 12. For a distribution $\mu\sim\rho$, fix a dataset $Z=(x_i,y_i)_{i=1}^n\sim\mu^n$ independent from $\mathbf{Z}$ and, for any $i\in\{1,\ldots,n\}$, fix a datapoint $z_i'=(x_i',y_i')\sim\mu$ independent from Z and $\mathbf{Z}$. Let $\hat{w}_{\bar{\theta}}$ be the output generated by applying $A_{\bar{\theta}}$ to the dataset Z and consider $s_{\bar{\theta},i}'\in\partial\ell(\cdot,y_i')(\langle x_i',\hat{w}_{\bar{\theta}}\rangle)$. Then, in expectation w.r.t. the sampling of the data $\mathbf{Z}$, recalling the minimum norm (true) risk minimizer w_μ associated to a distribution $\mu\sim\rho$, for any $\theta\in\Theta$ such that $\mathbb{E}_{\mu\sim\rho}\,f(w_\mu,\theta)<+\infty$, the following expected meta-excess risk bound holds for $\hat{w}_{\bar{\theta}}$*

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_\mu(A_{\bar{\theta}})\leq\lambda\,\mathbb{E}_{\mu\sim\rho}\,f(w_\mu,\theta)+\frac{2}{\lambda n}\,\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\,\mathbb{E}_{z_i'\sim\mu}\left\|x_i' s_{\bar{\theta},i}'\right\|_{\bar{\theta},*}^2\\+\frac{\eta F(\theta)}{T}+\frac{1}{2\eta T}\,\mathbb{E}_{\mathbf{Z}}\sum_{t=1}^T\left\|\hat{\nabla}_{\theta_t}'\right\|_*^2.\end{aligned} \quad (4.11)$$

Proof. For any $\theta\in\Theta$, we consider the following decomposition

$$\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_\mu(A_{\bar{\theta}})=\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\left[\mathcal{R}_\mu(\hat{w}_{\bar{\theta}})-\mathcal{R}_\mu(w_\mu)\right]=\mathrm{A}+\mathrm{B}+\mathrm{C} \quad (4.12)$$

where we have introduced the following terms

$$\begin{aligned}
\mathrm{A} &= \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \Big[\mathcal{R}_\mu(\hat{w}_{\bar{\theta}}) - \mathcal{R}_Z(\hat{w}_{\bar{\theta}})\Big] \\
\mathrm{B} &= \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \Big[\mathcal{R}_Z(\hat{w}_{\bar{\theta}}) - \mathcal{R}_{\theta,Z}(\hat{w}_\theta)\Big] \\
\mathrm{C} &= \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \Big[\mathcal{R}_{\theta,Z}(\hat{w}_\theta) - \mathcal{R}_\mu(w_\mu)\Big].
\end{aligned} \tag{4.13}$$

We bound the term A by applying Prop. 13 with $\theta = \bar{\theta}$. In this way, we obtain

$$\mathrm{A} \leq \frac{2}{\lambda n}\, \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z_i'\sim\mu} \left\| x_i' s_{\bar{\theta},i}' \right\|_{\bar{\theta},*}^2. \tag{4.14}$$

We now observe that, for any $\theta \in \Theta$ such that $\mathbb{E}_{\mu\sim\rho}\, f(w_\mu, \theta) < +\infty$, by definition of $\hat{w}_\theta$, since w_μ does not depend on the dataset Z, we can write

$$\begin{aligned}
\mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_{\theta,Z}(\hat{w}_\theta) &\leq \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{R}_Z(w_\mu) + \lambda\, \mathbb{E}_{\mu\sim\rho}\, f(w_\mu, \theta) \\
&= \mathbb{E}_{\mu\sim\rho}\, \mathcal{R}_\mu(w_\mu) + \lambda\, \mathbb{E}_{\mu\sim\rho}\, f(w_\mu, \theta).
\end{aligned} \tag{4.15}$$

Moving the terms in the above inequality, we get the following upper bound for the term C:

$$\mathrm{C} \leq \lambda\, \mathbb{E}_{\mu\sim\rho}\, f(w_\mu, \theta). \tag{4.16}$$

We now bound the term B. Similarly to what observed in Thm. 7, making the identification $\alpha_m' \rightsquigarrow \hat{\nabla}_{\theta_t}'$ for the (exact) meta-subgradients, the meta-algorithm we are using coincides with the primal-dual Alg. 1 applied to the outer-task problem in Eq. (3.1), but this time, with exact meta-subgradients. Hence, specializing the first point of Thm. 2 to this setting, since the associated dual optimality gap Δ_{Dual} introduced in Thm. 2 is non-negative by definition, normalizing by the number of tasks T, for any $\theta \in \Theta$, we get

$$\begin{aligned}
\frac{1}{T} \sum_{t=1}^T \mathcal{L}_t(\theta_t) &\leq \frac{1}{T} \min_{\theta\in\mathcal{M}} \left\{ \sum_{t=1}^T \mathcal{L}_t(\theta) + \eta F(\theta) \right\} + \frac{1}{2\eta T} \sum_{t=1}^T \left\| \hat{\nabla}_{\theta_t}' \right\|_*^2 \\
&\leq \frac{1}{T} \sum_{t=1}^T \mathcal{L}_t(\theta) + \frac{\eta F(\theta)}{T} + \frac{1}{2\eta T} \sum_{t=1}^T \left\| \hat{\nabla}_{\theta_t}' \right\|_*^2.
\end{aligned} \tag{4.17}$$

Rearranging the terms, we get

$$\frac{1}{T} \sum_{t=1}^T \Big(\mathcal{L}_t\big(\theta_t\big) - \mathcal{L}_t(\theta) \Big) \leq \frac{\eta F(\theta)}{T} + \frac{1}{2\eta T} \sum_{t=1}^T \left\| \hat{\nabla}_{\theta_t}' \right\|_*^2. \tag{4.18}$$

The bound on the term B derives from the following relations.

$$\begin{aligned}
\mathrm{B} &= \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \left[\mathcal{R}_Z(\hat{w}_{\bar{\theta}}) - \mathcal{R}_{\theta,Z}(\hat{w}_\theta)\right] \\
&\leq \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \left[\mathcal{R}_{\bar{\theta},Z}(\hat{w}_{\bar{\theta}}) - \mathcal{R}_{\theta,Z}(\hat{w}_\theta)\right] \\
&= \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \left[\mathcal{L}_Z(\bar{\theta}) - \mathcal{L}_Z(\theta)\right] \\
&\leq \mathbb{E}_{\mathbf{Z}}\, \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \left[\mathcal{L}_Z(\theta_t) - \mathcal{L}_Z(\theta)\right] \\
&= \mathbb{E}_{\mathbf{Z}}\, \frac{1}{T}\sum_{t=1}^{T} \Big(\mathcal{L}_t(\theta_t) - \mathcal{L}_t(\theta)\Big) \\
&\leq \frac{\eta F(\theta)}{T} + \frac{1}{2\eta T}\, \mathbb{E}_{\mathbf{Z}} \sum_{t=1}^{T} \left\|\hat{\nabla}'_{\theta_t}\right\|_*^2 .
\end{aligned} \tag{4.19}$$

In the first inequality above we have exploited the non-negativity of f provided by Asm. 1, in the second equality we have used the definition of $\mathcal{L}_Z$, in the second inequality we have applied Jensen's inequality (see Lemma 39 in App. A) to the convex function $\mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathcal{L}_Z$, in the third equality we have exploited the i.i.d. sampling of the datasets and the fact that θ_t depends only on $(Z_j)_{j=1}^{t-1}$ and, finally, in the last inequality we have applied Eq. (4.18). The desired statement derives from combining the bounds in Eq. (4.14), Eq. (4.16) and Eq. (4.19). ■

The bound we have obtained in Thm. 9 for our OWO method and the bound in Thm. 14 for the corresponding OWB variant present a similar structure, hiding possible delicate dependencies. In the settings we will consider in the following, the norm of the approximated meta-subgradients and the norm of the exact ones will be upper bounded by the same value. Thus, the difference between the bounds will be due to the different terms going as $1/\lambda$. As we will see, this difference will translate into an additional logarithmic factor $\log(n)$ in the OWO bound.

In the next subsection, we introduce a way to measure the performance of the OWB variant of our Meta-Learning approach. In other words, we give an upper bound on the second term in Eq. (2.15).

4.1.1 The Benchmark for the Method

As done for the OWO Meta-Learning method described in the chapter above, we compare the performance of the OWB variant with the best algorithm in our class, solving the problem in Eq. (2.13):

$$\min_{\theta\in\Theta}\ \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta), \tag{2.13}$$

where, this time, A_θ denotes the inner RERM in Eq. (2.24) and $\mathcal{E}_\mu(A_\theta)$ represents its expected excess risk over the task $\mu \sim \rho$. In the following result we give an upper bound for the above quantity for a meta-parameter in a subset of Θ containing, as we will see in the following, the optimal meta-parameter θ_ρ. This bound automatically translates into an expected across-tasks excess risk bound for the batch RERM inner algorithm $\hat{w}_\theta$ in Eq. (2.24) with an appropriate meta-parameter θ fixed in hindsight for any task. Such a bound will be used as benchmark for the corresponding bound we have obtained in Thm. 14 for the meta-parameter estimated by our OWB Meta-Learning procedure.

Theorem 15 (Across-Tasks Excess Risk Bound for the RERM Algorithm in Eq. (2.24))**.** *Consider the statistical setting and let Asm. 1 hold. For a distribution $\mu \sim \rho$, fix a dataset $Z = (x_i, y_i)_{i=1}^n \sim \mu^n$ and, for any $i \in \{1, \dots, n\}$, fix a datapoint $z_i' = (x_i', y_i') \sim \mu$ independent from Z. For any $\theta \in \Theta$, let A_θ be the corresponding inner RERM algorithm in Eq. (2.24). Let $\hat{w}_\theta$ be the output returned by A_θ over the dataset Z and let $s_{\theta,i}' \in \partial \ell(\cdot, y_i')(\langle x_i', \hat{w}_\theta \rangle)$. Then, recalling the minimum norm (true) risk minimizer w_μ associated to a distribution $\mu \sim \rho$, for any $\theta \in \Theta$ such that $\mathbb{E}_{\mu\sim\rho} f(w_\mu, \theta) < +\infty$, the following expected across-tasks excess risk bound holds for $\hat{w}_\theta$*

$$\mathbb{E}_{\mu\sim\rho} \mathcal{E}_\mu(A_\theta) \leq \lambda \, \mathbb{E}_{\mu\sim\rho} f(w_\mu, \theta) + \frac{2}{\lambda n} \mathbb{E}_{\mu\sim\rho} \mathbb{E}_{Z\sim\mu^n} \mathbb{E}_{z_i'\sim\mu} \left\| x_i' s_{\theta,i}' \right\|_{\theta,*}^2. \tag{4.20}$$

Proof. For any $\theta \in \Theta$ such that $\mathbb{E}_{\mu\sim\rho} f(w_\mu, \theta) < +\infty$, we consider the following decomposition

$$\mathbb{E}_{Z\sim\mu^n} \Big[\mathcal{R}_\mu(\hat{w}_\theta) - \mathcal{R}_\mu(w_\mu) \Big] = \mathrm{A} + \mathrm{B} \tag{4.21}$$

where we have introduced the terms

$$\begin{aligned} \mathrm{A} &= \mathbb{E}_{Z\sim\mu^n} \Big[\mathcal{R}_\mu(\hat{w}_\theta) - \mathcal{R}_Z(\hat{w}_\theta) \Big] \\ \mathrm{B} &= \mathbb{E}_{Z\sim\mu^n} \Big[\mathcal{R}_Z(\hat{w}_\theta) - \mathcal{R}_\mu(w_\mu) \Big]. \end{aligned} \tag{4.22}$$

We bound the term A by Prop. 13, getting

$$\mathrm{A} \leq \frac{2}{\lambda n} \mathbb{E}_{\mu\sim\rho} \mathbb{E}_{Z\sim\mu^n} \mathbb{E}_{z_i'\sim\mu} \left\| x_i' s_{\theta,i}' \right\|_{\theta,*}^2. \tag{4.23}$$

Regarding the term B, exploiting the non-negativity of f (provided by Asm. 1) and applying the same reasoning in Eq. (4.15), we get

$$\begin{aligned} \mathrm{B} = \mathbb{E}_{Z\sim\mu^n} \Big[\mathcal{R}_Z(\hat{w}_\theta) - \mathcal{R}_\mu(w_\mu) \Big] &\leq \mathbb{E}_{Z\sim\mu^n} \Big[\mathcal{R}_{\theta,Z}(\hat{w}_\theta) - \mathcal{R}_\mu(w_\mu) \Big] \\ &\leq \lambda f(w_\mu, \theta). \end{aligned} \tag{4.24}$$

The desired statement derives from combining the two terms in Eq. (4.23) and Eq. (4.24) above and taking the expectation w.r.t. $\mu \sim \rho$ in Eq. (4.21). ■

At this point of the presentation, the observations that we can give on the comparison between the performance of the OWB Meta-Learning method in Eq. (4.11) and the guarantees for the benchmark method in Eq. (4.20) are similar to the ones we have given in the previous Chpt. 3 for the OWO setting. Moreover, the benchmark bounds in Eq. (3.46) for the online Alg. 2 and the one in Eq. (4.20) for the batch RERM algorithm in Eq. (2.24) differ only in the term $1/\lambda$. Also in this case, as we will see in the following, this difference will translate into an additional logarithmic factor $\log(n)$ for the bound regarding the online Alg. 2.

In the next section, we describe how this OWB variant of our method fits in the recent literature.

4.2 Related Work

In (Bullins et al., 2019) the authors consider the statistical OWB setting for the family of the RERM algorithms in Eq. (2.24) with the same regularizer f introduced in Ex. 2 with a Lipschitz loss function. In order to estimate from the data the matrix $\theta \in \mathbb{S}_+^d$ parametrizing the inner algorithm, the authors propose to apply as meta-algorithm Frank-Wolfe or Exponentiated-Weighted by using the same meta-objectives $(\mathcal{L}_t)_{t=1}^T$ considered in this work. As we will see in the following Chpt. 6, the meta-algorithm we will use for this setting will be different.

We now briefly describe the content of the other two papers, (Denevi et al., 2018a) and (Denevi et al., 2018b), published during the PhD.

In (Denevi et al., 2018a) we focus on the statistical setting where the inner algorithm is Ridge Regression parametrized by a matrix $\theta \in \mathbb{S}_+^d$ as in Ex. 2. In other words, we consider the family of the within-task RERM algorithms in Eq. (2.24) with the same regularizer f introduced in Ex. 2 and with the square loss, i.e.

$$\hat{w}_\theta(Z) = \operatorname*{argmin}_{w \in \mathrm{Ran}(\theta)} \mathcal{R}_Z(w) + \frac{\lambda}{2} \langle w, \theta^\dagger w \rangle \qquad \mathcal{R}_Z(w) = \frac{1}{2n} \sum_{i=1}^n (\langle x_i, w \rangle - y_i)^2. \tag{4.25}$$

The meta-algorithm we propose to learn this matrix consists in a stochastic projected gradient descent scheme aiming at solving the surrogate problem

$$\min_{\theta \in \mathcal{S}} \mathbb{E}_{\mu \sim \rho} \mathbb{E}_{Z \sim \mu^n} \mathcal{R}_Z(\hat{w}_\theta(Z)), \tag{4.26}$$

where $\mathcal{S} = \{\theta \in \mathbb{S}_+^d : \mathrm{Tr}(\theta) \leq 1\}$ as in Ex. 2. The proposed approach processes one dataset (task) at the time, without the need to store previously encountered datasets. We observe that, in this case, differently from the setting described in the previous section, the surrogate function we use coincides with the empirical error $\mathcal{R}_Z(\hat{w}_\theta(Z))$ (without regularization term) incurred by the above variant of Ridge Regression in Eq. (4.25). While in general nothing can be said about the convexity of the above problem in Eq. (4.26), in this case, exploiting the specific closed form of the inner algorithm in Eq. (4.25), we manage to show that the above surrogate problem is a convex problem w.r.t. the meta-parameter θ. This allows us to give a non-asymptotic bound for the excess risk of the algorithm resulting from our procedure. A remarkable feature of our learning bound is that it is comparable to previous bounds for batch Meta-Learning. This is confirmed also by the experiments, where our OWB Meta-Learning method results to be competitive with the more expensive batch counterpart and outperforms solving the tasks independently, when the tasks satisfy the similarity assumption of sharing a low-rank linear feature map. The proof technique used in this work leverages previous work on Meta-Learning (specifically, the work (Maurer, 2009)) with tools from Online Learning.

In (Denevi et al., 2018b) we consider again the statistical setting, but this time, as in Ex. 1, we consider the family of inner algorithms given by a variant of Ridge Regression, in which the regularizer is the square distance to an unknown bias vector $\theta \in \mathbb{R}^d$. In other words, we consider the family of the within-task RERM algorithms in Eq. (2.24) with the same regularizer f introduced in Ex. 1 and with the square loss, i.e.

$$\hat{w}_\theta(Z) = \operatorname*{argmin}_{w \in \mathbb{R}^d} \mathcal{R}_Z(w) + \frac{\lambda}{2} \|w - \theta\|_2^2 \qquad \mathcal{R}_Z(w) = \frac{1}{2n} \sum_{i=1}^{n} (\langle x_i, w \rangle - y_i)^2. \tag{4.27}$$

Motivated by recent empirical studies on few shot-learning and Meta-Learning (Finn et al., 2017; Ravi and Larochelle, 2017), the meta-algorithm we propose to learn this bias vector from the data splits each dataset Z into two subsets $(Z_{\mathrm{tr}}, Z_{\mathrm{te}})$ used to train and to test the inner algorithm (respectively) and then it applies stochastic gradient descent on the surrogate problem

$$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_{\mu \sim \rho} \mathbb{E}_{Z=(Z_{\mathrm{tr}}, Z_{\mathrm{te}}) \sim \mu^n} \mathcal{R}_{Z_{\mathrm{te}}}(\hat{w}_\theta(Z_{\mathrm{tr}})), \tag{4.28}$$

where $\mathcal{R}_{Z_{\mathrm{te}}}(\hat{w}_\theta(Z_{\mathrm{tr}}))$ is the empirical error over the test set Z_{te} of the inner algorithm $\hat{w}_\theta$ in Eq. (4.27) trained over the training set Z_{tr}. Also in this case, the meta-algorithm processes one dataset (task) at the time and it does not need to store the previous datasets. A key aspect of our analysis is the fact that, exploiting the specific closed form of the inner algorithm in Eq. (4.27) (an affine transformation of the meta-parameter θ), we manage to show that the above surrogate problem in Eq. (4.28) is a Least Squares function w.r.t. the meta-parameter θ. This

allows us to provide, under specific assumptions, a statistical analysis for the proposed approach, which highlights the role of the splitting parameter, namely, the number of points we use to train the inner algorithm. Preliminary experiments confirm our theoretical findings, highlighting the advantage of our approach w.r.t. solving the tasks independently when the tasks satisfy the similarity assumption of having a small variance. Our proof technique leverages previous work on stochastic optimization for Least Squares (specifically, the work (Dieuleveut et al., 2017)) with tools from classic Meta-Learning theory, such as the ones in (Baxter, 2000; Maurer, 2005, 2009; Maurer et al., 2016).

4.3 Discussion

In this section we have introduced the corresponding OWB variant of our OWO Meta-Learning method and we have analyzed its guarantees in the statistical setting. Similarly to the OWO setting, also in this case, the results are not jet ready to be interpreted, because of the presence of some possible hidden dependencies. For this reason, we postpone a detailed discussion about the results we have obtained to the specific settings of Ex. 1 and Ex. 2, analyzed in the following Chpt. 5 and Chpt. 6, respectively. As we will see, in these settings, we will manage to show that, in determined regimes, the guarantees provided by the more expensive OWB variant can be comparable to the ones for the more efficient OWO variant. This is a supporting point for our fully online method.

Chapter 5

Example 1. Bias

In this chapter we specify our Meta-Learning framework to the setting in Ex. 1 outlined at the end of Sec. 3.1. We recall that, in such a case, the meta-parameter coincides with a bias vector $\theta \in \mathbb{R}^d$ and, as we will see in the following, the tasks' similarity translates into the existence of a bias vector closed to the tasks' target vectors. We start this chapter by specializing in Sec. 5.1 our general OWO method described in Chpt. 3 to Ex. 1, deriving the corresponding inner and meta-algorithm. The method is then analyzed in Sec. 5.2 and Sec. 5.3, where we specify the meta-regret bound in Thm. 7 and the meta-excess risk bound in Thm. 9, respectively. After this, in the subsequent Sec. 5.4, we describe the OWB variant of our method introduced in Chpt. 4. Finally, in Sec. 5.5 and Sec. 5.6, we discuss the results and we report the numerical evaluation of our method, respectively.

In this chapter, we will require the following assumption, which is for instance satisfied by the absolute loss $\ell(\hat{y}, y) = |\hat{y} - y|$ and the hinge loss $\ell(\hat{y}, y) = \max\{0, 1 - y\hat{y}\}$, where $y, \hat{y} \in \mathcal{Y}$.

Assumption 3 (Lipschitz Loss)**.** *Let $\ell(\cdot, y)$ be L-Lipschitz for any $y \in \mathcal{Y}$, where $L > 0$.*

In addition to this, for any task $t \in \{1, \dots, T\}$, we introduce the following quantities related to the inputs' covariance matrices

$$C_t = \frac{1}{n}\sum_{i=1}^{n} x_{t,i}x_{t,i}^\top \quad \hat{C}_t = \sum_{i=1}^{n} \frac{1}{i}\, x_{t,i}x_{t,i}^\top \quad C^{\text{tot}} = \frac{1}{T}\sum_{t=1}^{T} C_t \quad \hat{C}^{\text{tot}} = \frac{1}{T}\sum_{t=1}^{T} \hat{C}_t \tag{5.1}$$

$$\|C^{\text{tot}}\|_{\infty,a} = \frac{1}{T}\sum_{t=1}^{T} \|C_t\|_\infty^a \text{ with } a = 1, 2. \tag{5.2}$$

Algorithm 4 Within-Task Algorithm for Ex. 1

Input $\lambda > 0$, $\theta \in \mathbb{R}^d$, $Z = (z_i)_{i=1}^n$

Initialization $s_{\theta,1} = ()$, $w_{\theta,1} = \theta$

For $i = 1$ to n

Receive the datapoint $z_i = (x_i, y_i)$

Compute $s'_{\theta,i} \in \partial \ell_i(\langle x_i, w_{\theta,i} \rangle) \subseteq \mathbb{R}$

Define $(s_{\theta,i+1})_i = s'_{\theta,i}$, $\gamma_i = \lambda(i+1)$

Define $p_{\theta,i} = x_i s'_{\theta,i} + \lambda(w_{\theta,i} - \theta)$

Update $w_{\theta,i+1} = w_{\theta,i} - 1/\gamma_i \, p_{\theta,i}$

Return $(w_{\theta,i})_{i=1}^{n+1}$, $\bar{w}_\theta = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} w_{\theta,i}$, $s_{\theta,n+1}$

Algorithm 5 Meta-Algorithm for Ex. 1

Input $\eta > 0$, $\mathbf{Z} = (Z_t)_{t=1}^T$

Initialization $\theta_1 = 0 \in \mathbb{R}^d$

For $t = 1$ to T

Receive incrementally the dataset Z_t

Run Alg. 4 with θ_t over Z_t

Compute $s_{\theta_t,n+1}$

Define $\nabla'_{\theta_t} = X_t^\top s_{\theta_t,n+1}/n$

Update $\theta_{t+1} = \theta_t - \nabla'_{\theta_t}/\eta$

Return $(\theta_t)_{t=1}^{T+1}$, $\bar{\theta} = \dfrac{1}{T} \displaystyle\sum_{t=1}^{T} \theta_t$

In the statistical setting, we let also

$$C_\rho = \mathbb{E}_{\mu \sim \rho} \, \mathbb{E}_{(x,y) \sim \mu} \, x x^\top. \tag{5.3}$$

5.1 Deriving the Method

We start from specializing the generic inner algorithm in Alg. 2 and the generic meta-algorithm in Alg. 3 to the setting outlined in Ex. 1. The algorithms we obtain are reported in Alg. 4 and Alg. 5, respectively, where, $X_t \in \mathbb{R}^{n \times d}$ denotes the input vectors' matrix of the task t, having as i–th row the input vector $x_{t,i}$. The deduction is reported in Lemma 16 and Lemma 17 below, respectively.

We start from the deduction of the inner algorithm in Alg. 4.

Lemma 16 (Derivation of the Inner Alg. 4, Bias)**.** *For any $i \in \{0, \dots, n\}$, let $w_{\theta,i+1}$ be the update of the (primal) variable deriving from applying Alg. 2 to the dataset $Z = (x_i, y_i)_{i=1}^n$ in the setting outlined in Ex. 1 with bias $\theta \in \mathbb{R}^d$. Let $s'_{\theta,i} \in \partial \ell_i(\langle x_i, w_{\theta,i} \rangle)$ be the subgradient used by such an algorithm to compute $w_{\theta,i+1}$. Then, $w_{\theta,1} = \theta$ and, for any $i \in \{1, \dots, n\}$, introducing the subgradient of the regularized loss*

$$p_{\theta,i} = x_i s'_{\theta,i} + \lambda(w_{\theta,i} - \theta) \in \partial \Big(\ell_i(\langle x_i, \cdot \rangle) + \frac{\lambda}{2} \, \| \cdot - \theta \|_2^2 \Big)(w_{\theta,i}), \tag{5.4}$$

we have

$$w_{\theta,i+1} = w_{\theta,i} - \frac{1}{\lambda(i+1)}\, p_{\theta,i}. \tag{5.5}$$

Proof. We start from observing that, according to the choices made in Ex. 1, for any $\theta, w, u \in \mathbb{R}^d$, we have

$$f(w,\theta) = \frac{1}{2}\,\|w-\theta\|_2^2 \qquad f(\cdot,\theta)^*(u) = \frac{1}{2}\,\|u\|_2^2 + \langle u,\theta\rangle \qquad \nabla f(\cdot,\theta)^*(u) = u+\theta.$$

Consequently, according to the definition of $w_{\theta,1}$ in Alg. 2, we have

$$w_{\theta,1} = \nabla f(\cdot,\theta)^*(0) = \theta. \tag{5.6}$$

We now show the desired closed form of $w_{\theta,i+1}$ for any $i \in \{1,\dots,n\}$. In such a case, denoting by $X_{1:i} \in \mathbb{R}^{i\times d}$ the matrix containing the first i input vectors as rows, by definition of $w_{\theta,i+1}$ in Alg. 2, we can write

$$w_{\theta,i+1} = \nabla f(\cdot,\theta)^*\left(-\frac{1}{\lambda(i+1)}\, X_{1:i}^\top s_{\theta,i+1}\right) = -\frac{1}{\lambda(i+1)}\, X_{1:i}^\top s_{\theta,i+1} + \theta. \tag{5.7}$$

For $i = 1$ the statement holds, as a matter of fact, since $w_{\theta,1} = \theta$, exploiting Eq. (5.7) and introducing the subgradient $p_{\theta,1} = x_1 s'_{\theta,1} + \lambda(w_{\theta,1} - \theta) = x_1 s'_{\theta,1}$, we can write

$$w_{\theta,2} = -\frac{1}{2\lambda}\, x_1 s'_{\theta,1} + \theta = w_{\theta,1} - \frac{1}{2\lambda}\, p_{\theta,1}. \tag{5.8}$$

Now, we show that the statement holds also for $i \in \{2,\dots,n\}$. Since $X_{1:i}^\top s_{\theta,i+1} = X_{1:i-1}^\top s_{\theta,i} + x_i s'_{\theta,i}$, we can write the following

$$\begin{aligned} w_{\theta,i+1} &= -\frac{1}{\lambda(i+1)}\, X_{1:i}^\top s_{\theta,i+1} + \theta = -\frac{1}{\lambda(i+1)}\left(X_{1:i-1}^\top s_{\theta,i} + x_i s'_{\theta,i}\right) + \theta \\ &= \frac{\lambda i}{\lambda(i+1)}\left(-\frac{1}{\lambda i} X_{1:i-1}^\top s_{\theta,i}\right) - \frac{x_i s'_{\theta,i}}{\lambda(i+1)} + \theta \\ &= \frac{\lambda(i+1)(w_{\theta,i}-\theta) - x_i s'_{\theta,i} - \lambda(w_{\theta,i}-\theta)}{\lambda(i+1)} + \theta \\ &= \frac{\lambda(i+1) w_{\theta,i} - p_{\theta,i}}{\lambda(i+1)} = w_{\theta,i} - \frac{1}{\lambda(i+1)} p_{\theta,i}, \end{aligned} \tag{5.9}$$

where, in the first and the fourth equality, we have exploited Eq. (5.7) and in the fifth equality we have exploited the form of the subgradient $p_{\theta,i} = x_i s'_{\theta,i} + \lambda(w_{\theta,i} - \theta)$. ■

We now proceed with the deduction of the meta-algorithm in Alg. 5.

Lemma 17 (Derivation of the Meta-Algorithm in Alg. 5, Bias)**.** *For any $t \in \{0, \dots, T\}$, let θ_{t+1} be the update of the variable deriving from applying Alg. 3 to the data $\mathbf{Z} = (Z_t)_{t=1}^T$ in the setting outlined in Ex. 1. Let ∇'_{θ_t} be the approximated meta-subgradient computed as described in Prop. 6 and used by the algorithm to compute θ_{t+1}. Then, $\theta_1 = 0 \in \mathbb{R}^d$ and, for any $t \in \{1, \dots, T\}$, we have*

$$\theta_{t+1} = \theta_t - \frac{1}{\eta} \nabla'_{\theta_t}. \tag{5.10}$$

Moreover, for any $t \in \{1, \dots, T\}$, we have

$$\nabla'_{\theta_t} = \frac{1}{n} X_t^\top s_{\theta_t, n+1}, \tag{5.11}$$

where $s_{\theta_t, n+1} \in \mathbb{R}^n$ is the output of Alg. 5 with bias vector θ_t over the dataset Z_t and, under Asm. 3, according to the notation in Eq. (5.1),

$$\left\| \nabla'_{\theta_t} \right\|_2^2 \leq L^2 \|C_t\|_\infty. \tag{5.12}$$

Finally, the updating step and the bound above hold also for the exact meta-subgradients computed as described in Prop. 12, which are given, for any $t \in \{1, \dots T\}$, by

$$\hat{\nabla}'_{\theta_t} = \frac{1}{n} X_t^\top \hat{s}_{\theta_t} = -\lambda(\hat{w}_{\theta_t} - \theta_t), \tag{5.13}$$

where $\hat{w}_{\theta_t}$ and $\hat{s}_{\theta_t}$ denote, respectively, the RERM algorithm in Eq. (2.24) and a solution of the associated dual problem with meta-parameter θ_t and dataset Z_t for the setting in Ex. 1.

Proof. We start from observing that, according to the choices made in Ex. 1, for any $k, \theta, u \in \mathbb{R}^d$, we have

$$F(\theta) = \frac{1}{2} \|\theta\|_2^2 \qquad F^*(k) = \frac{1}{2} \|k\|_2^2 \qquad \nabla F^*(k) = k \qquad f(\cdot, \theta)^*(u) = \frac{1}{2} \|u\|_2^2 + \langle u, \theta \rangle.$$

Consequently, according to the definition of θ_1 in Alg. 3, we have

$$\theta_1 = \nabla F^*(0) = 0. \tag{5.14}$$

We now show the desired closed form of θ_{t+1}, for any $t \in \{1, \dots, T\}$. In such a case, by the definition of θ_{t+1} in Alg. 3, we can write

$$\theta_{t+1} = \nabla F^* \Big(-\frac{1}{\eta} \sum_{j=1}^t \nabla'_{\theta_j} \Big) = -\frac{1}{\eta} \sum_{j=1}^t \nabla'_{\theta_j}. \tag{5.15}$$

For $t = 1$ the statement holds, as a matter of fact, since $\theta_1 = 0$, exploiting Eq. (5.15), we can write

$$\theta_2 = -\frac{1}{\eta}\,\nabla'_{\theta_1} = \theta_1 - \frac{1}{\eta}\,\nabla'_{\theta_1}. \tag{5.16}$$

For $t \in \{2, \dots, T\}$, we observe that, according to Eq. (5.15), we have

$$\theta_{t+1} = -\frac{1}{\eta}\sum_{j=1}^{t}\nabla'_{\theta_j} = -\frac{1}{\eta}\sum_{j=1}^{t-1}\nabla'_{\theta_j} - \frac{1}{\eta}\,\nabla'_{\theta_t} = \theta_t - \frac{1}{\eta}\,\nabla'_{\theta_t}. \tag{5.17}$$

Obviously, the above steps hold also when we substitute the approximated meta-subgradients $(\nabla'_{\theta_t})_{t=1}^T$ with the exact counterparts $(\hat{\nabla}'_{\theta_t})_{t=1}^T$. We now specify the closed form of the approximated meta-subgradients, computed as described in Prop. 6 for Ex. 1. We start from observing that adding to the notation in Prop. 6 the further task index t, by strong duality (see Lemma 5), we can rewrite

$$\mathcal{L}_t(\theta) = \max_{s\in\mathbb{R}^n} \tilde{D}_{t,n+1}(s,\theta) \qquad \tilde{D}_{t,n+1}(s,\theta) = -\frac{1}{n} D_{t,n+1}(s,\theta) \tag{5.18}$$

where, according to Eq. (3.10), in the setting outlined in Ex. 1,

$$\begin{aligned} -D_{t,n+1}(s,\theta) &= -\sum_{i=1}^{n}\ell^*_{t,i}(s_i) - \lambda n f(\cdot,\theta)^*\left(-\frac{1}{\lambda n}\sum_{i=1}^{n} x_{t,i}s_i\right) \\ &= -\sum_{i=1}^{n}\ell^*_{t,i}(s_i) - \lambda n f(\cdot,\theta)^*\left(-\frac{1}{\lambda n}X_t^\top s\right) \\ &= -\sum_{i=1}^{n}\ell^*_{t,i}(s_i) - \frac{1}{2\lambda n}\left\|X_t{}^\top s\right\|_2^2 + \left\langle X_t{}^\top s, \theta\right\rangle. \end{aligned} \tag{5.19}$$

Consequently, recalling that the output $s_{\theta_t,n+1}$ of the inner algorithm coincides with the last iterate of the corresponding dual inner iteration, according to Prop. 6, we have

$$\nabla_{\theta_t} = X_t{}^\top s_{\theta_t,n+1} \tag{5.20}$$

and, consequently,

$$\nabla'_{\theta_t} = \nabla_{\theta_t}/n \in \partial_{\epsilon_{\theta_t}/n}\,\mathcal{L}_t(\theta_t), \tag{5.21}$$

where ϵ_{θ_t} is outlined in Prop. 6 and it must be specified to Ex. 1. In order to prove Eq. (5.12), we start from observing that $s_{\theta_t,n+1}$ is the vector in $\mathbb{R}^n$ having as component i the subgradient $s'_{\theta_t,i} \in \partial\ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i}\rangle)$. Hence, under Asm. 3, exploiting Lemma 50 in App. A, any component of $s_{\theta_t,n+1}$ is absolutely bounded by L, and, consequently, $\|s_{\theta_t,n+1}\|_2 \le L\sqrt{n}$. This allows us to get the desired bound by applying Holder's inequality (see Lemma 32 in App. A) to the matrices'

scalar product as follows

$$\left\|\nabla'_{\theta_t}\right\|_2^2 = \frac{1}{n}\operatorname{Tr}\Big(\frac{1}{n}\sum_{i=1}^n x_{t,i}x_{t,i}^\top s_{\theta_t,n+1}s_{\theta_t,n+1}^\top\Big) \leq \frac{1}{n}\left\|\frac{1}{n}\sum_{i=1}^n x_{t,i}x_{t,i}^\top\right\|_\infty \|s_{\theta_t,n+1}\|_2^2$$
$$\leq L^2\left\|\frac{1}{n}\sum_{i=1}^n x_{t,i}x_{t,i}^\top\right\|_\infty = L^2\|C_t\|_\infty,$$

where in the last equality we have introduced the definition of C_t in Eq. (5.1). Finally, regarding the exact meta-subgradients, the first closed form in Eq. (5.13) directly derives from Prop. 12 and the former discussion, while the second closed form is deduced by the first KKT condition in Eq. (3.17) specified to Ex. 1 and to the task t:

$$\hat{w}_{\theta_t} = -\frac{1}{\lambda n}X_t^\top \hat{s}_{\theta_t} + \theta_t \qquad \hat{s}_\theta \in \partial\Big(\sum_{i=1}^n \ell_i\Big)\Big(\langle x_1, \hat{w}_\theta\rangle, \dots, \langle x_n, \hat{w}_\theta\rangle\Big). \tag{5.22}$$

Moreover, thanks to the second KKT condition above, under Asm. 3, by Lemma 43 and Lemma 50 in App. A, any component of $\hat{s}_{\theta_t}$ is absolutely bounded by L. Consequently, repeating the same steps above, the bound in Eq. (5.12) holds also for the exact meta-subgradients in Eq. (5.13). ■

We observe that the inner Alg. 4 we have deduced is a slightly different version of the inner algorithm used in (Denevi et al., 2019a) in the statistical setting, where the step size decreases as $1/(\lambda i)$ instead of $1/(\lambda(i+1))$. Instead, the meta-algorithm in Alg. 5 we have retrieved is exactly the same analyzed in that work. We refer to the discussion in Sec. 3.5 for more details about that work.

We note also that the bound we have given in Lemma 17 on the norm of the exact meta-subgradients and the approximated ones is the same. This tells us that, from our bounds, the error we introduce with such an approximation should not affect the overall performance of our Meta-Learning method. As we will see in the following, this statement will be confirmed also by our experiments, where the trick we use to approximate the meta-subgradients will reveal to be an effective strategy to keep the process fully online.

We also observe that for the setting in Ex. 1, our Meta-Learning method in Alg. 4 and Alg. 5 scales linearly with the dimension of the input space. We thus expect that, in this setting, the method will be appropriate also for datasets in more rich observation spaces.

In the next section, we analyze the performance of our OWO Meta-Learning method applied to Ex. 1, in the non-statistical setting.

5.2 Method and Analysis in the Non-Statistical Setting

In the next result we specify Thm. 7 to Ex. 1, that is, we provide a (regularized) average meta-regret bound for the procedure deriving from combining Alg. 4 with Alg. 5.

Corollary 18 (Meta-Regret Bound, Bias)**.** *Let Asm. 3 hold and consider the setting in Thm. 7 applied to Ex. 1. In particular, for any $\theta \in \mathbb{R}^d$, let A_θ be the corresponding inner Alg. 4 and let $(\theta_t)_{t=1}^T$ be the sequence of the bias vectors estimated by the meta-algorithm in Alg. 5 over the data $\mathbf{Z} = (Z_t)_{t=1}^T$. Recall also the minimum norm empirical risk minimizers $(\hat{w}_t)_{t=1}^T$ associated to the datasets $(Z_t)_{t=1}^T$. Then, introducing the empirical variance of the vectors $(\hat{w}_t)_{t=1}^T$ w.r.t. a bias vector $\theta \in \mathbb{R}^d$*

$$\hat{V}(\theta) = \frac{1}{2T} \sum_{t=1}^{T} \|\hat{w}_t - \theta\|_2^2, \tag{5.23}$$

according to the notation in Eq. (5.1) and Eq. (5.2), the following (regularized) average meta-regret bound holds for any $\theta \in \mathbb{R}^d$

$$\frac{1}{T} \sum_{t=1}^{T} \mathcal{E}_{\theta_t, Z_t}(A_{\theta_t}) \leq \lambda \hat{V}(\theta) + \frac{L^2 \mathrm{Tr}(\hat{C}^{\mathrm{tot}})}{2\lambda n} + \frac{\eta \|\theta\|_2^2}{2T} + \frac{L^2 \|C^{\mathrm{tot}}\|_{\infty,1}}{2\eta}. \tag{5.24}$$

Hence, optimizing w.r.t. the hyper-parameters λ and η, for

$$\lambda = L \sqrt{\frac{\mathrm{Tr}(\hat{C}^{\mathrm{tot}})}{2n\hat{V}(\theta)}} \qquad \eta = \frac{L \sqrt{T \, \|C^{\mathrm{tot}}\|_{\infty,1}}}{\|\theta\|_2}, \tag{5.25}$$

we get

$$\frac{1}{T} \sum_{t=1}^{T} \mathcal{E}_{\theta_t, Z_t}(A_{\theta_t}) \leq L \left(\sqrt{\frac{2\hat{V}(\theta) \mathrm{Tr}(\hat{C}^{\mathrm{tot}})}{n}} + \|\theta\|_2 \sqrt{\frac{\|C^{\mathrm{tot}}\|_{\infty,1}}{T}} \right). \tag{5.26}$$

Proof. Specializing Thm. 7 to the quantities outlined in Ex. 1, exploiting the bound on the norm of the approximated meta-subgradients given in Eq. (5.12) (exploiting Asm. 3) and using the notation in Eq. (5.23) and Eq. (5.2), for any $\theta \in \mathbb{R}^d$, we get

$$\frac{1}{T} \sum_{t=1}^{T} \mathcal{E}_{\theta_t, Z_t}(A_{\theta_t}) \leq \lambda \hat{V}(\theta) + \frac{1}{2\lambda n T} \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1}{i} \left\| x_{t,i} s'_{\theta_t, i} \right\|_2^2 + \frac{\eta \|\theta\|_2^2}{2T} + \frac{L^2 \|C^{\mathrm{tot}}\|_{\infty,1}}{2\eta}. \tag{5.27}$$

The statement derives from the above inequality observing that, under Asm. 3 using the definition of $\hat{C}^{\mathrm{tot}}$ in Eq. (5.1), we can write

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1}{i} \left\| x_{t,i} s'_{\theta_t, i} \right\|_2^2 \leq L^2 \mathrm{Tr}\left(\frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1}{i} \, x_{t,i} x_{t,i}^\top \right) = L^2 \mathrm{Tr}(\hat{C}^{\mathrm{tot}}). \tag{5.28}$$

■

Before continuing with the theoretical analysis of our method, in the following remark, we stress an important aspect regarding the tuning of the hyper-parameters in our method.

Remark 11 (Hyper-parameters Tuning in Our Method)**.** *We want to point out to the attention of the reader that the bound in Eq. (5.26) in the statement above requires oracle tuning of the hyper-parameters. As a matter of fact, the choice of hyper-parameters in Eq. (5.25) for which such a bound holds requires knowledge of quantities depending on the optimal competitors and the future sequence of data and, thus, not available in practice in our OWO Meta-Learning setting. As observed in previous literature, see e.g. (Shalev-Shwartz, 2007), when an estimate (upper bound) of these quantities is available in practice, the bound above and the corresponding interpretation we will give in the following must be intended with such quantities replaced with the corresponding estimates. In the more complicate setting in which such estimates are not available in practice, how to address the choice of the hyper-parameters in our method in a more effective and theoretically grounded way than tuning them over a grid of values becomes a fundamental and still open problem. In such a case, it would be desirable to have a method in which such quantities are stably estimated on the fly directly from the data and the hyper-parameters are self-tuned. A possible direction to reach this target may be to adapt the so-called 'parameter-free methods' proposed in (Mcmahan and Streeter, 2012; McMahan and Orabona, 2014; Orabona, 2014; Orabona and Pál, 2016; Zhuang et al., 2019) for the Single-Task Learning setting to an appropriate Meta-Learning setting. On the other hand, as already proven for the single task setting in (Mcmahan and Streeter, 2012), we expect, at least, that this self-tuning of the hyper-parameters will come at the price of unavoidable additional logarithmic terms. As the reader can note proceeding with the reading, the observations in this remark apply also to the bounds reported in the following, also in the statistical setting.*

In order to evaluate the quality of the bound above in Cor. 18, we specify Thm. 8 to Ex. 1, that is, we provide a (regularized) average across-tasks regret bound for the procedure deriving from running the within-task Alg. 4 with a bias vector fixed in hindsight for any task.

Corollary 19 (Across-Tasks Regret Bound for Alg. 4, Bias)**.** *Let Asm. 3 hold and consider the setting in Thm. 8 applied to Ex. 1. In particular, for any $\theta \in \mathbb{R}^d$, let A_θ be the corresponding inner Alg. 4. Then, according to the notation in Eq. (5.23) and Eq. (5.1), the following (regularized) average across-tasks regret bound holds for any $\theta \in \mathbb{R}^d$*

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{\theta,Z_t}(A_\theta) \leq \lambda \hat{V}(\theta) + \frac{L^2 \mathrm{Tr}(\hat{C}^{\mathrm{tot}})}{2\lambda n}. \tag{5.29}$$

Hence, optimizing w.r.t. the hyper-parameter λ, for

$$\lambda = L\sqrt{\frac{\mathrm{Tr}(\hat{C}^{\mathrm{tot}})}{2n\hat{V}(\theta)}}, \tag{5.30}$$

we get

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{\theta,Z_t}(A_\theta) \leq L\sqrt{\frac{2\hat{V}(\theta)\mathrm{Tr}(\hat{C}^{\mathrm{tot}})}{n}}. \tag{5.31}$$

Proof. Specializing Thm. 8 to the quantities outlined in Ex. 1, using the notation in Eq. (5.23), for any $\theta \in \mathbb{R}^d$, we get

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{\theta,Z_t}(A_\theta) \leq \lambda\hat{V}(\theta) + \frac{1}{2\lambda nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|x_{t,i}s'_{t,i}\right\|_2^2. \tag{5.32}$$

The statement derives from the above inequality observing that, under Asm. 3, using the definition of $\hat{C}^{\mathrm{tot}}$ in Eq. (5.1), we can write

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|x_{t,i}s'_{t,i}\right\|_2^2 \leq L^2\mathrm{Tr}\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}x_{t,i}x_{t,i}^\top\right) = L^2\mathrm{Tr}(\hat{C}^{\mathrm{tot}}). \tag{5.33}$$

■

We postpone to Sec. 5.5 a discussion about the results we reported above. In the next section, we analyze the performance of our OWO Meta-Learning method applied to Ex. 1, in the statistical setting.

5.3 Method and Analysis in the Statistical Setting

In the result below we specify Thm. 9 to Ex. 1, that is, we provide a (regularized) expected meta-excess risk bound for the average $\bar{w}_{\bar{\theta}}$ of the estimators returned by the combination of Alg. 4 with Alg. 5.

Corollary 20 (OWO Meta-Excess Risk Bound, Bias)**.** *Let Asm. 3 hold and consider the statistical setting in Thm. 9 applied to Ex. 1. In particular, let $A_{\bar{\theta}}$ be the inner Alg. 4 with bias $\bar{\theta}$, the average of the vectors returned by the meta-algorithm in Alg. 5 using the data $\mathbf{Z} = (Z_t)_{t=1}^T$. Recall also the minimum norm risk minimizer w_μ associated to a task $\mu \sim \rho$. Then, introducing the exact*

variance of the vectors w_μ w.r.t. a bias vector $\theta \in \mathbb{R}^d$

$$V_\rho(\theta) = \frac{1}{2}\,\mathbb{E}_{\mu\sim\rho}\,\|w_\mu - \theta\|_2^2, \tag{5.34}$$

according to the notation in Eq. (5.2) and Eq. (5.3), the following (regularized) expected meta-excess risk bound holds for any $\theta \in \mathbb{R}^d$

$$\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_{\bar\theta,\mu}(A_{\bar\theta}) \leq \lambda V_\rho(\theta) + \frac{(\log(n)+1)L^2\mathrm{Tr}(C_\rho)}{\lambda n} + \frac{\eta\|\theta\|_2^2}{2T} + \frac{L^2\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,1}}{2\eta}. \tag{5.35}$$

Hence, optimizing w.r.t. the hyper-parameters λ and η, for

$$\lambda = L\sqrt{\frac{(\log(n)+1)\mathrm{Tr}(C_\rho)}{nV_\rho(\theta)}} \qquad \eta = \frac{L\sqrt{T\,\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,1}}}{\|\theta\|_2}, \tag{5.36}$$

we get

$$\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_{\bar\theta,\mu}(A_{\bar\theta}) \leq L\left(2\sqrt{\frac{(\log(n)+1)V_\rho(\theta)\mathrm{Tr}(C_\rho)}{n}} + \|\theta\|_2\sqrt{\frac{\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,1}}{T}}\right). \tag{5.37}$$

Proof. Specializing Thm. 9 to the quantities outlined in Ex. 1, exploiting the bound on the norm of the approximated meta-subgradients given in Eq. (5.12) (exploiting Asm. 3) and using the notation in Eq. (5.34) and Eq. (5.2), the following bound holds for any $\theta \in \mathbb{R}^d$

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_{\bar\theta,\mu}(A_{\bar\theta}) \leq \lambda V_\rho(\theta) &+ \frac{1}{2\lambda nT}\,\mathbb{E}_{\mathbf{Z}}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|x_{t,i}s'_{\theta_t,i}\right\|_2^2 \\ &+ \frac{\eta\|\theta\|_2^2}{2T} + \frac{L^2\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,1}}{2\eta} + \frac{1}{2\lambda n}\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\bar\theta,i}\right\|_2^2.\end{aligned}$$

The desired statement derives from the above inequality and from observing that, thanks to Asm. 3 and the i.i.d. sampling of the data, using the inequality $\sum_{i=1}^n 1/i \leq \log(n)+1$ and the definition of C_ρ in Eq. (5.3), we have

$$\mathbb{E}_{\mathbf{Z}}\,\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|x_{t,i}s'_{\theta_t,i}\right\|_2^2 \leq L^2\mathbb{E}_{\mathbf{Z}}\,\mathrm{Tr}\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\,x_{t,i}x_{t,i}^\top\right) \leq L^2(\log(n)+1)\mathrm{Tr}(C_\rho)$$

$$\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\sum_{i=1}^{n}\frac{1}{i}\left\|x_i s'_{\bar\theta,i}\right\|_2^2 \leq L^2\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\,\mathrm{Tr}\left(\sum_{i=1}^{n}\frac{1}{i}\,x_i x_i^\top\right) \leq L^2(\log(n)+1)\mathrm{Tr}(C_\rho).$$

■

In order to evaluate the quality of the bound above, we specify Thm. 11 to Ex. 1, that is, we provide an (regularized) expected across-tasks excess risk bound for $\bar{w}_\theta$, the average of the iterations returned by running the within-task Alg. 4 with bias vector θ fixed in hindsight for any task.

Corollary 21 (Across-Tasks Excess Risk Bound for Alg. 4, Bias)**.** *Let Asm. 3 hold and consider the statistical setting in Thm. 11 applied to Ex. 1. In particular, for any $\theta \in \mathbb{R}^d$, let A_θ be the corresponding inner Alg. 4. Then, according to the notation in Eq. (5.34) and Eq. (5.3), the following (regularized) across-tasks excess-risk bound holds for any $\theta \in \mathbb{R}^d$*

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_{\theta,\mu}(A_\theta) \leq \lambda V_\rho(\theta) + \frac{L^2(\log(n)+1)\mathrm{Tr}(C_\rho)}{2\lambda n}. \tag{5.38}$$

Hence, optimizing w.r.t. the hyper-parameter λ, for

$$\lambda = L\sqrt{\frac{(\log(n)+1)\mathrm{Tr}(C_\rho)}{2nV_\rho(\theta)}}, \tag{5.39}$$

we get

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_{\theta,\mu}(A_\theta) \leq L\sqrt{\frac{2(\log(n)+1)V_\rho(\theta)\mathrm{Tr}(C_\rho)}{n}}. \tag{5.40}$$

Proof. Specializing Thm. 11 to the quantities outlined in Ex. 1, using the notation in Eq. (5.34), for any $\theta \in \mathbb{R}^d$, we get

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_{\theta,\mu}(A_\theta) \leq \lambda V_\rho(\theta) + \frac{1}{2\lambda n}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \sum_{i=1}^n \frac{1}{i} \left\| x_i s'_{\theta,i} \right\|_2^2.$$

The statement derives from the above inequality observing that, under Asm. 3, exploiting the i.i.d. sampling of the data and the inequality $\sum_{i=1}^n 1/i \leq \log(n)+1$, introducing the definition of C_ρ in Eq. (5.3), we can write

$$\mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \sum_{i=1}^n \frac{1}{i} \left\| x_i s'_{\theta,i} \right\|_2^2 \leq L^2 \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathrm{Tr}\Big(\sum_{i=1}^n \frac{1}{i}\, x_i x_i^\top \Big) \leq L^2(\log(n)+1)\, \mathrm{Tr}(C_\rho).$$

■

Also in this case, the comments to the bounds above are postponed in the following Sec. 5.5. In the section below, we specify to Ex. 1 the OWB variant of our Meta-Learning method and the corresponding analysis.

5.4 The Statistical Online-Within-Batch Variant

In this section, we consider the within-task batch RERM algorithm in Eq. (2.24) applied to the setting outlined in Ex. 1, i.e., for any $\theta \in \mathbb{R}^d$ and any dataset Z, we consider

$$\hat{w}_\theta = \operatorname*{argmin}_{w \in \mathbb{R}^d} \mathcal{R}_Z(w) + \frac{\lambda}{2} \|w - \theta\|_2^2. \tag{5.41}$$

In the following, we specify Thm. 14 to Ex. 1, that is, we provide an expected meta-excess risk bound for $\hat{w}_{\bar{\theta}}$, the RERM in Eq. (5.41) with bias vector $\bar{\theta}$, the average of the vectors returned by the meta-algorithm in Alg. 5 working with exact meta-subgradients computed by Eq. (5.13).

Corollary 22 (OWB Meta-Excess Risk Bound, Bias)**.** *Let Asm. 3 hold and consider the statistical setting in Thm. 14 applied to Ex. 1. In particular, let $A_{\bar{\theta}}$ be the inner RERM algorithm in Eq. (5.41) with bias $\bar{\theta}$, the average of the vectors returned by the meta-algorithm in Alg. 5 using the data $\mathbf{Z}$ and the exact meta-subgradients in Eq. (5.13). Then, according to the notation in Eq. (5.34), Eq. (5.2) and Eq. (5.3), the following expected meta-excess risk bound holds for any $\theta \in \mathbb{R}^d$*

$$\mathbb{E}_{\mathbf{Z}} \mathbb{E}_{\mu \sim \rho} \mathcal{E}_\mu(A_{\bar{\theta}}) \leq \lambda V_\rho(\theta) + \frac{L^2 \mathrm{Tr}(C_\rho)}{\lambda n} + \frac{\eta \|\theta\|_2^2}{2T} + \frac{L^2 \mathbb{E}_{\mathbf{Z}} \|C^{\mathrm{tot}}\|_{\infty,1}}{2\eta}.$$

Hence, optimizing w.r.t. the hyper-parameters λ and η, for

$$\lambda = L \sqrt{\frac{\mathrm{Tr}(C_\rho)}{n V_\rho(\theta)}} \qquad \eta = \frac{L \sqrt{T \, \mathbb{E}_{\mathbf{Z}} \|C^{\mathrm{tot}}\|_{\infty,1}}}{\|\theta\|_2}, \tag{5.42}$$

we get

$$\mathbb{E}_{\mathbf{Z}} \mathbb{E}_{\mu \sim \rho} \mathcal{E}_\mu(A_{\bar{\theta}}) \leq L \left(2 \sqrt{\frac{V_\rho(\theta) \, \mathrm{Tr}(C_\rho)}{n}} + \|\theta\|_2 \sqrt{\frac{\mathbb{E}_{\mathbf{Z}} \|C^{\mathrm{tot}}\|_{\infty,1}}{T}} \right).$$

Proof. Specializing Thm. 14 to the quantities outlined in Ex. 1, exploiting the bound on the norm of the exact meta-subgradients given in Eq. (5.12) (exploiting Asm. 3) and using the notation in Eq. (5.34) and Eq. (5.2), for any $\theta \in \mathbb{R}^d$, we get

$$\begin{aligned} \mathbb{E}_{\mathbf{Z}} \mathbb{E}_{\mu \sim \rho} \mathcal{E}_\mu(A_{\bar{\theta}}) \leq \lambda V_\rho(\theta) + \frac{2}{\lambda n} \mathbb{E}_{\mathbf{Z}} \mathbb{E}_{\mu \sim \rho} \mathbb{E}_{Z \sim \mu^n} \mathbb{E}_{z_i' \sim \mu} \left\| x_i' s_{\bar{\theta},i}' \right\|_2^2 \\ + \frac{\eta \|\theta\|_2^2}{2T} + \frac{L^2 \mathbb{E}_{\mathbf{Z}} \|C^{\mathrm{tot}}\|_{\infty,1}}{2\eta}. \end{aligned}$$

The statement derives from the above inequality observing that, under Asm. 3, exploiting the i.i.d. sampling of the data and introducing the definition of C_ρ in Eq. (5.3), we can write

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z'_i\sim\mu}\, \left\| x'_i s'_{\bar{\theta},i} \right\|_2^2 \leq L^2 \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{z'_i\sim\mu}\, \mathrm{Tr}(x'_i x_i'^{\top}) = L^2 \mathrm{Tr}(C_\rho). \tag{5.43}$$

■

In order to evaluate the quality of the bound above, we specify Thm. 15 to Ex. 1, that is, we provide an expected across-tasks excess risk bound for $\hat{w}_\theta$, the within-task RERM algorithm in Eq. (5.41) with bias vector θ fixed in hindsight for any task.

Corollary 23 (Across-Tasks Excess Risk Bound for the RERM Algorithm in Eq. (5.41), Bias)**.** *Let Asm. 3 hold and consider the statistical setting in Thm. 15 applied to Ex. 1. In particular, for any $\theta \in \mathbb{R}^d$, let A_θ be the corresponding inner RERM algorithm in Eq. (5.41). Then, according to the notation in Eq. (5.34) and Eq. (5.3), the following expected across-tasks excess risk bound holds for any $\theta \in \mathbb{R}^d$*

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta) \leq \lambda V_\rho(\theta) + \frac{2L^2 \mathrm{Tr}(C_\rho)}{\lambda n}. \tag{5.44}$$

Hence, optimizing w.r.t. the hyper-parameter λ, for

$$\lambda = L \sqrt{\frac{2\mathrm{Tr}(C_\rho)}{n V_\rho(\theta)}}, \tag{5.45}$$

we get

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta) \leq 2L \sqrt{\frac{2V_\rho(\theta)\, \mathrm{Tr}(C_\rho)}{n}}. \tag{5.46}$$

Proof. Specializing Thm. 15 to the quantities outlined in Ex. 1, using the notation in Eq. (5.34), for any $\theta \in \mathbb{R}^d$, we get

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta) \leq \lambda V_\rho(\theta) + \frac{2}{\lambda n}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z'_i\sim\mu}\, \left\| x'_i s'_{\theta,i} \right\|_2^2. \tag{5.47}$$

The statement derives from the above inequality observing that, under Asm. 3, introducing the definition of C_ρ in Eq. (5.3), we can write

$$\mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z'_i\sim\mu}\, \left\| x'_i s'_{\theta,i} \right\|_2^2 \leq L^2 \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{z'_i\sim\mu}\, \mathrm{Tr}(x'_i x_i'^{\top}) = L^2 \mathrm{Tr}(C_\rho). \tag{5.48}$$

■

In the following section we present a detailed discussion about the bounds we have reported in the above sections.

5.5 Discussion

We start from discussing the results in Cor. 19, Cor. 21 and Cor. 23, where the bias vector used by the inner algorithm is fixed in hindsight for any task.

Advantage of Selecting the Right Bias. Looking at the bounds in Cor. 19, Cor. 21 and Cor. 23, we can state that the advantage in using one bias vector $\theta \in \mathbb{R}^d$ in comparison to the others is determined by the associated empirical variance $\hat{V}(\theta)$ in Cor. 19 or by the corresponding exact variance $V_\rho(\theta)$ in Cor. 21 and Cor. 23. This inspires us to consider as the best algorithm in our class (*oracle*) the algorithm associated to the bias vector minimizing the above quantities:

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \hat{V}(\theta) = \frac{1}{T} \sum_{t=1}^{T} \hat{w}_t, \tag{5.49}$$

for the non-statistical setting in Cor. 19, and

$$\theta_\rho = \operatorname*{argmin}_{\theta \in \mathbb{R}^d} V_\rho(\theta) = \mathbb{E}_{\mu \sim \rho} w_\mu, \tag{5.50}$$

for the statistical setting in Cor. 21 and Cor. 23. For clarity of exposition, we remark that the above vectors do not coincide with the optimal meta-parameters defined in Sec. 2.2, but they are two sub-optimal choices in that they minimize just un upper bound of the quantity we would like to minimize. In the following, we will consider these two reasonable vectors as benchmark in order to evaluate the quality of the bias returned by our Meta-Learning procedures. For this reason, in order to avoid further notation, we over-write the symbols used in Sec. 2.2 for the true optimal meta-parameters, to denote these sub-optimal versions.

On the other hand, solving the tasks independently (ITL), in this case, corresponds to the unbiased case, i.e. to the application of the inner Alg. 4 with bias $\theta_{\text{ITL}} = 0 \in \mathbb{R}^d$ for any task. In particular, from the above bounds, we can say that there is an advantage in using the optimal bias w.r.t. solving each task independently, when the tasks are *similar* in the sense that the variance of the associated target vectors is much smaller than their second moment, i.e. when

$$\hat{V}(\hat{\theta}) = \min_{\theta \in \mathbb{R}^d} \frac{1}{2T} \sum_{t=1}^{T} \|\hat{w}_t - \theta\|_2^2 = \frac{1}{2T} \sum_{t=1}^{T} \|\hat{w}_t - \hat{\theta}\|_2^2 \ll \frac{1}{2T} \sum_{t=1}^{T} \|\hat{w}_t\|_2^2 = \hat{V}(0) \tag{5.51}$$

for the non-statistical setting and

$$V_\rho(\theta_\rho) = \min_{\theta \in \mathbb{R}^d} \mathbb{E}_{\mu \sim \rho} \frac{1}{2} \|w_\mu - \theta\|_2^2 = \mathbb{E}_{\mu \sim \rho} \frac{1}{2} \|w_\mu - \theta_\rho\|_2^2 \ll \mathbb{E}_{\mu \sim \rho} \frac{1}{2} \|w_\mu\|_2^2 = V_\rho(0) \quad (5.52)$$

for the statistical setting. Finally, we observe that, inline with the origin of our Meta-Learning method, the quantity $\hat{V}(\hat{\theta})$ above coincides with the MTL variance regularizer in Eq. (2.25) associated to the matrix in $\mathbb{R}^{d \times T}$ with columns the tasks' target vectors $(\hat{w}_t)_{t=1}^T$ and the quantity $V_\rho(\theta_\rho)$ can be interpreted as its continuous or exact statistical variant.

We now can make the following observations about the bounds we have obtained in Cor. 18, Cor. 20 and Cor. 22 for our Meta-Learning procedures.

Bias Resulting from our Meta-Learning Methods. Looking at the bounds in Cor. 18, Cor. 20 and Cor. 22, we can state that our Meta-Learning methods are effective, because, when the number of training tasks is sufficiently large w.r.t. the number of points n (hence the term $T^{-1/2}$ is negligible), with an appropriate tuning of the hyper-parameters λ and η, the bias vector estimated by our methods can provide comparable guarantees as those for the corresponding best bias vector in hindsight in Cor. 19, Cor. 21 and Cor. 23. As a consequence, when the tasks are similar as explained above, our methods can provide a significant advantage w.r.t. ITL. These observations are in line with (Denevi et al., 2019a), where we only consider the statistical setting and we present the same bound in Cor. 20 with slightly worse constants.

Finally, in order to investigate the impact of considering the OWO framework instead of the OWB one, we compare Cor. 20 and Cor. 21 to Cor. 22 and Cor. 23. We recall that all these statements hold for the statistical setting.

Comparison Between OWO and OWB. Comparing the bound in Cor. 21 to the bound in Cor. 23, we can expect that, in the statistical setting, running the batch RERM algorithm in Eq. (5.41) with a fixed bias θ in hindsight for any task will outperform the twin method with the online Alg. 4 by a factor $(\log(n) + 1)V_\rho(\theta)$. As a consequence, for a fixed value of n, the gap between the performance of the two methods will depend on the quantity $V_\rho(\theta)$ which can amplify or reduce the discrepancy. In particular, for the optimal choice $\theta = \theta_\rho$ in Eq. (5.50), the corresponding gap between the two methods can be insignificant when the variance $V_\rho(\theta_\rho)$ of the tasks' target vectors is small. The same observation can be made also for the Meta-Learning methods analyzed in Cor. 20 and Cor. 22. In this case, the bias controlling the above gap is the one determining the optimal hyper-parameters used by the methods.

5.6 Experiments

In this section, we test the effectiveness of the Meta-Learning approach proposed in this work on synthetic and real data in the statistical setting of Ex. 1[1].

We recall that, in the proposed framework, the within-task regularization parameter λ and the meta-step size η must be appropriately chosen. In the following experiments, we validated these two hyper-parameters over a grid of discrete values looking at the performance of the corresponding estimated meta-parameters, according to the procedure described in App. C.1. In order to do this, the proposed OWO Meta-Learning method uses the inner online Alg. 2 in two ways (i) to estimate the meta-subgradients during meta-training as described in Prop. 6 and (ii) to evaluate the meta-parameter during the meta-validation / testing phase. In the following experiments, we compared the proposed OWO Meta-Learning approach with variants based on the batch RERM algorithm in Eq. (2.24). More precisely, in the setting outlined in Ex. 1, we evaluated the performance of the following three methods:

- META - OWO: our Online-Within-Online Meta-Learning method described in Chpt. 3, where we use the inner online Alg. 4 both during meta-training and meta-validation / testing phases;
- META - Hybrid: the hybrid Meta-Learning method in which we use exact meta-subgradients (computed by the batch RERM in Eq. (5.41), as described in Prop. 12) during the meta-training phase, but we apply the online inner Alg. 4 during the meta-validation / testing phases;
- META - OWB: the Online-Within-Batch variant of our Meta-Learning method described in Chpt. 4, where we use the batch inner RERM algorithm in Eq. (5.41) both for the meta-training process (to compute the exact meta-subgradients) and the meta-validation / testing phases.

We also added to the comparison the following methods, where the bias vector θ is fixed in hindsight for any task:

- ITL - B: we use the batch RERM algorithm in Eq. (5.41) with the ITL bias $\theta_{\text{ITL}} = 0$ for any task;
- ITL - O: we use the online Alg. 4 with the ITL bias $\theta_{\text{ITL}} = 0$ for any task;

[1]The code is available at https://github.com/prolearner/onlineLTL

- Oracle - B: we use the batch RERM algorithm in Eq. (5.41) with the optimal bias θ_ρ in Eq. (5.50) for any task (only in synthetic experiments, in which this quantity is available);
- Oracle - O: we use the online Alg. 4 with the optimal bias θ_ρ in Eq. (5.50) for any task (only in synthetic experiments in which this quantity is available).

We compared the above methods in the following synthetic and real experimental settings, where, as described in App. C.2, we computed an approximation of the RERM algorithm in Eq. (5.41) by applying Fast Iterative Shrinkage-Thresholding Algorithm (FISTA, see (Beck and Teboulle, 2009, Sec. 4)) on the associated within-task dual problem. The following experimental settings are exactly the same considered in (Denevi et al., 2019a). The only difference is the choice of the inner step size for the inner Alg. 4: here the step size is $1/(\lambda(i+1))$, instead of $1/(\lambda i)$ as in (Denevi et al., 2019a). However, as the reader can observe, this difference results to be insignificant for the overall performance of the method.

Synthetic Data. We considered two different settings, regression with the absolute loss and binary classification with the hinge loss. In both cases, we generated an environment of tasks in which the introduction of an appropriate bias is expected to bring a substantial benefit in comparison to the unbiased case (ITL). Motivated by our observations in Sec. 5.5, we generated $T_{\text{tot}} = 800$ linear tasks with target vectors characterized by a variance which is significantly smaller than their second moment. Specifically, for each task μ, we created a target vector w_μ from a Gaussian distribution with mean θ_ρ given by the vector in $\mathbb{R}^d$ with all components equal to 4 and standard deviation $\sqrt{V_\rho(\theta_\rho)} = 1$. For each task we generated a dataset $(x_i, y_i)_{i=1}^{n_{\text{tot}}}$, where $x_i \in \mathbb{R}^d$ with $d = 30$ and $n_{\text{tot}} = 110$. In the regression case, the inputs were uniformly sampled on the unit sphere and the labels were generated as $y = \langle x, w_\mu \rangle + \epsilon$, with ϵ sampled from a zero-mean Gaussian distribution, with standard deviation chosen to have signal-to-noise ratio equal to 10 for each task. In the classification case, the inputs were uniformly sampled on the unit sphere, excluding those points with margin $|\langle x, w_\mu \rangle|$ smaller than 0.5 and the binary labels were generated according to a logistic model where $\mathbb{P}(y = 1) = \left(1 + 10 \exp(-\langle x, w_\mu \rangle)\right)^{-1}$. The inner regularization parameter λ and the meta-step size η were validated following the procedure described in App. C.1. Specifically, we considered 10 candidates values for both λ and η in the range $[10^{-6}, 10^3]$ with logarithmic spacing and we evaluated the performance of the estimated bias vectors by using $T = T_{\text{tr}} = 500$, $T_{\text{va}} = 100$, $T_{\text{te}} = 200$ of the above tasks for meta-training, meta-validation and meta-testing, respectively. Moreover, in order to train and to test the inner algorithm, we used $n = n_{\text{tr}} = 10$ and $n_{\text{te}} = 100$ points in each dataset.

The results, reported in Fig. 5.1, confirm our theoretical findings. First of all, we observe that all the Meta-Learning methods (META - OWO, META - Hybrid, META - OWB) perform better

than solving each task independently (ITL - B and ITL - O) by a large margin, and, as expected, they tend to match the performance of the optimal algorithms in the class (Oracle - B and Oracle - O) as the number of training tasks increases. We also point out that the OWB Meta-Learning method (META - OWB) achieves lower error than the other two Meta-Learning methods (META - OWO and META - Hybrid), but the difference is almost negligible and this is coherent with the results obtained in Chpt. 4. In addition to this, we also observe that the performance of META - OWO and META - Hybrid are comparable. This confirms, as already observed from the theory, that the way in which we approximated the meta-subgradients is an effective way to keep the process fully online. Finally, we observe that the gap between the lines ITL - O and ITL - B is wider than the gap between the lines Oracle - O and Oracle - B. This is inline with our theoretical findings according to which running the batch RERM algorithm in Eq. (5.41) with a fixed bias outperforms the twin method with the online Alg. 4 by a factor $(\log(n) + 1)V_\rho(\theta)$. Since in our setting $V_\rho(\theta) \ll V_\rho(0)$, this explains the different gaps between the horizontal lines above.

Real Data. We ran experiments on the computer survey data from (Lenk et al., 1996), in which $T_{\text{tot}} = 180$ people (tasks) rated the likelihood of purchasing one of $n_{\text{tot}} = 12$ different personal computers. The input represents $d = 13$ different computer characteristics (price, CPU, RAM, etc.), while, the output is an integer rating from 0 to 10. Similarly to the synthetic data experiments, we considered a regression setting with the absolute loss and a classification setting with the hinge loss. In the latter case each task is to predict whether the rating is above 5. Also in this case, in order to validate the hyper-parameters λ and η, we followed the procedure described in App. C.1. Specifically, we considered 30 candidates values for both λ and η in the range $[10^{-3}, 10^3]$ with logarithmic spacing and we evaluated the performance of the estimated bias vectors by splitting the tasks into $T = T_{\text{tr}} = 100$, $T_{\text{va}} = 40$, $T_{\text{te}} = 40$ tasks for meta-training, meta-validation and meta-testing, respectively. Moreover, in order to train and to test the inner algorithm, we splitted each within-task dataset into $n = n_{\text{tr}} = 8$ and $n_{\text{te}} = 4$ points.

We compared all the methods we described above over this dataset. The results are reported in Fig. 5.2. The figures are in line with the results obtained on synthetic experiments, indicating that the bias Meta-Learning framework proposed in this work is effective for these data. Furthermore, the results for regression are in line with what observed for the Multi-Task Learning setting with variance regularization in (McDonald et al., 2016). The classification setting has not been used before and has been created ad-hoc for our purpose. In this case, we have a wider variance probably due to the fact that the splitted datasets are highly unbalanced.

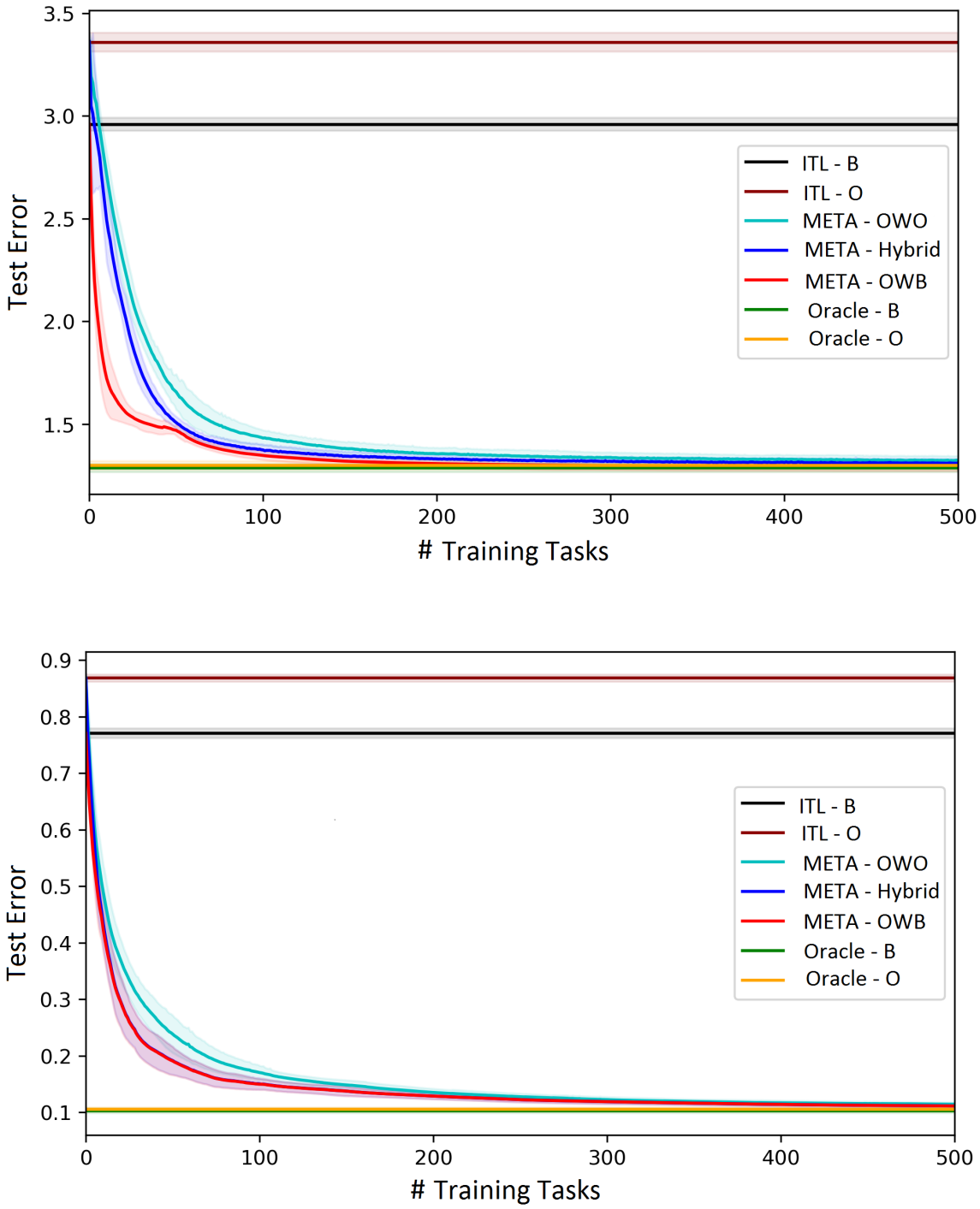

Figure 5.1 **Synthetic Data.** Test error of different methods as the number of training tasks increases. **(Top)** Regression with absolute loss. **(Bottom)** Classification with hinge loss. The results are averaged over 10 independent generations of the data.

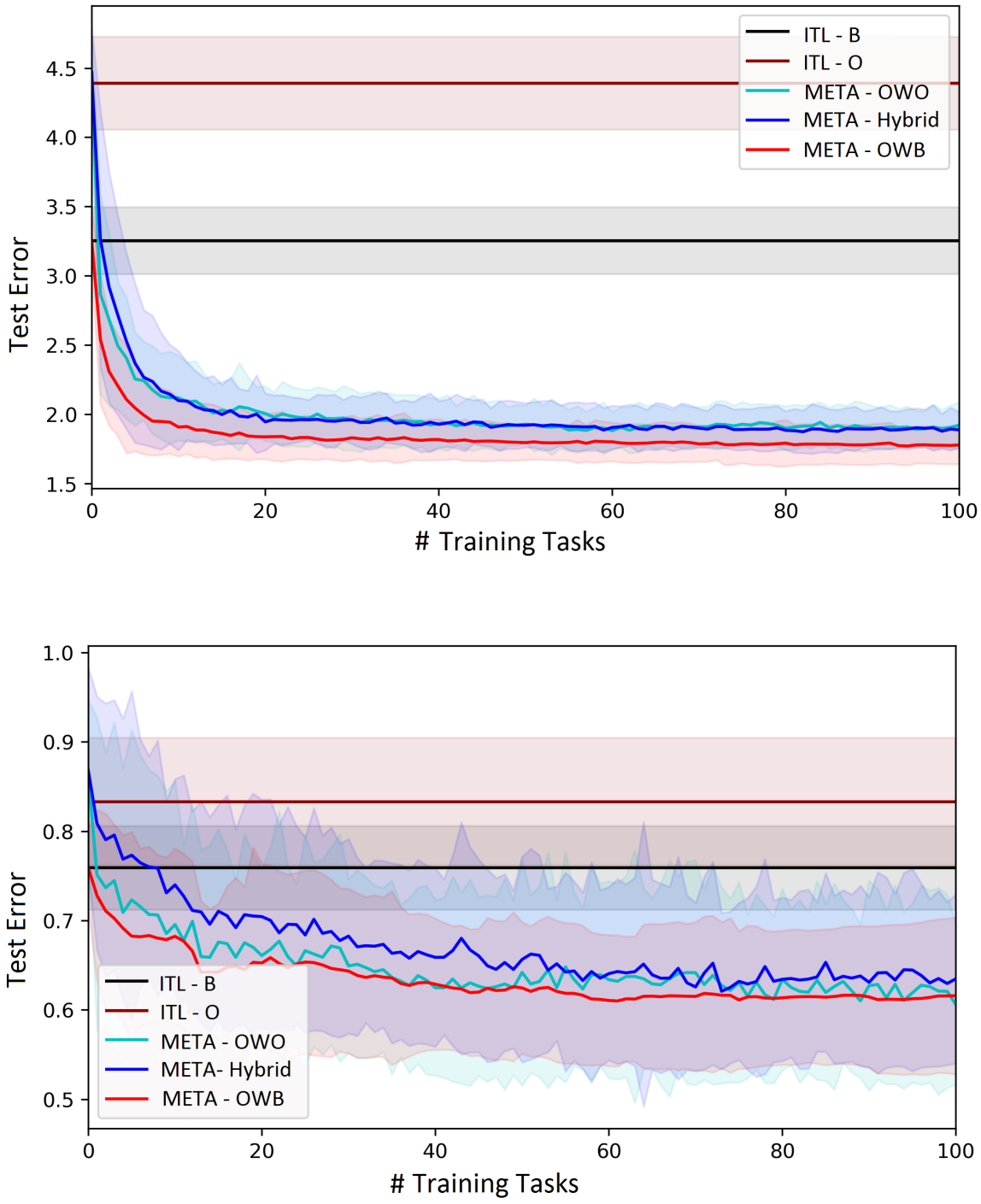

Figure 5.2 **Lenk Dataset.** Test error of different methods as the number of training tasks increases. **(Top)** Regression with absolute loss. **(Bottom)** Classification with hinge loss. The results are averaged over 30 independent splits of the data.

Chapter 6

Example 2. Feature Map

In this chapter we specify our Meta-Learning framework to the setting in Ex. 2 outlined at the end of Sec. 3.1. We recall that, in such a case, the meta-parameter coincides with a linear feature map $\theta \in \mathbb{S}_+^d$ and, as we will see in the following, the tasks' similarity translates into the existence of a low-rank linear feature map containing in its range the tasks' target vectors. We start this chapter by specializing in Sec. 6.1 our general OWO method described in Chpt. 3 to Ex. 2, deriving the corresponding inner and meta-algorithm. The method is then analyzed in Sec. 6.2 and Sec. 6.3, where we specify the meta-regret bound in Thm. 7 and the meta-excess risk bound in Thm. 9, respectively. After this, in the subsequent Sec. 6.4, we describe the OWB variant of our method introduced in Chpt. 4. Finally in Sec. 6.5 and Sec. 6.6, we discuss the results and we report the numerical evaluation of our method, respectively.

In this chapter we will use Asm. 3 and the notation in Eq. (5.1), Eq. (5.2) and Eq. (5.3) introduced in the former Chpt. 5.

6.1 Deriving the Method

We start from specializing the generic inner algorithm in Alg. 2 and the generic meta-algorithm in Alg. 3 to the setting outlined in Ex. 2. The algorithms we obtain are reported in Alg. 6 and Alg. 7, respectively, where, $\text{proj}_{\mathcal{S}}$ is the Euclidean projection over the set $\mathcal{S}$ and we recall that $X_t \in \mathbb{R}^{n \times d}$ denotes the input vectors' matrix of the task t, having as i–th row the input vector $x_{t,i}$. The deduction is reported in Lemma 24 and Lemma 25 below, respectively.

Algorithm 6 Within-Task Algorithm for Ex. 2

Input $\lambda > 0, \theta \in \mathcal{S}, Z = (z_i)_{i=1}^n$

Initialization $s_{\theta,1} = (), w_{\theta,1} = 0$

For $i = 1$ to n

Receive the datapoint $z_i = (x_i, y_i)$

Compute $s'_{\theta,i} \in \partial \ell_i(\langle x_i, w_{\theta,i} \rangle) \subseteq \mathbb{R}$

Define $(s_{\theta,i+1})_i = s'_{\theta,i}, \gamma_i = \lambda(i+1)$

Define $p_{\theta,i} = x_i s'_{\theta,i} + \lambda \theta^\dagger w_{\theta,i}$

Update $w_{\theta,i+1} = w_{\theta,i} - 1/\gamma_i \, \theta p_{\theta,i}$

Return $(w_{\theta,i})_{i=1}^{n+1}, \ \bar{w}_\theta = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} w_{\theta,i}, \ s_{\theta,n+1}$

Algorithm 7 Meta-Algorithm for Ex. 2

Input $\eta > 0, \mathbf{Z} = (Z_t)_{t=1}^T, \theta_0 \in \mathcal{S}$

Initialization $\theta_1 = \theta_0, P_1 = 0 \in \mathbb{S}^d$

For $t = 1$ to T

Receive incrementally the dataset Z_t

Run Alg. 6 with θ_t over Z_t

Compute $s_{\theta_t,n+1}$

Define $\nabla'_{\theta_t} = -\dfrac{q_t q_t^\top}{2\lambda n^2} \quad q_t = {X_t}^\top s_{\theta_t,n+1}$

Update $P_{t+1} = P_t + \nabla'_{\theta_t}$

Update $\theta_{t+1} = \mathrm{proj}_{\mathcal{S}}\Big(-\mathrm{P}_{\mathrm{t+1}}/\eta + \theta_0\Big)$

Return $(\theta_t)_{t=1}^{T+1}, \bar{\theta} = \dfrac{1}{T} \displaystyle\sum_{t=1}^{T} \theta_t$

We start from the deduction of the inner-algorithm in Alg. 6.

Lemma 24 (Derivation of the Inner Alg. 6, Feature Map)**.** *For any $i \in \{0, \dots, n\}$, let $w_{\theta,i+1}$ be the update of the (primal) variable deriving from applying Alg. 2 to the dataset $Z = (x_i, y_i)_{i=1}^n$ in the setting outlined in Ex. 2 with feature map $\theta \in \mathcal{S}$. Let $s'_{\theta,i} \in \partial \ell_i(\langle x_i, w_{\theta,i} \rangle)$ be the subgradient used by such an algorithm to compute $w_{\theta,i+1}$. Then, $w_{\theta,i+1} \in \mathrm{Ran}(\theta)$. Moreover, $w_{\theta,1} = 0 \in \mathbb{R}^d$ and, for any $i \in \{1, \dots, n\}$, introducing the subgradient of the regularized loss*

$$p_{\theta,i} = x_i s'_{\theta,i} + \lambda \theta^\dagger w_{\theta,i} \in \partial\Big(\ell_i(\langle x_i, \cdot \rangle) + \frac{\lambda}{2} \langle \cdot, \theta^\dagger \cdot \rangle\Big)(w_{\theta,i}), \tag{6.1}$$

we have

$$w_{\theta,i+1} = w_{\theta,i} - \frac{1}{\lambda(i+1)} \Big(\theta x_i s'_{\theta,i} + \lambda w_{\theta,i}\Big) = w_{\theta,i} - \frac{1}{\lambda(i+1)} \theta p_{\theta,i}. \tag{6.2}$$

Proof. We start from observing that, according to the choices made in Ex. 2, for any $\theta \in \mathcal{S}$ and for any $w, u \in \mathbb{R}^d$, we have

$$f(w, \theta) = \frac{1}{2} \langle w, \theta^\dagger w \rangle + \iota_{\mathrm{Ran}(\theta)}(w) \quad f(\cdot, \theta)^*(u) = \frac{1}{2} \|\theta^{1/2} u\|_2^2 \quad \nabla f(\cdot, \theta)^*(u) = \theta u. \tag{6.3}$$

As a consequence, as observed in Prop. 3, for any $\theta \in \Theta$, we get that $w_{\theta,i+1} \in \mathrm{Dom} f(\cdot,\theta) = \mathrm{Ran}(\theta)$, for any $i \in \{0,\dots,n\}$. Moreover, according to the definition of $w_{\theta,1}$ in Alg. 2, we have

$$w_{\theta,1} = \nabla f(\cdot,\theta)^*(0) = 0. \tag{6.4}$$

We now show the closed form of $w_{\theta,i+1}$ for any $i \in \{1,\dots,n\}$. In such a case, denoting by $X_{1:i} \in \mathbb{R}^{i\times d}$ the matrix containing the first i input vectors as rows, by definition of $w_{\theta,i+1}$ in Alg. 2, we can write

$$w_{\theta,i+1} = \nabla f(\cdot,\theta)^*\Big(-\frac{1}{\lambda(i+1)}\, X_{1:i}^\top s_{\theta,i+1}\Big) = -\frac{1}{\lambda(i+1)}\,\theta X_{1:i}^\top s_{\theta,i+1}. \tag{6.5}$$

For $i = 1$ the statement holds, as a matter of fact, since $w_{\theta,1} = 0$, exploiting Eq. (6.5) and introducing the subgradient $p_{\theta,1} = x_1 s'_{\theta,1} + \lambda\theta^\dagger w_{\theta,1} = x_1 s'_{\theta,1}$, we can write

$$w_{\theta,2} = -\frac{1}{2\lambda}\,\theta x_1 s'_{\theta,1} = w_{\theta,1} - \frac{1}{2\lambda}\,\theta p_{\theta,1}. \tag{6.6}$$

Now, we show that the statement holds also for $i \in \{2,\dots,n\}$. Since $X_{1:i}^\top s_{\theta,i+1} = X_{1:i-1}^\top s_{\theta,i} + x_i s'_{\theta,i}$, we can write the following

$$\begin{aligned} w_{\theta,i+1} &= -\frac{1}{\lambda(i+1)}\,\theta X_{1:i}^\top s_{\theta,i+1} = -\frac{1}{\lambda(i+1)}\Big(\theta X_{1:i-1}^\top s_{\theta,i} + \theta x_i s'_{\theta,i}\Big) \\ &= \frac{\lambda i}{\lambda(i+1)}\Big(-\frac{1}{\lambda i}\,\theta X_{1:i-1}^\top s_{\theta,i}\Big) - \frac{\theta x_i s'_{\theta,i}}{\lambda(i+1)} \\ &= \frac{\lambda(i+1)w_{\theta,i} - \theta x_i s'_{\theta,i} - \lambda w_{\theta,i}}{\lambda(i+1)} \\ &= w_{\theta,i} - \frac{1}{\lambda(i+1)}\Big(\theta x_i s'_{\theta,i} + \lambda w_{\theta,i}\Big) = w_{\theta,i} - \frac{1}{\lambda(i+1)}\,\theta p_{\theta,i}, \end{aligned} \tag{6.7}$$

where, in the first and the fourth equality, we have exploited Eq. (6.5) and in the sixth equality we have exploited the form of the subgradient $p_{\theta,i} = x_i s'_{\theta,i} + \lambda\theta^\dagger w_{\theta,i}$ and the fact that $w_{\theta,i} \in \mathrm{Ran}(\theta)$. ∎

We now proceed with the deduction of the meta-algorithm in Alg. 7.

Lemma 25 (Derivation of the Meta-Algorithm in Alg. 7, Feature Map)**.** *For any $t \in \{0,\dots,T\}$, let θ_{t+1} be the update of the variable deriving from applying Alg. 3 to the data $\mathbf{Z} = (Z_t)_{t=1}^T$ in the setting outlined in Ex. 2. Let ∇'_{θ_t} be the approximated meta-subgradient computed as described in Prop. 6 and used by the algorithm to compute θ_{t+1}. Then, $\theta_{t+1} \in \mathcal{S}$. Specifically, we have*

$\theta_1 = \theta_0$ *and, for any* $t \in \{1, \dots, T\}$,

$$\theta_{t+1} = \mathrm{proj}_{\mathcal{S}}\Big(-\frac{1}{\eta}\sum_{\mathrm{j}=1}^{\mathrm{t}} \nabla'_{\theta_{\mathrm{j}}} + \theta_0\Big). \tag{6.8}$$

Moreover, for any $t \in \{1, \dots, T\}$,

$$\nabla'_{\theta_t} = -\frac{1}{2\lambda n^2}\, {X_t}^\top s_{\theta_t, n+1} s_{\theta_t, n+1}^\top X_t, \tag{6.9}$$

where $s_{\theta_t, n+1} \in \mathbb{R}^n$ *is the output of Alg. 7 with feature map* θ_t *over the dataset* Z_t *and, under Asm. 3, according to the notation in Eq. (5.1),*

$$\left\|\nabla'_{\theta_t}\right\|_F^2 \leq \frac{L^4 \|C_t\|_\infty^2}{4\lambda^2}. \tag{6.10}$$

Finally, the updating step and the bound above hold also for the exact meta-subgradients computed as described in Prop. 12, which are given, for any $t \in \{1, \dots T\}$, *by*

$$\nabla_{\theta_t} = -\frac{1}{2\lambda n^2}\, {X_t}^\top \hat{s}_{\theta_t} \hat{s}_{\theta_t}^\top X_t = -\frac{\lambda}{2} \theta_t^\dagger \hat{w}_{\theta_t} \hat{w}_{\theta_t}^\top \theta_t^\dagger, \tag{6.11}$$

where $\hat{w}_{\theta_t}$ *and* $\hat{s}_{\theta_t}$ *denote, respectively, the RERM algorithm in Eq. (2.24) and a solution of the associated dual problem with meta-parameter* θ_t *and dataset* Z_t *for the setting in Ex. 2.*

Proof. We start from observing that, according to the choices made in Ex. 2, according to Lemma 57 in App. A, for any $K \in \mathbb{S}^d$, $\theta \in \mathcal{S}$ and $u \in \mathbb{R}^d$, we have

$$\begin{aligned}
F(\theta) &= \frac{1}{2}\|\theta - \theta_0\|_F^2 + \iota_{\mathcal{S}}(\theta) \\
F^*(K) &= \max_{\theta \in \mathcal{S}} \langle \theta, K \rangle - \frac{1}{2}\, \|\theta - \theta_0\|_F^2 \\
\nabla F^*(K) &= \operatorname*{argmax}_{\theta \in \mathcal{S}} \langle \theta, K \rangle - \frac{1}{2}\, \|\theta - \theta_0\|_F^2 = \operatorname*{argmin}_{\theta \in \mathcal{S}} \frac{1}{2}\, \|\theta - \theta_0\|_F^2 - \langle \theta, K \rangle \\
&= \operatorname*{argmin}_{\theta \in \mathcal{S}} \frac{1}{2}\, \|\theta - (K + \theta_0)\|_F^2 - \frac{1}{2}\, \|K\|_F^2 + \langle \theta - \theta_0, K \rangle - \langle \theta, K \rangle \\
&= \operatorname*{argmin}_{\theta \in \mathcal{S}} \frac{1}{2}\, \|\theta - (K + \theta_0)\|_F^2 - \frac{1}{2}\, \|K\|_F^2 - \langle \theta_0, K \rangle \\
&= \mathrm{proj}_{\mathcal{S}}(\mathrm{K} + \theta_0) \\
f(\cdot, \theta)^*(u) &= \frac{1}{2}\, \|\theta^{1/2} u\|_2^2.
\end{aligned} \tag{6.12}$$

Consequently, according to the definition of θ_1 in Alg. 3, we have

$$\theta_1 = \nabla F^*(0) = \theta_0. \tag{6.13}$$

The desired closed form of θ_{t+1} for any $t \in \{1, \dots, T\}$ directly derives from the definition of θ_{t+1} in Alg. 3, according to which

$$\theta_{t+1} = \nabla F^*\Big(-\frac{1}{\eta}\sum_{j=1}^{t} \nabla'_{\theta_j}\Big) = \mathrm{proj}_{\mathcal{S}}\Big(-\frac{1}{\eta}\sum_{\mathrm{j}=1}^{\mathrm{t}} \nabla'_{\theta_{\mathrm{j}}} + \theta_0\Big). \tag{6.14}$$

Obviously, the above steps hold also when we substitute the approximated meta-subgradients $(\nabla'_{\theta_t})_{t=1}^T$ with the exact counterparts $(\hat{\nabla}'_{\theta_t})_{t=1}^T$. We now specify the closed form of the approximated meta-subgradients, computed as described in Prop. 6 for Ex. 2. We start from observing that adding to the notation in Prop. 6 the further task index t, by strong duality (see Lemma 5), we can rewrite

$$\mathcal{L}_t(\theta) = \max_{s \in \mathbb{R}^n} \tilde{D}_{t,n+1}(s, \theta) \qquad \tilde{D}_{t,n+1}(s, \theta) = -\frac{1}{n} D_{t,n+1}(s, \theta) \tag{6.15}$$

where, according to Eq. (3.10), in the setting outlined in Ex. 2,

$$\begin{aligned} -D_{t,n+1}(s, \theta) &= -\sum_{i=1}^{n} \ell^*_{t,i}(s_i) - \lambda n f(\cdot, \theta)^*\Big(-\frac{1}{\lambda n}\sum_{i=1}^{n} x_{t,i} s_i\Big) \\ &= -\sum_{i=1}^{n} \ell^*_{t,i}(s_i) - \lambda n f(\cdot, \theta)^*\Big(-\frac{1}{\lambda n} {X_t}^\top s\Big) \\ &= -\sum_{i=1}^{n} \ell^*_{t,i}(s_i) - \frac{1}{2\lambda n} s^\top X_t \theta {X_t}^\top s. \end{aligned} \tag{6.16}$$

Consequently, recalling that the output $s_{\theta_t,n+1}$ of the inner algorithm coincides with the last iterate of the corresponding dual inner iteration, according to Prop. 6, we have

$$\nabla_{\theta_t} = -\frac{1}{2\lambda n} {X_t}^\top s_{\theta_t,n+1} s_{\theta_t,n+1}^\top X_t \tag{6.17}$$

and, consequently,

$$\nabla'_{\theta_t} = \nabla_{\theta_t}/n \in \partial_{\epsilon_{\theta_t}/n} \mathcal{L}_t(\theta_t), \tag{6.18}$$

where ϵ_{θ_t} is outlined in Prop. 6 and it must be specified to Ex. 2. In order to prove Eq. (6.10), we start from observing that $s_{\theta_t,n+1}$ is the vector in $\mathbb{R}^n$ having as component i the subgradient $s'_{\theta_t,i} \in \partial \ell_{t,i}(\langle x_{t,i}, w_{\theta_t,i} \rangle)$. Hence, under Asm. 3, by Lemma 50 in App. A, any component of $s_{\theta_t,n+1}$ is absolutely bounded by L, and, consequently, $\|s_{\theta_t,n+1}\|_2 \leq L\sqrt{n}$. This allows us to get the desired bound by applying Holder's inequality (see Lemma 32 in App. A) to the matrices'

scalar product as follows

$$\left\|\nabla'_{\theta_t}\right\|_F = \frac{1}{2\lambda n}\,\mathrm{Tr}\Big(\frac{1}{n}\sum_{i=1}^{n} x_{t,i}x_{t,i}^\top s_{\theta_t,n+1}s_{\theta_t,n+1}^\top\Big) \leq \frac{1}{2\lambda n}\left\|\frac{1}{n}\sum_{i=1}^{n} x_{t,i}x_{t,i}^\top\right\|_\infty \|s_{\theta_t,n+1}\|_2^2$$
$$\leq \frac{L^2}{2\lambda}\left\|\frac{1}{n}\sum_{i=1}^{n} x_{t,i}x_{t,i}^\top\right\|_\infty = \frac{L^2\|C_t\|_\infty}{2\lambda},$$

where in the last equality we have introduced the definition of C_t in Eq. (5.1). Finally, regarding the exact meta-subgradients, the first closed form in Eq. (6.11) directly derives from Prop. 12 and the former discussion, while the second closed form is deduced by the first KKT condition in Eq. (3.17) specified to Ex. 2 and to the task t:

$$\hat{w}_{\theta_t} = -\frac{1}{\lambda n}\,\theta_t X_t^\top \hat{s}_{\theta_t} \in \mathrm{Ran}(\theta_t) \qquad \hat{s}_\theta \in \partial\Big(\sum_{i=1}^{n}\ell_i\Big)\Big(\langle x_1,\hat{w}_\theta\rangle,\ldots,\langle x_n,\hat{w}_\theta\rangle\Big). \tag{6.19}$$

Moreover, thanks to the second KKT condition above, under Asm. 3, by Lemma 43 and Lemma 50 in App. A, any component of $\hat{s}_{\theta_t}$ is absolutely bounded by L. Consequently, repeating the same steps above, the bound in Eq. (6.10) holds also for the exact meta-subgradients in Eq. (6.11). ■

We observe that the meta-algorithm we have retrieved in Alg. 7 is a slightly different version of that one proposed in (Denevi et al., 2018a), where we consider only an OWB statistical Meta-Learning framework. We refer to the discussion in Sec. 4.2 for more details about that work.

Also in this case, in Lemma 25, we provide the same upper bound for the norm of the exact meta-subgradients and the approximated ones. As a consequence, the error we introduce with such an approximation will not affect the final bound for our OWO method. Again, this theoretical suggestion will be confirmed by our experiments.

We observe that for the setting in Ex. 2, our Meta-Learning method in Alg. 6 and Alg. 7 requires to compute the eigenvalue decomposition of a rank one perturbation of the current matrix. Rank-one updates can be performed using methods such as the ones described in (Stange, 2008), which essentially scale quadratically with respect to the input dimension, instead of the standard cubic rate. As done in (Bullins et al., 2019), a cheaper alternative here may be to use as meta-algorithm Frank-Wolfe, which requires to compute only the maximum eigenvalue. However, the better scaling property of this method comes at the price of a slower learning and convergence rate.

In the next section, we analyze the performance of our OWO Meta-Learning method applied to Ex. 2, in the non-statistical setting.

6.2 Method and Analysis in the Non-Statistical Setting

In the next result we specify Thm. 7 to Ex. 2, that is we provide a (regularized) average meta-regret bound for the procedure deriving from combining Alg. 6 with Alg. 7.

Corollary 26 (Across-Tasks Regret Bound, Feature Map)**.** *Let Asm. 3 hold and consider the setting in Thm. 7 applied to Ex. 2. In particular, for any $\theta \in \mathcal{S}$, let A_θ be the corresponding inner Alg. 6 and let $(\theta_t)_{t=1}^T$ be the sequence of the feature maps estimated by the meta-algorithm in Alg. 7 over the data $\mathbf{Z} = (Z_t)_{t=1}^T$. Recall also the minimum norm empirical risk minimizers $(\hat{w}_t)_{t=1}^T$ associated to the datasets $(Z_t)_{t=1}^T$. Then, introducing the empirical covariance matrix of the vectors $(\hat{w}_t)_{t=1}^T$*

$$\hat{B} = \frac{1}{T}\sum_{t=1}^{T} \hat{w}_t \hat{w}_t^\top, \tag{6.20}$$

according to the notation in Eq. (5.2), the following (regularized) average meta-regret bound holds for any $\theta \in \mathcal{S}$ such that $\mathrm{Ran}(\hat{B}) \subseteq \mathrm{Ran}(\theta)$*,*

$$\frac{1}{T}\sum_{t=1}^{T} \mathcal{E}_{\theta_t, Z_t}(A_{\theta_t}) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger \hat{B})}{2} + \frac{L^2 \mathrm{Tr}(\hat{C}^{\mathrm{tot}}_{\theta_{1:T}})}{2\lambda n} + \frac{\eta \|\theta - \theta_0\|_F^2}{2T} + \frac{L^4 \|C^{\mathrm{tot}}\|_{\infty,2}}{8\lambda^2 \eta}, \tag{6.21}$$

where, according to the notation in Eq. (5.1), we have defined the matrix

$$\hat{C}^{\mathrm{tot}}_{\theta_{1:T}} = \frac{1}{T}\sum_{t=1}^{T} \theta_t \hat{C}_t. \tag{6.22}$$

Hence, optimizing w.r.t. the hyper-parameters λ and η, for

$$\lambda = L\sqrt{\frac{1}{\mathrm{Tr}(\theta^\dagger \hat{B})}\left(\frac{\mathrm{Tr}(\hat{C}^{\mathrm{tot}}_{\theta_{1:T}})}{n} + \|\theta - \theta_0\|_F \sqrt{\frac{\|C^{\mathrm{tot}}\|_{\infty,2}}{T}}\right)} \qquad \eta = \frac{L^2\sqrt{T\,\|C^{\mathrm{tot}}\|_{\infty,2}}}{2\lambda \|\theta - \theta_0\|_F}, \tag{6.23}$$

we get

$$\frac{1}{T}\sum_{t=1}^{T} \mathcal{E}_{\theta_t, Z_t}(A_{\theta_t}) \leq L\sqrt{\mathrm{Tr}(\theta^\dagger \hat{B})\left(\frac{\mathrm{Tr}(\hat{C}^{\mathrm{tot}}_{\theta_{1:T}})}{n} + \|\theta - \theta_0\|_F \sqrt{\frac{\|C^{\mathrm{tot}}\|_{\infty,2}}{T}}\right)}. \tag{6.24}$$

Proof. Specializing Thm. 7 to the quantities outlined in Ex. 2, exploiting the bound on the norm of the approximated meta-subgradients given in Eq. (6.10) (exploiting Asm. 3) and using the

notation in Eq. (6.20) and Eq. (5.2), for any $\theta \in \mathcal{S}$ such that $\mathrm{Ran}(\hat{B}) \subseteq \mathrm{Ran}(\theta)$, we get

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{\theta_t,Z_t}(A_{\theta_t}) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger \hat{B})}{2} + \frac{1}{2\lambda n T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|{\theta_t}^{1/2} x_{t,i} s'_{\theta_t,i}\right\|_2^2 + \frac{\eta\|\theta-\theta_0\|_F^2}{2T} + \frac{L^4\|C^{\mathrm{tot}}\|_{\infty,2}}{8\lambda^2\eta}. \tag{6.25}$$

The statement derives from the above inequality observing that, under Asm. 3 using the definition of $\hat{C}^{\mathrm{tot}}_{\theta_{1:T}}$ in Eq. (6.22), we can write

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|{\theta_t}^{1/2} x_{t,i} s'_{t,i}\right\|_2^2 \leq L^2 \mathrm{Tr}\left(\frac{1}{T}\sum_{t=1}^{T}\theta_t \sum_{i=1}^{n}\frac{1}{i}x_{t,i}x_{t,i}^\top\right) = L^2\mathrm{Tr}(\hat{C}^{\mathrm{tot}}_{\theta_{1:T}}). \tag{6.26}$$

■

Also in this case, in order to evaluate the quality of the bound above, we specify Thm. 8 to Ex. 2, that is, we provide a (regularized) average across-tasks regret bound for the procedure deriving from running the within-task Alg. 6 with a feature map fixed in hindsight for any task.

Corollary 27 (Across-Tasks Regret Bound for Alg. 6, Feature Map)**.** *Let Asm. 3 hold and consider the setting in Thm. 8 applied to Ex. 2. In particular, for any $\theta \in \mathcal{S}$, let A_θ be the corresponding inner Alg. 6. Then, according to the notation in Eq. (6.20) and Eq. (5.1), the following (regularized) average across-tasks regret bound holds for any $\theta \in \mathcal{S}$ such that* $\mathrm{Ran}(\hat{B}) \subseteq \mathrm{Ran}(\theta)$

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{\theta,Z_t}(A_\theta) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger \hat{B})}{2} + \frac{L^2\mathrm{Tr}(\theta \hat{C}^{\mathrm{tot}})}{2\lambda n}. \tag{6.27}$$

Hence, optimizing w.r.t. the hyper-parameter λ, for

$$\lambda = L\sqrt{\frac{\mathrm{Tr}(\theta \hat{C}^{\mathrm{tot}})}{n\mathrm{Tr}(\theta^\dagger \hat{B})}}, \tag{6.28}$$

we get

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{\theta,Z_t}(A_\theta) \leq L\sqrt{\frac{\mathrm{Tr}(\theta^\dagger \hat{B})\ \mathrm{Tr}(\theta \hat{C}^{\mathrm{tot}})}{n}}. \tag{6.29}$$

Proof. Specializing Thm. 8 to the quantities outlined in Ex. 2, using the notation in Eq. (6.20), for any $\theta \in \mathcal{S}$ such that $\mathrm{Ran}(\hat{B}) \subseteq \mathrm{Ran}(\theta)$, we get

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}_{\theta,Z_t}(A_\theta) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger \hat{B})}{2} + \frac{1}{2\lambda nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|\theta^{1/2}x_{t,i}s'_{t,i}\right\|_2^2.$$

The statement derives from the above inequality observing that, under Asm. 3 using the definition of the matrix $\hat{C}^{\mathrm{tot}}$ in Eq. (5.1), we can write

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|\theta^{1/2}x_{t,i}s'_{t,i}\right\|_2^2 \leq L^2\mathrm{Tr}\Big(\theta\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\,x_{t,i}x_{t,i}^\top\Big) = L^2\mathrm{Tr}(\theta\hat{C}^{\mathrm{tot}}). \tag{6.30}$$

■

We postpone to Sec. 6.5 a discussion about the results we have reported above. In the next section, we analyze the performance of our OWO Meta-Learning method applied to Ex. 2, in the statistical setting.

6.3 Method and Analysis in the Statistical Setting

In the result below we specify Thm. 9 to Ex. 2, that is, we provide a (regularized) expected meta-excess risk bound for the average $\bar{w}_{\bar{\theta}}$ of the estimators returned by the combination of Alg. 6 with Alg. 7.

Corollary 28 (OWO Meta-Excess Risk Bound, Feature Map)**.** *Let Asm. 3 hold and consider the statistical setting in Thm. 9 applied to Ex. 2. In particular, let $A_{\bar{\theta}}$ be the inner Alg. 6 with feature map $\bar{\theta}$, the average of the feature maps returned by the meta-algorithm in Alg. 7 using the data* $\mathbf{Z}$*. Recall also the minimum norm risk minimizer w_μ associated to a task $\mu \sim \rho$. Then, introducing the exact covariance matrix of the vectors w_μ*

$$B_\rho = \mathbb{E}_{\mu\sim\rho} w_\mu w_\mu^\top, \tag{6.31}$$

according to the notation in Eq. (5.2) and Eq. (5.3), the following (regularized) expected meta-excess risk bound holds for any $\theta \in \mathcal{S}$ such that $\mathrm{Ran}(B_\rho) \subseteq \mathrm{Ran}(\theta)$

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_{\bar{\theta},\mu}(A_{\bar{\theta}}) \leq{}& \frac{\lambda\mathrm{Tr}(\theta^\dagger B_\rho)}{2} + \frac{L^2(\log(n)+1)\mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\,\bar{\theta}C_\rho\Big)}{\lambda n} \\ &+ \frac{\eta\|\theta-\theta_0\|_F^2}{2T} + \frac{L^4\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,2}}{8\lambda^2\eta}.\end{aligned} \tag{6.32}$$

Hence, optimizing w.r.t. the hyper-parameters λ and η, for

$$\lambda = L\sqrt{\frac{1}{\mathrm{Tr}(\theta^\dagger B_\rho)}\left(\frac{2(\log(n)+1)\mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\,\bar{\theta}C_\rho\Big)}{n} + \|\theta-\theta_0\|_F\sqrt{\frac{\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,2}}{T}}\right)} \tag{6.33}$$

$$\eta = \frac{L^2\sqrt{T\,\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,2}}}{2\lambda\|\theta-\theta_0\|_F}, \tag{6.34}$$

we get

$$\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_{\bar{\theta},\mu}(A_{\bar{\theta}}) \le L\sqrt{\mathrm{Tr}(\theta^\dagger B_\rho)\left(\frac{2(\log(n)+1)\,\mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\,\bar{\theta}C_\rho\Big)}{n} + \|\theta-\theta_0\|_F\sqrt{\frac{\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,2}}{T}}\right)}.$$

Proof. Specializing Thm. 9 to the quantities outlined in Ex. 2, exploiting the bound on the norm of the approximated meta-subgradients given in Eq. (6.10) (exploiting Asm. 3) and using the notation in Eq. (6.31) and Eq. (5.2), for any $\theta \in \mathcal{S}$ such that $\mathrm{Ran}(B_\rho) \subseteq \mathrm{Ran}(\theta)$, we get the following

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathcal{E}_{\bar{\theta},\mu}(A_{\bar{\theta}}) \le{}& \frac{\lambda\mathrm{Tr}(\theta^\dagger B_\rho)}{2} + \frac{1}{2\lambda nT}\,\mathbb{E}_{\mathbf{Z}}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|{\theta_t}^{1/2}x_{t,i}s'_{\theta_t,i}\right\|_2^2 \\ &+ \frac{\eta\|\theta-\theta_0\|_F^2}{2T} + \frac{L^4\mathbb{E}_{\mathbf{Z}}\,\|C^{\mathrm{tot}}\|_{\infty,2}}{8\lambda^2\eta} \\ &+ \frac{1}{2\lambda n}\,\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\sum_{i=1}^{n}\frac{1}{i}\left\|\bar{\theta}^{1/2}x_i s'_{\bar{\theta},i}\right\|_2^2.\end{aligned} \tag{6.35}$$

The desired statement derives from the above inequality and from observing that, thanks to Asm. 3, the i.i.d. sampling of the data and the fact that θ_t depends only on the previous datasets $(Z_j)_{j=1}^{t-1}$, using the inequality $\sum_{i=1}^{n} 1/i \le \log(n)+1$ and the definition of C_ρ in Eq. (5.3), we have

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}}\,\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\left\|{\theta_t}^{1/2}x_{t,i}s'_{\theta_t,i}\right\|_2^2 &\le L^2\mathbb{E}_{\mathbf{Z}}\,\mathrm{Tr}\Big(\frac{1}{T}\sum_{t=1}^{T}\theta_t\sum_{i=1}^{n}\frac{1}{i}\,x_{t,i}x_{t,i}^\top\Big) \\ &= L^2(\log(n)+1)\mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\,\bar{\theta}C_\rho\Big)\end{aligned} \tag{6.36}$$

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\sum_{i=1}^{n}\frac{1}{i}\left\|\bar{\theta}x_i s'_{\bar{\theta},i}\right\|_2^2 &\le L^2\mathbb{E}_{\mathbf{Z}}\,\mathbb{E}_{\mu\sim\rho}\,\mathbb{E}_{Z\sim\mu^n}\,\mathrm{Tr}\Big(\bar{\theta}\sum_{i=1}^{n}\frac{1}{i}\,x_i x_i^\top\Big) \\ &= L^2(\log(n)+1)\mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\,\bar{\theta}C_\rho\Big).\end{aligned} \tag{6.37}$$

■

In order to evaluate the quality of the bound above, we specify Thm. 11 to Ex. 2, that is, we provide an (regularized) expected across-tasks excess risk bound for $\bar{w}_\theta$, the average of the iterations returned by running the within-task Alg. 6 with an appropriate feature map θ fixed in hindsight for any task.

Corollary 29 (Across-Tasks Excess Transfer Risk Bound for Alg. 6, Feature Map)**.** *Let Asm. 3 hold and consider the statistical setting in Thm. 11 applied to Ex. 2. In particular, for any $\theta \in \mathcal{S}$, let A_θ be the corresponding inner Alg. 6. Then, according to the notation in Eq. (6.31) and Eq. (5.3), the following (regularized) expected across-tasks excess risk bound holds for any $\theta \in \mathcal{S}$ such that* $\mathrm{Ran}(B_\rho) \subseteq \mathrm{Ran}(\theta)$

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_{\theta,\mu}(A_\theta) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger B_\rho)}{2} + \frac{L^2(\log(n)+1)\mathrm{Tr}(\theta C_\rho)}{2\lambda n}. \tag{6.38}$$

Hence, optimizing w.r.t. the hyper-parameter λ, for

$$\lambda = L\sqrt{\frac{(\log(n)+1)\mathrm{Tr}(\theta C_\rho)}{n\mathrm{Tr}(\theta^\dagger B_\rho)}}, \tag{6.39}$$

we get

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_{\theta,\mu}(A_\theta) \leq L\sqrt{\frac{(\log(n)+1)\mathrm{Tr}(\theta^\dagger B_\rho)\mathrm{Tr}(\theta C_\rho)}{n}}.$$

Proof. Specializing Thm. 11 to the quantities outlined in Ex. 2, using the notation in Eq. (6.31), for any $\theta \in \mathcal{S}$ such that $\mathrm{Ran}(B_\rho) \subseteq \mathrm{Ran}(\theta)$, we get the following

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_{\theta,\mu}(A_\theta) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger B_\rho)}{2} + \frac{1}{2\lambda n}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n} \sum_{i=1}^n \frac{1}{i} \left\| \theta^{1/2} x_i s'_{\theta,i} \right\|_2^2.$$

The desired statement derives from the above inequality and from observing that, under Asm. 3, exploiting the i.i.d. sampling of the data and the inequality $\sum_{i=1}^n 1/i \leq \log(n)+1$, introducing the definition of the matrix C_ρ in Eq. (5.3), we can write

$$\begin{aligned} \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{z_n\sim\mu^n} \sum_{i=1}^n \frac{1}{i} \left\| \theta^{1/2} x_i s'_{\theta,i} \right\|_2^2 &\leq L^2 \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathrm{Tr}\Big(\theta \sum_{i=1}^n \frac{1}{i}\, x_i x_i^\top\Big) \\ &\leq L^2(\log(n)+1)\mathrm{Tr}(\theta C_\rho). \end{aligned} \tag{6.40}$$

■

Also in this case, the comments to the bounds above are postponed in the following Sec. 6.5. In the section below, we specify to Ex. 2 the OWB variant of our Meta-Learning method and the corresponding analysis.

6.4 The Statistical Online-Within-Batch Variant

In this section, we consider the within-task batch RERM algorithm in Eq. (2.24) applied to the setting outlined in Ex. 2, i.e., for any $\theta \in \mathcal{S}$ and any dataset Z, we consider

$$\hat{w}_\theta = \operatorname*{argmin}_{w \in \mathrm{Ran}(\theta)} \mathcal{R}_Z(w) + \frac{\lambda}{2}\langle w, \theta^\dagger w \rangle. \tag{6.41}$$

In the following, we specify Thm. 14 to Ex. 2, that is, we provide an expected meta-excess risk bound for $\hat{w}_{\bar{\theta}}$, the RERM in Eq. (6.41) with feature map $\bar{\theta}$, the average of the feature maps returned by the meta-algorithm in Alg. 7 working with exact meta-subgradients computed by Eq. (6.11).

Corollary 30 (OWB Meta-Excess Risk Bound, Feature Map)**.** *Let Asm. 3 hold and consider the statistical setting in Thm. 14 applied to Ex. 2. In particular, let $A_{\bar{\theta}}$ be the inner RERM algorithm in Eq. (6.41) with feature map $\bar{\theta}$, the average of the feature maps returned by the meta-algorithm in Alg. 7 using the data $\mathbf{Z}$ and the exact meta-subgradients in Eq. (6.11). Then, according to the notation in Eq. (6.31), Eq. (5.2) and Eq. (5.3), the following expected meta-excess risk bound holds for any $\theta \in \mathcal{S}$ such that* $\mathrm{Ran}(B_\rho) \subseteq \mathrm{Ran}(\theta)$

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_{\bar{\theta}}) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger B_\rho)}{2} + \frac{2L^2 \mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\, \bar{\theta} C_\rho\Big)}{\lambda n} + \frac{\eta \|\theta - \theta_0\|_F^2}{2T} + \frac{L^4 \mathbb{E}_{\mathbf{Z}}\, \|C^{\mathrm{tot}}\|_{\infty,2}}{8\lambda^2 \eta}. \tag{6.42}$$

Hence, optimizing w.r.t. the hyper-parameters λ and η, for

$$\lambda = L\sqrt{\frac{1}{\mathrm{Tr}(\theta^\dagger B_\rho)}\left(\frac{4\mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\, \bar{\theta} C_\rho\Big)}{n} + \|\theta - \theta_0\|_F \sqrt{\frac{\mathbb{E}_{\mathbf{Z}}\, \|C^{\mathrm{tot}}\|_{\infty,2}}{T}}\right)} \qquad \eta = \frac{L^2\sqrt{T\, \mathbb{E}_{\mathbf{Z}}\, \|C^{\mathrm{tot}}\|_{\infty,2}}}{2\lambda \|\theta - \theta_0\|_F},$$

we get

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_{\bar{\theta}}) \leq L\sqrt{\mathrm{Tr}(\theta^\dagger B_\rho)\left(\frac{4\mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\, \bar{\theta} C_\rho\Big)}{n} + \|\theta - \theta_0\|_F \sqrt{\frac{\mathbb{E}_{\mathbf{Z}}\, \|C^{\mathrm{tot}}\|_{\infty,2}}{T}}\right)}. \tag{6.43}$$

Proof. Specializing Thm. 14 to the quantities outlined in Ex. 2, exploiting the bound on the norm of the exact meta-subgradients given in Eq. (6.10) (exploiting Asm. 3) and using the notation in Eq. (6.31) and Eq. (5.2), for any $\theta \in \mathcal{S}$ such that $\mathrm{Ran}(B_\rho) \subseteq \mathrm{Ran}(\theta)$, we get

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_{\bar{\theta}}) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger B_\rho)}{2} + \frac{2}{\lambda n}\, \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z_i'\sim\mu} \left\| \bar{\theta}^{1/2} x_i' s_{\bar{\theta},i}' \right\|_2^2 + \frac{\eta \|\theta - \theta_0\|_F^2}{2T} + \frac{L^4 \mathbb{E}_{\mathbf{Z}} \|C^{\mathrm{tot}}\|_{\infty,2}}{8\eta\lambda^2}.$$

The statement derives from the above inequality, observing that, under Asm. 3, exploiting the i.i.d. sampling of the data and introducing the definition of C_ρ in Eq. (5.3), we can write

$$\mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z_i'\sim\mu} \left\| \bar{\theta}^{1/2} x_i' s_{\bar{\theta},i}' \right\|_2^2 \leq L^2 \mathbb{E}_{\mathbf{Z}}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathrm{Tr}(\bar{\theta} x_i' x_i'^\top) = L^2 \mathrm{Tr}\Big(\mathbb{E}_{\mathbf{Z}}\, \bar{\theta} C_\rho\Big)$$

■

Also in this case, in order to evaluate the quality of the bound above, we specify Thm. 15 to Ex. 2, that is, we provide an expected across-tasks excess risk bound for $\hat{w}_\theta$, the within-task RERM algorithm in Eq. (6.41) with feature map θ fixed in hindsight for any task.

Corollary 31 (Across-Tasks Excess Risk Bound for the RERM Algorithm in Eq. (6.41), Feature Map)**.** *Let Asm. 3 hold and consider the statistical setting in Thm. 15 applied to Ex. 2. In particular, for any $\theta \in \mathcal{S}$, let A_θ be the corresponding inner RERM algorithm in Eq. (6.41). Then, according to the notation in Eq. (6.31) and Eq. (5.3), the following expected across-tasks excess risk bound holds for any $\theta \in \mathcal{S}$ such that* $\mathrm{Ran}(B_\rho) \subseteq \mathrm{Ran}(\theta)$

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta) \leq \frac{\lambda \mathrm{Tr}(\theta^\dagger B_\rho)}{2} + \frac{2L^2 \mathrm{Tr}(\theta C_\rho)}{\lambda n}. \tag{6.44}$$

Hence, optimizing w.r.t. the hyper-parameter λ, for

$$\lambda = 2L \sqrt{\frac{\mathrm{Tr}(\theta C_\rho)}{n \mathrm{Tr}(\theta^\dagger B_\rho)}}, \tag{6.45}$$

we get

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta) \leq\ 2L \sqrt{\frac{\mathrm{Tr}(\theta^\dagger B_\rho)\ \mathrm{Tr}(\theta C_\rho)}{n}}. \tag{6.46}$$

Proof. Specializing Thm. 15 to the quantities outlined in Ex. 2, using the notation in Eq. (6.31), for any $\theta \in \mathcal{S}$ such that $\text{Ran}(B_\rho) \subseteq \text{Ran}(\theta)$, we get the following

$$\mathbb{E}_{\mu\sim\rho}\, \mathcal{E}_\mu(A_\theta) \leq \frac{\lambda \text{Tr}(\theta^\dagger B_\rho)}{2} + \frac{2}{\lambda n}\, \mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z_i'\sim\mu} \left\| \theta^{1/2} x_i' s_{\theta,i}' \right\|_2^2. \tag{6.47}$$

The statement derives from the above inequality observing that, under Asm. 3, introducing the definition of C_ρ in Eq. (5.3), we can write

$$\mathbb{E}_{\mu\sim\rho}\, \mathbb{E}_{Z\sim\mu^n}\, \mathbb{E}_{z_i'\sim\mu} \left\| \theta^{1/2} x_i' s_{\theta,i}' \right\|_2^2 \leq L^2 \text{Tr}(\theta C_\rho). \tag{6.48}$$

■

In the following section we present a detailed discussion about the bounds we have reported in the above sections.

6.5 Discussion

We start from discussing the results in Cor. 27, Cor. 29 and Cor. 31, where the feature map used by the inner algorithm is fixed in hindsight for any task.

Advantage of Selecting the Right Feature Map. We first comment the bounds in the statistical setting in Cor. 29 and Cor. 31. In this case, proceeding in the same way as described for Ex. 1, we should define the best algorithm in our class (*oracle*) the algorithm associated to the feature map minimizing the bound in Cor. 29 or Cor. 31. However, in our case, to simplify the analysis we consider as the oracle the algorithm associated to the feature map θ_ρ minimizing only a part of the above bound which is available is closed form. Specifically, appealing to the infimal formulation of the MTL trace norm regularizer in Eq. (2.26) and to (Argyriou et al., 2008a, Eq. (13)), we minimize only the term $\text{Tr}(\theta^\dagger B_\rho)$ over the subset of the feature maps $\{\theta \in \mathcal{S} : \text{Ran}(B_\rho) \subseteq \text{Ran}(\theta)\}$ for which our bound holds:

$$\min_{\theta\in\mathcal{S}:\text{Ran}(B_\rho)\subseteq\text{Ran}(\theta)} \text{Tr}(\theta^\dagger B_\rho) = \text{Tr}(B_\rho^{1/2})^2 = \|W_\rho\|_{\text{Tr}}^2, \tag{6.49}$$

where W_ρ is a square root of B_ρ. We consider as the optimal feature map the corresponding minimizer

$$\theta_\rho = \operatorname*{argmin}_{\theta\in\mathcal{S}:\text{Ran}(B_\rho)\subseteq\text{Ran}(\theta)} \text{Tr}(\theta^\dagger B_\rho) = \frac{W_\rho}{\text{Tr}(W_\rho)}. \tag{6.50}$$

Similarly to what observed in Sec. 5.5 for the setting in Ex. 1, we will consider this feature map as benchmark in order to evaluate the performance of our Meta-Learning procedures. Furthermore, also in this case, in order to avoid further notation, we over-write the symbol used in Sec. 2.2 for the true optimal feature map, to denote this sub-optimal version. With such a choice of feature map θ_ρ, the bounds in Cor. 29 (up to logarithmic factors) and in Cor. 31 become proportional to

$$\sqrt{\frac{\mathrm{Tr}(\theta_\rho{}^\dagger B_\rho)\ \mathrm{Tr}(\theta_\rho C_\rho)}{n}} \leq \|W_\rho\|_{\mathrm{Tr}} \sqrt{\frac{\|C_\rho\|_\infty}{n}}, \tag{6.51}$$

where, in the inequality above, we have applied Holder's inequality (see Lemma 32 in App. A) to the matrices' scalar product and we have exploited the fact $\mathrm{Tr}\,(\theta_\rho) = 1$.

On the other hand, solving the tasks independently (ITL), in this case, corresponds to apply Alg. 6 with the feature map $\theta_{\mathrm{ITL}} = I/d$ for any task. Substituting this value, the bounds above become proportional to

$$\|W_\rho\|_F \sqrt{\frac{\mathrm{Tr}(C_\rho)}{n}}. \tag{6.52}$$

Comparing the bounds in Eq. (6.51) and Eq. (6.52), we can conclude that there is an advantage in using the optimal feature map w.r.t. solving each task independently, when the tasks are *similar* in the sense that $\|C_\rho\|_\infty \ll \mathrm{Tr}\,(C_\rho)$ (e.g. when the inputs are high-dimensional) and $\|W_\rho\|_{\mathrm{Tr}}$ is comparable to $\|W_\rho\|_F$ (i.e. when the matrix W_ρ is low-rank), meaning that the tasks' target vectors are expected to lie in a low-dimensional subspace, i.e. the range of the optimal feature map. This is inline with previous literature, such as (Denevi et al., 2018a; Maurer et al., 2013, 2016).

Regarding the non-statistical setting, in order to comment the average meta-regret bound in Cor. 27, one can proceed as above introducing the corresponding sub-optimal algorithm in the class associated to the corresponding sub-optimal feature map $\hat{\theta}$. The associated bound, in this case, becomes proportional to

$$\|\hat{W}\|_{\mathrm{Tr}} \sqrt{\frac{\|\hat{C}^{\mathrm{tot}}\|_\infty}{n}}, \tag{6.53}$$

where $\hat{W}$ is a square root of $\hat{B}$. Comparing this last bound to the corresponding bound for ITL

$$\|\hat{W}\|_F \sqrt{\frac{\mathrm{Tr}\,(\hat{C}^{\mathrm{tot}})}{n}}, \tag{6.54}$$

we see that there is an advantage in using our Meta-Learning method w.r.t. solving each task independently, when $\|\hat{C}^{\mathrm{tot}}\|_\infty \ll \mathrm{Tr}(\hat{C}^{\mathrm{tot}})$ and $\|\hat{W}\|_{\mathrm{Tr}}$ is comparable to $\|\hat{W}\|_F$ (i.e. $\hat{W}$ is low-rank). The first condition on the weighted input covariance matrix $\hat{C}^{\mathrm{tot}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{i}\,x_{t,i}x_{t,i}^\top$

is less clear to interpret than the more natural one $\|C^{\text{tot}}\|_\infty \ll \text{Tr}\,(C^{\text{tot}})$ with the standard empirical input covariance matrix $C^{\text{tot}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n} x_{t,i}x_{t,i}^\top$. However, in certain data configurations these two input covariance matrices, may still be closed one to each other. We think that this issue is avoidable by choosing the inner step size in different way and we will address it in future work.

We now can make the following observations about the bounds we have obtained in Cor. 26, Cor. 28 and Cor. 30 for our Meta-Learning procedures.

Bounds for the Feature Map Resulting from our Meta-Learning Methods. In order to analyze the effectiveness of our Meta-Learning methods, we investigate whether they mimic the performance of the best algorithm in the class, when the number of training tasks is sufficiently large w.r.t. the number of within-task points. In such a case, the term $T^{-1/4}$ is negligible and, applying Holder's inequality and exploiting the fact that $\text{Tr}(\bar{\theta}) \leq 1$ as described above, the bounds in Cor. 28 (up to logarithmic factors) and Cor. 30 can be upper bounded by

$$\sqrt{\frac{\text{Tr}(\theta^\dagger B_\rho)\text{Tr}(C_\rho)}{n}}, \tag{6.55}$$

where $\theta \in \mathcal{S}$ is the fixed feature map in the statement, defining the optimal choice of the hyperparameters for our method. In particular, choosing $\theta = \theta_\rho$ in Eq. (6.50), the quantity above in Eq. (6.55) can be upper bounded by the bound in Eq. (6.51) for the best algorithm in the class. As a consequence, when the tasks are similar as explained above, our methods can provide a significant advantage w.r.t. ITL in the statistical setting. We conclude observing that the average meta-regret bound in Cor. 26 for the non-statistical setting is less clear to interpret because of the presence of the modified version of the covariance matrix $\hat{C}^{\text{tot}}_{\theta_{1:T}}$. Future work may be devoted to investigate this point, which could be either an artifact of our analysis or due to some intrinsic characteristics of the feature learning problem we are considering.

Finally, in order to investigate the impact of considering the OWO framework instead of the OWB one, we compare Cor. 28 and Cor. 29 to Cor. 30 and Cor. 31. We recall that all these statements hold for the statistical setting.

Comparison Between OWO and OWB. Comparing the OWO bound in Cor. 29 to the OWB bound in Cor. 31, we can expect that running the batch RERM algorithm in Eq. (6.41) with a fixed feature map θ in hindsight for any task will outperform the twin method with the online Alg. 6 by a factor $(\log(n)+1)\text{Tr}(\theta^\dagger B_\rho)\text{Tr}(\theta C_\rho)$. As a consequence, for a fixed value of n, the gap between the performance of the two methods will depend on the feature map θ used by the algorithm which can amplify or reduce the discrepancy. In particular, for the optimal feature

map $\theta = \theta_\rho$ in Eq. (6.50), this quantity can be upper bounded (up to logarithmic factors) as in Eq. (6.51). Hence, for such a choice, the corresponding gap between the two methods can be insignificant when the associated quantities $\|W_\rho\|_{\mathrm{Tr}}$ and $\|C_\rho\|_\infty$ are small. A similar observation can be made also for the Meta-Learning methods analyzed in Cor. 28 and Cor. 30. In this case, the gap above is proportional to $(\log(n)+1)\mathrm{Tr}(\theta^\dagger B_\rho)\mathrm{Tr}(\mathbb{E}_{\mathbf{Z}}\,\bar{\theta}C_\rho)$, where θ is the fixed feature map determining the hyper-parameters used in the methods and $\bar{\theta}$ is the average feature map estimated the meta-algorithm. As already observed, for the optimal feature map $\theta = \theta_\rho$, this quantity can be upper bounded (up to logarithmic factors) as described above in Eq. (6.51) and the same considerations hold.

Finally, note that the bounds we have obtained for the feature map setting in Ex. 2 converge, as the number of tasks grows, to the corresponding bounds for the best algorithm at a rate $\mathcal{O}(T^{-1/4})$, whereas, the corresponding bounds for the bias setting in Ex. 1 yield a faster rate $\mathcal{O}(T^{-1/2})$, suggesting that the feature learning problem is more difficult than the bias one.

6.6 Experiments

In this section we test the performance of the Meta-Learning method described in this work on synthetic and real data in the statistical setting of Ex. 2[1].

Also in this case, we validated the hyper-parameters λ and η by the procedure described in App. C.1 and we compared the proposed OWO Meta-Learning approach with methods based on the batch RERM algorithm in Eq. (6.41). More precisely, we evaluated the performance of the following methods:

- META - OWO: our Online-Within-Online Meta-Learning method described in Chpt. 3, where we use the inner online Alg. 6 both during meta-training and meta-validation / testing phases;
- META - Hybrid: the hybrid Meta-Learning method in which we use exact meta-subgradients (computed by the batch RERM in Eq. (6.41), as described in Prop. 12) during the meta-training phase, but we apply the online inner Alg. 6 during the meta-validation / testing phases;
- ITL - O: we use the online Alg. 6 with the ITL feature map $\theta_{\mathrm{ITL}} = I/d$ for any task;

[1] The code is available at https://github.com/dstamos/Adversarial-LTL

- Oracle - O: we use the online Alg. 6 with the optimal feature map θ_ρ in Eq. (6.50) for any task (only in synthetic experiments in which this quantity is available).

We compared the above methods in the following synthetic and real experimental settings, where, as described in App. C.2, also in this case, we computed an approximation of the RERM algorithm in Eq. (6.41) by applying FISTA (see (Beck and Teboulle, 2009, Sec. 4)) on the associated within-task dual problem. In all the experiments, we fixed the starting point for the meta-algorithm Alg. 7 equal to the ITL feature map $\theta_0 = \theta_{\text{ITL}} = I/d$.

Synthetic data. We tested the performance of the above methods over regression tasks, measuring the errors by the absolute loss. Specifically, we generate $T_{\text{tot}} = 3600$ tasks, where, for each task, the corresponding dataset $(x_i, y_i)_{i=1}^{n_{\text{tot}}}$ of $n_{\text{tot}} = 80$ points was generated according to the linear regression equation $y = \langle x, w_\mu \rangle + \epsilon$, with x sampled uniformly on the unit sphere in $\mathbb{R}^d$ with $d = 20$ and ϵ sampled from a zero-mean Gaussian distribution with standard deviation 0.2. The tasks' target vectors w_μ were generated as $w_\mu = P\tilde{w}_\mu$ with the components of $\tilde{w}_\mu \in \mathbb{R}^{d/5}$ sampled from a zero-mean Gaussian distribution with standard deviation 1 and then $\tilde{w}_\mu$ normalized to have unit norm, with $P \in \mathbb{R}^{d \times d/5}$ a matrix with orthonormal columns. In this setting, the operator norm of the input covariance C_ρ in Eq. (5.3) is small (equal to $1/d$) and the target vectors' covariance matrix B_ρ in Cor. 28 is low-rank, a favorable setting for our method, according to our theory. In order to validate the inner regularization parameter λ and the meta-step size η, we followed the procedure described in App. C.1. Specifically, we considered 14 candidates values for both λ and η in the range $[10^{-5}, 10^5]$ with logarithmic spacing and we evaluated the performance of the estimated feature maps by using $T = T_{\text{tr}} = 3000$, $T_{\text{va}} = 100$, $T_{\text{te}} = 500$ of the above tasks for meta-training, meta-validation and meta-testing, respectively. Moreover, in order to train and to test the inner algorithm, we used $n = n_{\text{tr}} = 50\%\ n_{\text{tot}}$ and $n_{\text{te}} = 50\%\ n_{\text{tot}}$ points in each dataset.

In Fig. 6.1 (Top) we reported the test error for all the methods above. The results are inline with those in Sec. 5.6 for the setting outlined in Ex. 1. More specifically, looking at the results, we can state that, in this setting, our OWO Meta-Learning method (META - OWO) outperforms Independent-Task Learning (ITL - O) and it tends to the best algorithm in the class (Oracle - O) as the number of training tasks increases. Moreover, the performance of the methods META - OWO and META - Hybrid are comparable, suggesting that, also in this case, our approximation of the meta-subgradients is an effective way to keep the entire process fully online.

Real data. We evaluated our method on the Movielens 100k dataset containing the ratings of different users to different movies[2]. We considered each user as a task and, removing all

[2]See https://grouplens.org/datasets/movielens/

movies that have been seen less than 20 times, we ended with a total number of $T_{\text{tot}} = 939$ users and $n_{\text{tot}} = 939$ movies. We casted each task as a regression problem, where the labels are the ratings of the users and the raw features are simply the index of the movie (i.e. we reformulated the problem in a matrix completion setting, where $d = n$). Also in this case, we used the absolute loss to measure the errors. To validate the hyper-parameters λ and η, following the procedure described in App. C.1, we considered 14 candidates values for both λ and η in the range $[10^{-5}, 10^{5}]$ with logarithmic spacing and we evaluated the performance of the estimated feature maps by splitting the tasks into $T = T_{\text{tr}} = 700$, $T_{\text{va}} = 100$, $T_{\text{te}} = 139$ tasks used for meta-training, meta-validation and meta-testing, respectively. Moreover, in order to train and to test the inner algorithm, we splitted each within-task dataset into $n = n_{\text{tr}} = 75\%\ n_{\text{tot}}$ and $n_{\text{te}} = 25\%\ n_{\text{tot}}$ points.

In the above experiment, we observed that the performance of the algorithm ITL - O was very bad. In fact, looking at the closed form of Alg. 6 with $\theta_{\text{ITL}} = I/d$, it is possible to show that, in the considered formulation of the problem, the algorithm ITL - O is not able to predict any rate for the films without any observed rate. This explains the bad performance we got. For this reason, in order to evaluate the performance of the Meta-Learning methods (META - OWO and META - Hybrid), we decided to substitute this silly method with a more challenging strategy for this particular formulation of the problem. Specifically, we introduced a batch Meta-Learning method in which, for the films without any observed rate, we predicted, at the end of the entire sequence of tasks, the rate coinciding with the average of the rates of all the observed users. We denoted this method as META - B (Batch).

The results we got are reported in Fig. 6.1 (Bottom) and they are consistent with the synthetic experiments above, showing the effectiveness of our feature map Meta-Learning method also in real-life scenarios. In particular, we note that both the online Meta-Learning methods (META - OWO and META - Hybrid) outperform the batch Meta-Learning method (META - B) described above, as the number of training tasks increases.

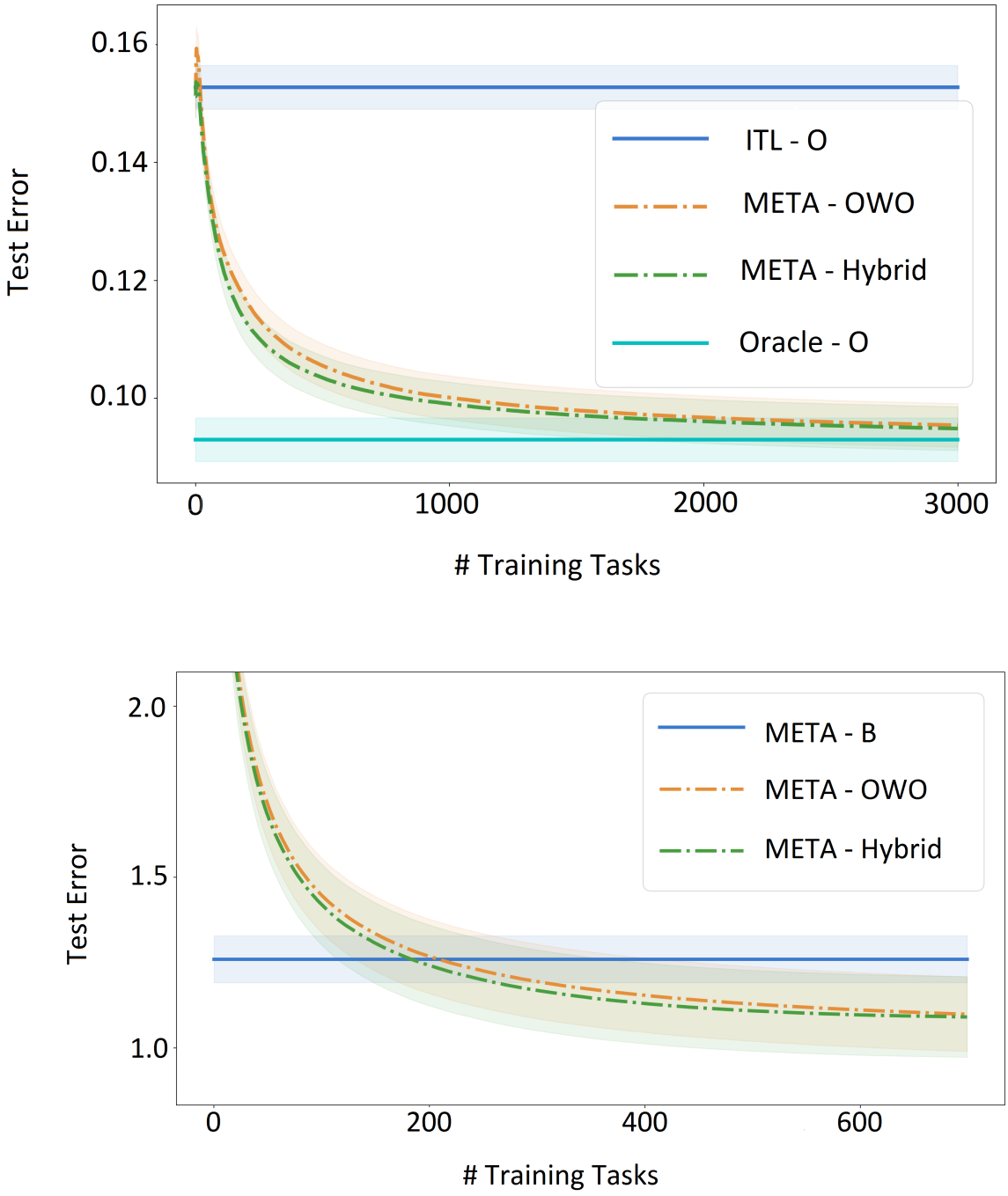

Figure 6.1 **Synthetic Data (Top) and Movielens Dataset (Bottom).** Test error of different methods as the number of training tasks increases. Regression with absolute loss. The results are averaged over 10 independent generations / splits of the data.

Chapter 7

Conclusion and Future Directions

Starting from recalling the standard Single-Task Learning setting, we introduced and formalized the Meta-Learning problem. In such a problem the broad goal is to select an algorithm in a prescribed family (within-task algorithm) that is well suited to address a class of similar learning problems (tasks). Practically, this goal can be addressed by designing a Meta-Learning procedure (meta-algorithm) aiming at inferring the tasks' similarity from a sequence of observed tasks. Such information is then exploited in order to select an appropriate within-task algorithm in the family.

In this dissertation, we focused on the so-called Online-Within-Online (OWO) Meta-Learning setting in which data are received and processed sequentially both within and across the tasks. Specifically, inspired by a common formulation shared by many Multi-Task Learning problems, we presented an OWO Meta-Learning method, stemming from primal-dual Online Learning. The method can cover various types of tasks' relatedness and it can be adapted to a wide class of learning algorithms. By means of a new analysis technique, we derived a meta-regret bound for our method, based on which, it is also possible to obtain guarantees in the statistical setting by online-to-batch arguments. We also showed that, in specific settings, the proposed method can provide comparable statistical guarantees as its more expensive variant in which the within-task data are processed in one single batch. Finally, we illustrated our framework with two important examples in which the method attempts to infer a bias vector or a feature map shared among the tasks, recovering or improving upon state-of-the-art results. We think that the generality of our framework and our method of proof could be a valuable starting point for future theoretical investigations on Meta-Learning.

In the future, it would be valuable to investigate the tightness of our bounds with respect to the number of within-task points and the number of tasks. In this case, it is not clear whether

summing over the tasks pre-existing lower bounds for a fixed single task algorithm – such as the bounds presented in (Mcmahan and Streeter, 2012; McMahan and Orabona, 2014) – can return an accurate benchmark or something more sophisticate, taking into account the interplay between the inner and the meta-algorithm, is required.

Another open point is to extend the analysis we developed in this work also to non Lipschitz loss functions, such as the square and the logistic loss. In such a case, the main difficulty arising is how to control in the bounds the norm of the subgradients and the approximation error on them. Inspired by (Shalev-Shwartz and Ben-David, 2014, Eq. 12.6), we tried to control these quantities exploiting the self-boundedness and the non-negativity of the loss functions above, but this approach did not reveal to be effective as the Lipschitz assumption, returning bounds with quantities we did not manage to effectively control, such as $\sum_{t=1}^{T} \mathcal{L}_t(\theta_t)$.

In addition to this, it would be interesting to explore even more flexible and effective Meta-Learning strategies, which can be applied more naturally to real world scenarios. Regarding this aspect, as already pointed out in Rem. 11, one important point to tackle is to investigate how the choice of the hyper-parameters in our method could be addressed in practice in a more effective and theoretically grounded way. A possible starting step for answering this question could be to understand whether the so-called 'parameter-free methods' proposed in (Mcmahan and Streeter, 2012; McMahan and Orabona, 2014; Orabona, 2014; Orabona and Pál, 2016; Zhuang et al., 2019) for the Single-Task Learning setting could be adapted to address and to cover a Meta-Learning framework similar to the one presented here. However, since the methods above seem to be quite different from the primal-dual Online Learning framework considered here, at the moment, the solution of this problem does not seem straightforward.

Another interesting point is to study how our framework can be adapted to more realistic settings in which data deriving from different tasks are received simultaneously and the inner algorithm is updated more frequently (not necessarily at the end of each task as described in this work). In this case, to provide the method with a memory able to efficiently keep track of the tasks already encountered in the past seems to be a necessary step for the success of the project. Moreover, the meta-objectives $(\mathcal{L}_t)_{t=1}^{T}$ used here seem to be not appropriate for this evolving setting, since they are defined over a continuous stream of data coming from the *same* task.

Finally, another possible research direction may be to study the applicability of the proposed framework to time series and meta-reinforcement learning. However, since the formulation of the problem in these settings is different from the one considered in this work, at the moment, such extension does not seem straightforward and it requires further investigation.

Appendix A

Convex Analysis

In this appendix we recall some basic concepts of convex analysis. We refer to (Bauschke and Combettes, 2011; Bertsekas et al., 2003; Borwein and Lewis, 2010; Boyd and Vandenberghe, 2004; Jean-Baptiste, 2010; Peypouquet, 2015) for a complete and detailed overview.

Let $\mathcal{V}$ be an Euclidean space, i.e a finite dimensional real vector space endowed with an inner product $\langle \cdot, \cdot \rangle$. Moreover, for a generic norm $\| \cdot \|$ over $\mathcal{V}$, we recall that its dual norm $\| \cdot \|_*$ at the point $\alpha \in \mathcal{V}$ is defined as

$$\|\alpha\|_* = \sup_{v \in \mathcal{V}: \|v\| \leq 1} \langle \alpha, v \rangle. \tag{A.1}$$

As direct consequence of the definition above, we have the following standard fact.

Lemma 32 (Generalized Holder's Inequality)**.** *For any $\alpha, w \in \mathcal{V}$,*

$$\langle \alpha, w \rangle \leq \|\alpha\|_* \, \|w\|. \tag{A.2}$$

Proof. We start from observing that $\|w\| = 0$ if, and only if, $w = 0$. If $w = 0$, the statement above is obvious. Thus, we consider the case $w \neq 0$. In such a case, by definition of the dual norm, we can write the following

$$\langle \alpha, w \rangle = \|w\| \left\langle \alpha, \frac{w}{\|w\|} \right\rangle \leq \|w\| \, \|\alpha\|_*. \tag{A.3}$$

This coincides with the desired statement. ■

In the following, we consider extended real-valued functions. We start from giving the following basic definitions, which are frequently used in this dissertation.

Definition 33 (ϵ-Minimizer)**.** *A point $\hat{v}_\epsilon \in \mathcal{V}$ is an ϵ-minimizer (with $\epsilon \geq 0$) of a function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ if, for any $v \in \mathcal{V}$,*

$$f(\hat{v}_\epsilon) \leq f(v) + \epsilon. \tag{A.4}$$

The concept of exact minimizer is retrieved from the definition above by setting $\epsilon = 0$. Moreover, an ϵ-maximizer of a function f must be intended as an ϵ-minimizer of the opposite function $-f$.

Definition 34 (Domain of a Function, see e.g. (Peypouquet, 2015, Sec. 2.1))**.** *For a given function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$, define its domain as*

$$\mathrm{Dom} f = \left\{ v \in \mathcal{V} : f(v) < +\infty \right\} \subseteq \mathcal{V}. \tag{A.5}$$

Definition 35 (Epigraph of a Function, see e.g. (Peypouquet, 2015, Sec. 2.1))**.** *For a given function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$, define its epigraph as*

$$\mathrm{Epi} f = \left\{ (v, t) \in \mathcal{V} \times \mathbb{R} : f(v) \leq t \right\} \subseteq \mathcal{V} \times \mathbb{R}. \tag{A.6}$$

The above quantities are now exploited to introduce the following basic definitions.

Definition 36 (Proper Function, see e.g. (Peypouquet, 2015, Sec. 2.1))**.** *A function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ is proper if* $\mathrm{Dom} f \neq \emptyset$.

Definition 37 (Closed or Lower Semi-Continuous Function, see e.g. (Peypouquet, 2015, Sec. 2.2))**.** *A function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ is closed or lower semi-continuous if* $\mathrm{Epi} f$ *is a closed set of $\mathcal{V} \times \mathbb{R}$.*

Definition 38 (Convex Function, see e.g. (Peypouquet, 2015, Sec. 2.3))**.** *A function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ is convex if, for any $t \in [0, 1]$ and any $v, v' \in \mathrm{Dom} f$,*

$$f(tv + (1-t)v') \leq tf(v) + (1-t)f(v'). \tag{A.7}$$

The above inequality is known as Jensen's inequality and it can be extended to combinations of more points or expectations of random variables in the following way.

Lemma 39 (Convex Functions and Generalized Jensen's Inequality, see e.g. (Boyd and Vandenberghe, 2004, Sec. 3.1.8))**.** *Let $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ be a convex function and consider a random variable X taking values in* Dom$f$ *with probability* 1*. Then, provided that the following expectations exist,*

$$f(\mathbb{E}\, X) \leq \mathbb{E}\, f(X). \tag{A.8}$$

In particular, in the discrete case, for any sequence of vectors $(v_j)_{j=1}^m \in \mathcal{V}^m$ and weights $(a_j)_{j=1}^m \in \mathbb{R}^m$ such that $a_j \geq 0$ for any $j \in \{1, \ldots, m\}$ and $\sum_{j=1}^m a_j = 1$, we have

$$f\Big(\sum_{j=1}^m a_j v_j\Big) \leq \sum_{j=1}^m a_j f(v_j). \tag{A.9}$$

One key property of convex functions is the following.

Lemma 40 (Convex Functions and Continuity, see e.g. (Peypouquet, 2015, Prop. 3.5))**.** *Let $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ be a convex function. Then, f is continuous on the interior of its domain. In particular, a (real-valued) convex function $f : \mathcal{V} \to \mathbb{R}$ is continuous on the entire space $\mathcal{V}$.*

We now have all the ingredients necessary to introduce the set of functions

$$\Gamma_0(\mathcal{V}) = \Big\{ f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\} : f \text{ is proper, closed and convex} \Big\}. \tag{A.10}$$

We now recall the following definition, which is frequently used in the dissertation.

Definition 41 (ϵ-Subdifferential of a Function, see e.g. (Peypouquet, 2015, Sec. 3.4))**.** *Let $\epsilon \geq 0$. Then, the ϵ-subdifferential of $f \in \Gamma_0(\mathcal{V})$ at the point $v \in$* Dom$f$ *is the collection of the ϵ-subgradients at that point, namely,*

$$\partial_\epsilon f(v) = \Big\{ \alpha \in \mathcal{V} : f(v') \geq f(v) + \langle \alpha, v' - v \rangle - \epsilon, \textit{for any } v' \in \text{Dom} f \Big\}. \tag{A.11}$$

The standard subdifferential ∂f is retrieved from the above definition by setting $\epsilon = 0$. The following result is a direct consequence of the definition above and it links the concept of the ϵ-subdifferential of a function to the corresponding set of ϵ-minimizers.

Lemma 42 (Fermat Rule, see e.g. (Jean-Baptiste, 2010, Thm. 1.1.5))**.** *$\hat{v}_\epsilon \in \mathcal{V}$ is an ϵ-minimizer of $f \in \Gamma_0(\mathcal{V})$ if, and only if, $0 \in \partial_\epsilon f(\hat{v}_\epsilon)$.*

The behavior of the subdifferential of separable functions is described in the following.

Lemma 43 (Separable Functions and Subdifferential, see e.g. (Bauschke and Combettes, 2011, Prop. 16.8))**.** *Let $\mathcal{V}_1, \ldots, \mathcal{V}_m$ be Euclidean spaces. For any $v = (v_1, \ldots, v_m) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m$, let*

$$f(v) = \sum_{j=1}^{m} f_j(v_j), \tag{A.12}$$

with $f_j \in \Gamma_0(\mathcal{V}_j)$. Then, for any $v = (v_1, \ldots, v_m) \in \mathrm{Dom} f$, the subdifferential of f at v coincides with the following Cartesian product

$$\partial f(v) = \partial f_1(v_1) \times \cdots \times \partial f_m(v_m). \tag{A.13}$$

Before proceeding, we recall the definition of the Fenchel conjugate of a function.

Definition 44 (Fenchel Conjugate of a Function, see e.g. (Peypouquet, 2015, Sec. 3.6))**.** *Let $f \in \Gamma_0(\mathcal{V})$. Then, its Fenchel conjugate $f^* : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ is defined at $\alpha \in \mathcal{V}$ as*

$$f^*(\alpha) = \sup_{v \in \mathcal{V}} \langle v, \alpha \rangle - f(v). \tag{A.14}$$

In our proofs, we exploit the following standard properties of the conjugate function.

Lemma 45 (Fenchel Conjugate and Rescaling, see e.g. (Boyd and Vandenberghe, 2004, Sec. 3.3.2))**.** *Let $f \in \Gamma_0(\mathcal{V})$ and $c > 0$. Then, for any $\alpha \in \mathcal{V}$, $(cf)^*(\alpha) = cf^*(\alpha/c)$.*

Lemma 46 (Separable Functions and Fenchel Conjugate, see e.g. (Boyd and Vandenberghe, 2004, Sec. 3.3.2))**.** *Let $\mathcal{V}_1, \ldots, \mathcal{V}_m$ be Euclidean spaces. For any $v = (v_1, \ldots, v_m) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m$, let*

$$f(v) = \sum_{j=1}^{m} f_j(v_j), \tag{A.15}$$

with $f_j \in \Gamma_0(\mathcal{V}_j)$. Then, for any $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m$, we have

$$f^*(\alpha) = \sum_{j=1}^{m} f_j^*(\alpha_j). \tag{A.16}$$

Lemma 47 (Fenchel Conjugate and Monotonicity, see e.g. (Peypouquet, 2015, Prop. 3.50))**.** *Let $f_1, f_2 \in \Gamma_0(\mathcal{V})$ such that $f_1 \leq f_2$. Then, $f_1^* \geq f_2^*$.*

Lemma 48 (Young-Fenchel Inequality, see e.g. (Jean-Baptiste, 2010, Prop. 1.2.1))**.** *Let $f \in \Gamma_0(\mathcal{V})$ and consider $v \in \mathrm{Dom} f$. Then, $\alpha \in \partial_\epsilon f(v)$ if, and only if,*

$$f^*(\alpha) - \langle \alpha, v \rangle \leq -f(v) + \epsilon. \tag{A.17}$$

We now introduce a further definition which is used throughout this work.

Definition 49 (Lipschitz Function, see e.g. (Shalev-Shwartz and Ben-David, 2014, Def. 12.6))**.** *A function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ is L-Lipschitz (with $L > 0$) w.r.t. a norm $\|\cdot\|$ over $\mathcal{V}$ if, for any $v, v' \in \mathrm{Dom} f$,*

$$\left| f(v) - f(v') \right| \leq L \, \|v - v'\|. \tag{A.18}$$

The above definition implies the following bound on the dual norm of the subgradients.

Lemma 50 (Lipschitz Functions and Bounded Subgradients, see e.g. (Shalev-Shwartz and Ben-David, 2014, Lemma 14.7))**.** *Let $\|\cdot\|$ be a norm over $\mathcal{V}$ and let $\|\cdot\|_*$ be its dual. A function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ with open domain is L-Lipschitz w.r.t. $\|\cdot\|$ if, and only if, for any $v \in \mathrm{Dom} f$ and for any $\alpha \in \partial f(v)$, $\|\alpha\|_* \leq L$.*

Another definition we need is the following.

Definition 51 (Lipschitz Smooth Function)**.** *Let $\|\cdot\|$ be a norm over $\mathcal{V}$ and let $\|\cdot\|_*$ be its dual. A (real-valued) function $f : \mathcal{V} \to \mathbb{R}$ is β-Lipschitz smooth (with $\beta > 0$) w.r.t. $\|\cdot\|$ if it is differentiable and, for any $v, v' \in \mathcal{V}$, it holds that*

$$\left\| \nabla f(v) - \nabla f(v') \right\|_* \leq \beta \, \|v - v'\|. \tag{A.19}$$

The following result describes a well-known property of Lipschitz smooth functions.

Lemma 52 (Lipschitz Smooth Functions and Descent Lemma, see e.g. (Peypouquet, 2015, Lemma 1.30))**.** *Let $f : \mathcal{V} \to \mathbb{R}$ be a β-Lipschitz smooth function w.r.t. a norm $\|\cdot\|$ over $\mathcal{V}$. Then, for any $v, v' \in \mathcal{V}$,*

$$f(v') \leq f(v) + \langle \nabla f(v), v' - v \rangle + \frac{\beta}{2} \|v' - v\|^2. \tag{A.20}$$

Before proceeding, we strengthen the notion of convexity as follows.

Definition 53 (Strongly Convex Function, see e.g. (Peypouquet, 2015, Sec 2.3))**.** *A function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ is σ-strongly convex (with $\sigma > 0$) w.r.t. a norm $\|\cdot\|$ over $\mathcal{V}$ if, for any*

$t \in [0,1]$ *and any* $v, v' \in \mathrm{Dom} f$*,*

$$f(tv + (1-t)v') \leq tf(v) + (1-t)f(v') - \frac{\sigma}{2} t(1-t) \|v - v'\|^2. \tag{A.21}$$

The following two results describe two key properties of strongly convex functions.

Lemma 54 (Strongly Convex Functions and Minimizers, see e.g. (Peypouquet, 2015, Prop. 3.23))**.** *Let* $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ *be a proper, closed and* σ*-strongly convex function w.r.t. a norm* $\|\cdot\|$ *over* $\mathcal{V}$*. Then,* f *admits a minimizer over* $\mathcal{V}$ *and such a minimizer is unique.*

Lemma 55 (Strongly Convex Functions and Ascent Lemma, see e.g. (Peypouquet, 2015, Prop. 3.23))**.** *Let* $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ *be a* σ*-strongly convex function w.r.t. a norm* $\|\cdot\|$ *over* $\mathcal{V}$*. Then, for any* $v, v' \in \mathrm{Dom} f$ *and any* $\alpha \in \partial f(v)$*, we have*

$$f(v') \geq f(v) + \langle \alpha, v' - v \rangle + \frac{\sigma}{2} \|v' - v\|^2. \tag{A.22}$$

The following standard fact links the optimality w.r.t. the function values with the optimality w.r.t. the variables for a strongly convex function.

Lemma 56 (Strongly Convex Functions and Growth Condition)**.** *Let* $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ *be a proper, closed and* σ*-strongly convex function w.r.t. a norm* $\|\cdot\|$ *over* $\mathcal{V}$ *and denote by* $\hat{v}$ *its (exact) minimizer. Then, for any* $v \in \mathcal{V}$*,*

$$\frac{\sigma}{2} \|v - \hat{v}\|^2 \leq f(v) - f(\hat{v}). \tag{A.23}$$

Proof. We first note that, by Lemma 54, the minimizer of f in fact exists and it is unique. The statement immediately follows from applying Lemma 55 with $v = \hat{v}$, the minimizer of f, and $\alpha = 0 \in \partial f(\hat{v})$, thanks to Lemma 42. ■

We now give two key results for our proofs. The first one describes the duality between strong convexity and Lipschitz smoothness, the second one allows us to study the scaling effect on the Fenchel conjugate function.

Lemma 57 (Duality Between Strong Convexity and Lipschitz Smoothness, see e.g. (Kakade et al., 2009, Thm. 6), (Shalev-Shwartz and Kakade, 2009, Lemma 3))**.** *Let* $\|\cdot\|$ *be a norm over* $\mathcal{V}$ *and let* $\|\cdot\|_*$ *be its dual. Let* $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ *be a proper, closed and* σ*-strongly convex*

function w.r.t. $\|\cdot\|$. *Then,* f^* *is* $(1/\sigma)$*-Lipschitz smooth w.r.t.* $\|\cdot\|_*$. *Moreover, for any* $\alpha \in \mathcal{V}$,

$$\nabla f^*(\alpha) = \operatorname*{argmax}_{v \in \mathcal{V}} \langle \alpha, v \rangle - f(v) \in \mathrm{Dom} f. \tag{A.24}$$

Lemma 58 (Fenchel Conjugate and Scaling Effect, see e.g. (Shalev-Shwartz and Kakade, 2009, Lemma 4))**.** *Let* $\|\cdot\|$ *be a norm over* $\mathcal{V}$ *and let* $\|\cdot\|_*$ *be its dual. Let* $f \in \Gamma_0(\mathcal{V})$ *be a strongly convex function w.r.t.* $\|\cdot\|$ *and consider* $c_1, c_2 > 0$. *Then, for any* $\alpha \in \mathcal{V}$, *introducing the vector* $v_{c_2} = \nabla f^*(\alpha/c_2)$, *we have*

$$(c_2 f)^*(\alpha) - (c_1 f)^*(\alpha) = c_2 f^*(\alpha/c_2) - c_1 f^*(\alpha/c_1) \leq (c_1 - c_2) f(v_{c_2}). \tag{A.25}$$

We conclude by recalling the definition of proximity operator of a function and two well known properties that will be used for the experimental implementation.

Definition 59 (Proximity Operator, see e.g. (Bauschke and Combettes, 2011, Def. 12.23))**.** *For a function* $f \in \Gamma_0(\mathcal{V})$, *its proximity operator with parameter* $\gamma > 0$ *at the point* $\alpha \in \mathcal{V}$ *is defined as*

$$\mathrm{prox}_{\gamma f}(\alpha) = \operatorname*{argmin}_{v \in \mathcal{V}} \; f(v) + \frac{\gamma}{2} \|v - \alpha\|_2^2. \tag{A.26}$$

Lemma 60 (Separable Functions and Proximity Operator, see e.g. (Bauschke and Combettes, 2011, Prop. 23.30))**.** *Let* $\mathcal{V}_1, \ldots, \mathcal{V}_m$ *be Euclidean spaces. For any* $v = (v_1, \ldots, v_m) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m$, *let*

$$f(v) = \sum_{j=1}^{m} f_j(v_j), \tag{A.27}$$

with $f_j \in \Gamma_0(\mathcal{V}_j)$. *Then, for any* $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m$, $\mathrm{prox}_{\gamma f}(\alpha) \in \mathbb{R}^m$ *and, for any* $j \in \{1, \ldots, m\}$,

$$\Big(\mathrm{prox}_{\gamma f}(\alpha)\Big)_j = \mathrm{prox}_{\gamma f_j}(\alpha_j). \tag{A.28}$$

Lemma 61 (Moreau Identity, see e.g. (Bauschke and Combettes, 2011, Thm. 14.3))**.** *Consider* $f \in \Gamma_0(\mathcal{V})$ *and let* $f^* \in \Gamma_0(\mathcal{V})$ *its conjugate. Then, for any* $\alpha \in \mathcal{V}$ *and any* $\gamma > 0$, *we have*

$$\mathrm{prox}_{\gamma f^*}(\alpha) = \alpha - \gamma \, \mathrm{prox}_{f/\gamma}(\alpha/\gamma). \tag{A.29}$$

In the following section we briefly recall the main results we need from Fenchel Duality.

A.1 Fenchel Duality

For the content in this section, the reader can refer to (Peypouquet, 2015, Sec. 3.6.2). Given two Euclidean spaces $\mathcal{V}$ and $\mathcal{U}$, a linear operator $\mathcal{A} : \mathcal{V} \to \mathcal{U}$ and two functions $J \in \Gamma_0(\mathcal{V})$ and $G \in \Gamma_0(\mathcal{U})$, consider the primal problem

$$\hat{P} = \inf_{v \in \mathcal{V}} \; P(v) \qquad P(v) = G(\mathcal{A}v) + J(v). \tag{A.30}$$

The associated dual problem reads as follows

$$\hat{D} = \inf_{\alpha \in \mathcal{U}} \; D(\alpha) \qquad D(\alpha) = G^*(\alpha) + J^*(-\mathcal{A}^*\alpha), \tag{A.31}$$

where $\mathcal{A}^* : \mathcal{U} \to \mathcal{V}$ is the adjoint operator of $\mathcal{A}$ and G^* and J^* are the Fenchel conjugates of G and J, respectively. We recall also that the *duality gap* associated to two generic points $v \in \mathcal{V}$ and $\alpha \in \mathcal{U}$ is defined as

$$P(v) + D(\alpha). \tag{A.32}$$

It is well know that, for any $v \in \mathcal{V}$ and $\alpha \in \mathcal{U}$, the above quantity is always non-negative, i.e.

$$-D(\alpha) \leq P(v). \tag{A.33}$$

As a consequence, we have

$$\sup_{\alpha \in \mathcal{U}} \{-D(\alpha)\} = -\inf_{\alpha \in \mathcal{U}} \; D(\alpha) = -\hat{D} \leq \inf_{v \in \mathcal{V}} \; P(v) = \hat{P}. \tag{A.34}$$

The following proposition studies when the above inequality is in fact an equality.

Proposition 62 (Strong Duality, see e.g. (Peypouquet, 2015, Thm. 3.51))**.** *Consider the primal and the dual problems in Eq. (A.30) and Eq. (A.31). Assume that there exist a point $v \in \mathrm{Dom} J$ such that G is continuous at $\mathcal{A}v$ and assume that the primal problem in Eq. (A.30) admits a solution*

$$\hat{v} \in \operatorname*{argmin}_{v \in \mathcal{V}} \; P(v). \tag{A.35}$$

Then, the dual problem in Eq. (A.31) admits a solution

$$\hat{\alpha} \in \operatorname*{argmin}_{\alpha \in \mathcal{U}} \; D(\alpha). \tag{A.36}$$

Moreover, the following statements hold.

1. *Strong duality holds, namely,*

$$-\min_{\alpha\in\mathcal{U}} D(\alpha) = -D(\hat{\alpha}) = -\hat{D} = \min_{v\in\mathcal{V}} P(v) = \hat{P}(\hat{v}) = \hat{P}. \tag{A.37}$$

2. *The optimality conditions, also known as the Karush–Kuhn–Tucker (KKT) conditions, read as follows*

$$\hat{v} \in \partial J^*(-\mathcal{A}^*\hat{\alpha}) \qquad\qquad \hat{\alpha} \in \partial G(\mathcal{A}\hat{v}). \tag{A.38}$$

Appendix B

Primal-Dual Online Learning

In this appendix we recall the primal-dual Online Learning framework. Specifically, in App. B.1 we report some background material which is then used in the following App. B.2 for the proof of Thm. 2 in Sec. 3.2 in the main body. The material in this appendix is based on (Shalev-Shwartz and Kakade, 2009; Shalev-Shwartz and Singer, 2007a,b; Shalev-Shwartz et al., 2012).

Many online algorithms on a (primal) problem can be derived from the following primal-dual framework. At each iteration $m \in \{1, \dots, M\}$, $a)$ we define a pair of *instantaneous* primal-dual problems, $b)$ we update the dual variable according to an appropriate greedy coordinate descent procedure on the dual, $c)$ we update the new primal variable by evaluating the KKT conditions at the current dual variable. We now describe the above steps in detail. Throughout this appendix, we will let $\mathcal{V}$ be an Euclidean space endowed with a scalar product $\langle \cdot, \cdot \rangle$ and a generic norm $\| \cdot \|$ with dual $\| \cdot \|_*$.

a) The Primal and the Dual Problems. Regarding the first step, for any iteration $m \in \{1, \dots, M\}$, consider the primal problem of the following form as in Eq. (3.2)

$$\hat{P}_{m+1} = \inf_{v \in \mathcal{V}} P_{m+1}(v) \qquad P_{m+1}(v) = \sum_{j=1}^{m} g_j(A_j v) + c_m r(v), \tag{B.1}$$

where $c_m > 0$, $r \in \Gamma_0(\mathcal{V})$ is a σ_r-strongly convex function (with $\sigma_r > 0$) w.r.t. a norm $\| \cdot \|$ such that $\inf_{v \in \mathcal{V}} r(v) = 0$, for any $j \in \{1, \dots, M\}$, letting $\mathcal{V}_j$ an Euclidean space, $g_j \in \Gamma_0(\mathcal{V}_j)$ and $A_j : \mathcal{V} \to \mathcal{V}_j$ is a linear operator with adjoint A_j^*. Even though it is not necessary, to simplify the presentation, we set $P_1 \equiv 0$. Introducing the following linear operator

$$\mathcal{A}_m : \mathcal{V} \to \mathcal{V}_1 \times \cdots \times \mathcal{V}_m \qquad v \in \mathcal{V} \mapsto (A_1 v, \dots, A_m v) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m \tag{B.2}$$

and the function $G_m \in \Gamma_0(\mathcal{V}_1 \times \dots \mathcal{V}_m)$ defined, for any $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m$, as

$$G_m(\alpha) = \sum_{j=1}^{m} g_j(\alpha_j), \tag{B.3}$$

we can rewrite the problem in Eq. (B.1) as

$$\hat{P}_{m+1} = \inf_{v \in \mathcal{V}} P_{m+1}(v) \qquad P_{m+1}(v) = G_m(\mathcal{A}_m v) + c_m r(v). \tag{B.4}$$

Hence, according to what observed in App. A.1, exploiting the separability of G_m (see Lemma 46 in App. A), using the scaling properties of the conjugate (see Lemma 45 in App. A) and observing that the adjoint operator of $\mathcal{A}_m$ is give by

$$\mathcal{A}_m^* : \mathcal{V}_1 \times \cdots \times \mathcal{V}_m \to \mathcal{V} \qquad \alpha = (\alpha_1, \dots, \alpha_m) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m \mapsto \sum_{j=1}^{m} A_j^* \alpha_j \in \mathcal{V}, \tag{B.5}$$

the dual of the problem in Eq. (B.1) is given by

$$\hat{D}_{m+1} = \inf_{\alpha \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_m} D_{m+1}(\alpha) \qquad D_{m+1}(\alpha) = \underbrace{\sum_{j=1}^{m} g_j^*(\alpha_j)}_{G_m^*(\alpha)} + \underbrace{c_m r^*\left(-\frac{1}{c_m} \sum_{j=1}^{m} A_j^* \alpha_j\right)}_{(c_m r)^*(-\mathcal{A}_m^* \alpha)}, \tag{B.6}$$

where g_j^* and r^* represent the conjugate function of g_j and r, respectively. To simplify, we set also in this case $D_1 \equiv 0$. We observe that, when the above problems satisfy the assumptions in Prop. 62 in App. A, since the strong convexity of r is equivalent to the Lipschitz-smoothness of r^* (see Lemma 57 in App. A), denoting by $\hat{v}_{m+1}$ and $\hat{\alpha}_{m+1}$ a solution of the primal and the dual problem above, respectively, the corresponding KKT conditions read as follows

$$\hat{v}_{m+1} = \nabla r^*\left(-\frac{1}{c_m} \mathcal{A}_m^* \hat{\alpha}_{m+1}\right) \qquad \hat{\alpha}_{m+1} \in \partial G_m(\mathcal{A}_m \hat{v}_{m+1}), \tag{B.7}$$

where, more explicitly, we recall that

$$\mathcal{A}_m^* \hat{\alpha}_{m+1} = \sum_{j=1}^{m} A_j^* \hat{\alpha}_{m+1,j}. \tag{B.8}$$

We observe that, under the assumptions above, the primal objective P_{m+1} results to be proper, closed and strongly convex w.r.t. the norm $\|\cdot\|$. As a consequence, by Lemma 54 in App. A, we can in fact ensure the existence and the uniqueness of the primal solution $\hat{v}_{m+1}$.

We now are ready to describe the dual and the primal updating steps.

Algorithm 8 Primal-Dual Online Algorithm (more general version of Alg. 1 in Sec. 3.2)

Input $(g_m)_{m=1}^M$, $(A_m)_{m=1}^M$, $(c_m)_{m=1}^M$, $(\epsilon_m)_{m=1}^M$, r as described in the text

Initialization $\alpha_1 = (\,)$, $v_1 = \nabla r^*(0) \in \text{Dom } r$

For $m = 1$ to M

Receive g_m, A_m, c_{m+1}, ϵ_m

Suffer $g_m(A_m v_m)$ and compute $\alpha'_m \in \partial_{\epsilon_m} g_m(A_m v_m)$

Update α_{m+1} according to Eq. (B.9) by using α'_m

Define $v_{m+1} = \nabla r^*\Big(-\frac{1}{c_{m+1}} \mathcal{A}_m^* \alpha_{m+1}\Big) = \nabla r^*\Big(-\frac{1}{c_{m+1}} \sum_{j=1}^m A_j^* \alpha_{m+1,j}\Big) \in \text{Dom } r$

Return $(\alpha_m)_{m=1}^{M+1}$, $(v_m)_{m=1}^{M+1}$

b) c) The Updating Rules. The algorithm updates the dual variable α_{m+1} in a such way that, for a given parameter $\epsilon_m \geq 0$, there exist $\alpha'_m \in \partial_{\epsilon_m} g_m(A_m v_m)$ such that

$$D_{m+1}(\alpha_{m+1}) \leq D_{m+1}(\underbrace{\alpha_{m,1}\,,\ \ldots,\ \alpha_{m,m-1}}\,,\ \alpha'_m) = D_{m+1}(\underbrace{\alpha_m}\,,\ \alpha'_m). \tag{B.9}$$

The primal variable is then updated by the KKT conditions from the dual one. More precisely, following (Shalev-Shwartz and Singer, 2007b), in this last step we use a slightly different version of the KKT conditions in which we divide by c_{m+1} instead of c_m as in Eq. (B.7). For more details we refer to Alg. 8, which is a more general version of Alg. 1 given in the main body in Sec. 3.2. We also observe that, by definition, thanks to Lemma 57 in App. A, the primal variables $(v_m)_{m=1}^M$ generated by the algorithm are guaranteed to belong to Dom r.

Note that the requirement above about the dual update in Eq. (B.9) is satisfied (with the equality) by the update described in the main body $\alpha_{m+1} = (\alpha_m, \alpha'_m)$. As already underlined, the resulting primal algorithm coincides in this case with Follow The Regularized Leader applied to the linearized loss functions $v \mapsto \langle v, A_m^* \alpha'_m \rangle$. However, we stress that Eq. (B.9) is satisfied also by other more aggressive dual steps, including for example the one generating the primal Follow The Regularized Leader updating scheme applied to the original loss functions. We refer to (Shalev-Shwartz and Kakade, 2009; Shalev-Shwartz and Singer, 2007a,b; Shalev-Shwartz et al., 2012) for more details about this.

We finally conclude by observing that the framework above is a slightly different version of the standard primal-dual Online Learning setting described in the papers mentioned above. The differences in our presentation are the introduction of the linear operators $(A_m)_{m=1}^M$ inside the functions $(g_m)_{m=1}^M$ and the possibility to deal with an approximation of the subdifferential

$\partial g_m(A_m v_m)$. These two modifications will allow us to adapt the theory above to the Meta-Learning setting described in the main body.

B.1 Main Inequality on the Dual Gap

In the next proposition we study the behavior of the gap between two consecutive iterations on the dual objective for Alg. 8 (or Alg. 1). This statement will be the main tool used in App. B.2 in order to prove Thm. 2 in Sec. 3.2.

Proposition 63 (Dual Gap, see (Shalev-Shwartz, 2007, Lemma 1))**.** *Let* $(\alpha_m)_{m=1}^{M+1}$ *and* $(v_m)_{m=1}^{M+1}$ *be the iterates returned by Alg. 8 (or Alg. 1). Then,*

$$\Delta_1 = D_2(\alpha_2) - D_1(\alpha_1) \leq -g_1(A_1 v_1) + \frac{1}{2\sigma_r c_1} \left\| A_1^* \alpha_1' \right\|_*^2 + \epsilon_1. \tag{B.10}$$

Furthermore, for any $m \in \{2, \dots, M\}$*, we have*

$$\begin{aligned} \Delta_m &= D_{m+1}(\alpha_{m+1}) - D_m(\alpha_m) \\ &\leq - g_m(A_m v_m) + \frac{1}{2\sigma_r c_m} \left\| A_m^* \alpha_m' \right\|_*^2 + \epsilon_m \\ &\quad + c_m r^* \left(-\frac{1}{c_m} \mathcal{A}_{m-1}^* \alpha_m \right) - c_{m-1} r^* \left(-\frac{1}{c_{m-1}} \mathcal{A}_{m-1}^* \alpha_m \right). \end{aligned} \tag{B.11}$$

Proof. We first prove Eq. (B.10). Thanks to the updating rule in Eq. (B.9), the closed form of the dual objective in Eq. (B.6) and the definition $D_1 \equiv 0$, we can write

$$\Delta_1 = D_2(\alpha_2) - D_1(\alpha_1) = D_2(\alpha_2) \leq D_2(\alpha_1') = g_1^*(\alpha_1') + c_1 r^* \left(-\frac{1}{c_1} A_1^* \alpha_1' \right), \tag{B.12}$$

where $\alpha_1' \in \partial_{\epsilon_1} g_1(A_1 v_1)$ is the approximated subgradient used by Alg. 8 (or Alg. 1). But, thanks to Lemma 57 in App. A, the σ_r-strong convexity of r w.r.t. $\|\cdot\|$ is equivalent to the $(1/\sigma_r)$-Lipschitz smoothness of r^* w.r.t. $\|\cdot\|_*$, hence, applying Lemma 52 in App. A, exploiting the assumption $r^*(0) = \inf_{v \in \mathcal{V}} r(v) = 0$ and the definition of v_1 in Alg. 8 (or Alg. 1), we have

$$\begin{aligned} r^* \left(-\frac{1}{c_1} A_1^* \alpha_1' \right) &\leq r^*(0) - \frac{1}{c_1} \left\langle \nabla r^*(0), A_1^* \alpha_1' \right\rangle + \frac{1}{2\sigma_r c_1^2} \left\| A_1^* \alpha_1' \right\|_*^2 \\ &= -\frac{1}{c_1} \left\langle v_1, A_1^* \alpha_1^* \right\rangle + \frac{1}{2\sigma_r c_1^2} \left\| A_1^* \alpha_1' \right\|_*^2. \end{aligned} \tag{B.13}$$

Substituting in Eq. (B.12), we get the statement

$$\begin{aligned}\Delta_1 &\leq g_1^*(\alpha_1') + c_1 r^*\left(-\frac{1}{c_1}A_1^*\alpha_1'\right) \leq g_1^*(\alpha_1') - \langle v_1, A_1^*\alpha_1^*\rangle + \frac{1}{2\sigma_r c_1}\left\|A_1^*\alpha_1'\right\|_*^2 \\ &\leq -g_1(A_1 v_1) + \epsilon_1 + \frac{1}{2\sigma_r c_1}\left\|A_1^*\alpha_1'\right\|_*^2,\end{aligned} \tag{B.14}$$

where, in the last inequality, we have exploited the fact that $\alpha_1' \in \partial_{\epsilon_1} g_1(A_1 v_1)$ and Lemma 48 in App. A. We now prove the statement for $m \in \{2, \dots M\}$. By Eq. (B.9), the closed form of the dual objective in Eq. (B.6) and the rewriting

$$\mathcal{A}_m^*\alpha_{m+1} = \mathcal{A}_{m-1}^*\alpha_m + A_m^*\alpha_m', \tag{B.15}$$

with $\alpha_m' \in \partial_{\epsilon_m} g_m(A_m v_m)$ the approximated subgradient used by Alg. 8 (or Alg. 1), we have

$$\begin{aligned}\Delta_m &= D_{m+1}(\alpha_{m+1}) - D_m(\alpha_m) \leq D_{m+1}(\alpha_m\ ,\ \alpha_m') - D_m(\alpha_m) \\ &= g_m^*(\alpha_m') + c_m r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m - \frac{1}{c_m}A_m^*\alpha_m'\right) - c_{m-1} r^*\left(-\frac{1}{c_{m-1}}\mathcal{A}_{m-1}^*\alpha_m\right).\end{aligned} \tag{B.16}$$

Again, thanks to Lemma 57 in App. A, the σ_r-strong convexity of r w.r.t. $\|\cdot\|$ is equivalent to the $(1/\sigma_r)$-Lipschitz smoothness of r^* w.r.t. $\|\cdot\|_*$, hence, applying Lemma 52 in App. A and exploiting the definition of v_m in Alg. 8 (or Alg. 1), we have

$$\begin{aligned}&r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m - \frac{1}{c_m}A_m^*\alpha_m'\right) \\ &\leq r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m\right) - \frac{1}{c_m}\left\langle \nabla r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m\right), A_m^*\alpha_m'\right\rangle + \frac{1}{2\sigma_r c_m^2}\left\|A_m^*\alpha_m'\right\|_*^2 \\ &= r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m\right) - \frac{1}{c_m}\left\langle v_m, A_m^*\alpha_m'\right\rangle + \frac{1}{2\sigma_r c_m^2}\left\|A_m^*\alpha_m'\right\|_*^2.\end{aligned}$$

Substituting into Eq. (B.16), we can write the following

$$\begin{aligned}\Delta_m &\leq g_m^*(\alpha_m') + c_m r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m - \frac{1}{c_m}A_m^*\alpha_m'\right) - c_{m-1} r^*\left(-\frac{1}{c_{m-1}}\mathcal{A}_{m-1}^*\alpha_m\right) \\ &\leq g_m^*(\alpha_m') - \langle v_m, A_m^*\alpha_m'\rangle + \frac{1}{2\sigma_r c_m}\left\|A_m^*\alpha_m'\right\|_*^2 \\ &\qquad + c_m r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m\right) - c_{m-1} r^*\left(-\frac{1}{c_{m-1}}\mathcal{A}_{m-1}^*\alpha_m\right) \\ &\leq -g_m(A_m v_m) + \epsilon_m + \frac{1}{2\sigma_r c_m}\left\|A_m^*\alpha_m'\right\|_*^2 \\ &\qquad + c_m r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m\right) - c_{m-1} r^*\left(-\frac{1}{c_{m-1}}\mathcal{A}_{m-1}^*\alpha_m\right),\end{aligned}$$

where, in the last inequality, we have exploited the fact that $\alpha'_m \in \partial_{\epsilon_m} g_m(A_m v_m)$ and Lemma 48 in App. A. The last inequality above coincides with the desired statement. ■

B.2 Proof of Thm. 2

In this section, starting from the result described above in Prop. 63, we present the proof of Thm. 2 reported in the main body. More precisely, we provide the proof of a more general statement with a generic strong convexity parameter $\sigma_r > 0$ for the function r. For convenience of the reader, we restate Thm. 2 here. Finally, as the reader can immediately observe from the proof, the statement below holds for the more general Alg. 8, not only for Alg. 1.

Theorem 2 (Dual Optimality Gap for Alg. 1)**.** *Let $(v_m)_{m=1}^M$ be the primal iterates returned by the primal-dual online Alg. 1 when applied to the generic problem in Eq. (3.2) and let*

$$\Delta_{\text{Dual}} = D_{M+1}(\alpha_{M+1}) - \hat{D}_{M+1} \tag{3.4}$$

be the corresponding (non-negative) dual optimality gap at the last dual iterate α_{M+1}.

1. *If, for any $m \in \{1, \dots, M\}$, $c_{m+1} \geq c_m$, then,*

$$\Delta_{\text{Dual}} \leq -\sum_{m=1}^{M} g_m(A_m v_m) + \hat{P}_{M+1} + \frac{1}{2}\sum_{m=1}^{M} \frac{1}{c_m} \left\| A_m^* \alpha'_m \right\|_*^2 + \sum_{m=1}^{M} \epsilon_m.$$

2. *If, for any $m \in \{1, \dots, M\}$, $c_m = \sum_{j=1}^{m} \lambda_j$ for some $\lambda_j > 0$, then,*

$$\Delta_{\text{Dual}} \leq -\sum_{m=1}^{M} \Big\{ g_m(A_m v_m) + \lambda_m r(v_m) \Big\} + \hat{P}_{M+1} + \frac{1}{2}\sum_{m=1}^{M} \frac{1}{c_m} \left\| A_m^* \alpha'_m \right\|_*^2 + \sum_{m=1}^{M} \epsilon_m.$$

B.2.1 Proof of Thm. 2 Point 1.

In this subsection we prove the first point of Thm. 2, namely, the bound linking the optimality reached by the last dual iteration of Alg. 8 (or Alg. 1) to the cumulative error of the corresponding primal iterates.

Proof of Thm. 2 Point 1. We first show that, for any $m \in \{1, \dots, M\}$,

$$\Delta_m \leq -g_m(A_m v_m) + \frac{1}{2\sigma_r c_m} \left\| A_m^* \alpha'_m \right\|_*^2 + \epsilon_m. \tag{B.17}$$

As described in Prop. 63, the statement above in Eq. (B.17) holds for the case $m = 1$. For $m \in \{2, \dots, M\}$, we observe the following. Thanks to the choice of the increasing parameters $c_{m+1} \geq c_m$ and the non-negativity of r, according to Lemma 47 in App. A, we have

$$\begin{aligned} &c_m r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m\right) - c_{m-1} r^*\left(-\frac{1}{c_{m-1}}\mathcal{A}_{m-1}^*\alpha_m\right) \\ &= (c_m r)^*(-\mathcal{A}_{m-1}^*\alpha_m) - (c_{m-1} r)^*(-\mathcal{A}_{m-1}^*\alpha_m) \leq 0. \end{aligned} \tag{B.18}$$

Substituting this last inequality in Prop. 63, we get the statement in Eq. (B.17) for $m \in \{2, \dots, M\}$. Now, we observe that, thanks to the definition $D_1 \equiv 0$, we can write

$$D_{M+1}(\alpha_{M+1}) = \sum_{m=1}^{M} \Delta_m + D_1(\alpha_1) = \sum_{m=1}^{M} \Delta_m. \tag{B.19}$$

Thus, summing Eq. (B.17) over $m \in \{1, \dots, M\}$, we obtain that

$$D_{M+1}(\alpha_{M+1}) \leq -\sum_{m=1}^{M} g_m(A_m v_m) + \frac{1}{2\sigma_r} \sum_{m=1}^{M} \frac{1}{c_m} \left\| A_m^* \alpha'_m \right\|_*^2 + \sum_{m=1}^{M} \epsilon_m. \tag{B.20}$$

The desired statement now follows by summing to this last inequality the following relation

$$-\hat{D}_{M+1} \leq \hat{P}_{M+1}, \tag{B.21}$$

coinciding with the non-negativity of the duality gap in Eq. (A.34). ■

B.2.2 Proof of Thm. 2 Point 2.

In this subsection we prove the second point of Thm. 2, namely, the bound linking the optimality reached by the last dual iteration of Alg. 8 (or Alg. 1) to the *regularized* cumulative error of the corresponding primal iterates.

Proof of Thm. 2 Point 2. We first show that, for any $m \in \{1, \dots, M\}$,

$$\Delta_m \leq -\Big(g_m(A_m v_m) + \lambda_m r(v_m)\Big) + \frac{1}{2\sigma_r c_m} \left\| A_m^* \alpha'_m \right\|_*^2 + \epsilon_m. \tag{B.22}$$

Thanks to the definition $v_1 = \nabla r^*(0)$ in Alg. 8 (or Alg. 1), Lemma 57 in App. A and the assumption $\inf_{v\in\mathcal{V}} r(v) = 0$, we can write $r(v_1) = r(\nabla r^*(0)) = \inf_{v\in\mathcal{V}} r(v) = 0$. As a consequence, by Prop. 63, the above statement in Eq. (B.22) holds for the case $m = 1$. For any $m \in \{2, \dots, M\}$, introducing the notation $\lambda_{1:m} = \sum_{j=1}^{m} \lambda_j$, we can write

$$\begin{aligned} c_m r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m\right) - c_{m-1} r^*\left(-\frac{1}{c_{m-1}}\mathcal{A}_{m-1}^*\alpha_m\right) & \\ \leq (c_{m-1} - c_m)\, r\left(\nabla r^*\left(-\frac{1}{c_m}\mathcal{A}_{m-1}^*\alpha_m\right)\right) & \\ = (\lambda_{1:m-1} - \lambda_{1:m})\, r\left(\nabla r^*\left(-\frac{1}{\lambda_{1:m}}\mathcal{A}_{m-1}^*\alpha_m\right)\right) & \\ = (\lambda_{1:m-1} - \lambda_{1:m})\, r(v_m) = -\lambda_m r(v_m), & \end{aligned} \tag{B.23}$$

where, in the inequality we have applied Lemma 58 in App. A to $c_1 \rightsquigarrow c_{m-1}$, $c_2 \rightsquigarrow c_m$, $f \rightsquigarrow r$, $\alpha \rightsquigarrow -\mathcal{A}_{m-1}^*\alpha_m$, in the first equality we have introduced the definition of the parameter $c_m = \lambda_{1:m}$ and in the second equality we have exploited the definition of v_m in Alg. 8 (or Alg. 1). Substituting this last inequality in Prop. 63, we get the statement in Eq. (B.22) for $m \in \{2, \dots, M\}$. Now, we observe again that, thanks to the definition $D_1 \equiv 0$, we have

$$D_{M+1}(\alpha_{M+1}) = \sum_{m=1}^{M} \Delta_m + D_1(\alpha_1) = \sum_{m=1}^{M} \Delta_m. \tag{B.24}$$

Thus, summing Eq. (B.22) over $m \in \{1, \dots, M\}$, we obtain

$$D_{M+1}(\alpha_{M+1}) \leq -\left(\sum_{m=1}^{M} g_m(A_m v_m) + \lambda_m r(v_m)\right) + \frac{1}{2\sigma_r} \sum_{m=1}^{M} \frac{1}{\lambda_{1:m}} \left\|A_m^*\alpha_m'\right\|_*^2 + \sum_{m=1}^{M} \epsilon_m.$$

Also in this case, the desired statement follows by summing to this last inequality the following relation

$$-\hat{D}_{M+1} \leq \hat{P}_{M+1}, \tag{B.25}$$

coinciding the non-negativity of the duality gap in Eq. (A.34). ■

Appendix C

Experimental Details

In this chapter we provide some experimental details we skipped in the main body. We start from describing in App. C.1 how we tuned the hyper-parameters in our Lifelong Learning method in the statistical setting, after that, in App. C.2, we describe how we computed an approximation of the batch RERM inner algorithm in Eq. (2.24) and, finally, in App. C.3 we give some closed form expressions that we used for the implementation.

C.1 Hyper-Parameters Tuning in Lifelong Learning

Denote by $\bar{\theta}_{T,\lambda,\eta}$ the average of the meta-parameters computed with T iterations (hence T datasets and tasks) of our meta-algorithm with hyper-parameters λ and η. In all the experiments, we obtained this meta-parameter by learning it on a collection $\mathbf{Z}_{\rm tr}$ of $T_{\rm tr}$ *training* datasets (tasks), each comprising a dataset $Z_{\rm tr}$ of $n = n_{\rm tr}$ input-output pairs $z = (x, y) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. We performed this meta-training for different values of $\lambda \in \{\lambda_1, \ldots, \lambda_p\}$ and $\eta \in \{\eta_1, \ldots, \eta_r\}$ and we selected the best meta-parameter based on the prediction error measured on a separate set $\mathbf{Z}_{\rm va}$ of $T_{\rm va}$ *validation* datasets (tasks). Once such optimal λ and η values were selected, we reported the error of the corresponding estimator on a set $\mathbf{Z}_{\rm te}$ of $T_{\rm te}$ *test* datasets (tasks).

Note that the tasks in the test and validation sets $\mathbf{Z}_{\rm te}$ and $\mathbf{Z}_{\rm va}$ were all provided with a training inner dataset $Z_{\rm tr}$ of $n_{\rm tr}$ points and a test inner dataset $Z_{\rm te}$ of $n_{\rm te}$ points, both sampled from the same distribution. Indeed, in order to evaluate the performance of a meta-parameter θ, we needed first to train the corresponding algorithm A_θ on the training dataset $Z_{\rm tr}$, and then, to test the

performance of the resulting vector $A_\theta(Z_{\rm tr})$, by computing the empirical risk $\mathcal{R}_{Z_{\rm te}}(A_\theta(Z_{\rm tr}))$ on the test set $Z_{\rm te}$.

In addition to this, since we considered the online setting, the training datasets arrived one at the time, therefore model selection was performed *online*: the system kept track of all candidate values $\bar{\theta}_{T_{\rm tr},\lambda_j,\eta_k}$, $j \in \{1,\dots,p\}$, $k \in \{1,\dots,r\}$, and, whenever a new training task was presented, these meta-parameters were all updated by incorporating the corresponding new observations. The best meta-parameter θ was then returned at each iteration, based on its performance on the validation set $\mathbf{Z}_{\rm va}$, as explained before. The previous procedure describes how to tune simultaneously both λ and η. When the meta-parameter θ we used was fixed a priori (e.g. in ITL), we just needed to tune the hyper-parameter λ; in such a case the procedure was analogous to that one described above.

C.2 Approximating RERM by FISTA

In this section we describe how we applied FISTA (see (Beck and Teboulle, 2009, Sec. 4)) on the dual within-task problem in Eq. (3.10) in order to compute an approximation of the RERM algorithm in Eq. (2.24). We start from the setting outlined in Ex. 1 and then we focus on Ex. 2.

In the sequel, in order to simplify the presentation, we will exploit the following assumption which is, in fact, satisfied in all the experimental settings we considered in the main body.

Assumption 4 (Bounded Inputs)**.** *Let $\mathcal{X} \subseteq \mathcal{B}(0,R)$, where $\mathcal{B}(0,R) = \left\{x \in \mathbb{R}^d : \|x\|_2 \leq R\right\}$, with $R > 0$.*

C.2.1 Example 1. Variance

We start from recalling the (primal) RERM within-task problem in Eq. (3.9) for Ex. 1

$$\hat{w}_\theta = \operatorname*{argmin}_{w\in\mathbb{R}^d} \; P_{n+1}(w,\theta) \qquad P_{n+1}(w,\theta) = \sum_{i=1}^{n} \ell_i(\langle x_i, w\rangle) \; + \; \frac{\lambda n}{2}\,\|w-\theta\|_2^2 \tag{C.1}$$

and we rewrite its dual in Eq. (3.10) as follows

$$\hat{s}_\theta \in \operatorname*{argmin}_{s\in\mathbb{R}^n} \; D_{n+1}(s,\theta) \qquad D_{n+1}(s,\theta) = G(s) + J_\theta(s), \tag{C.2}$$

where, denoting by $X \in \mathbb{R}^{n \times d}$ the matrix containing as rows the input vectors $(x_i)_{i=1}^n$, we have introduced the following functions

$$G(s) = \sum_{i=1}^{n} \ell_i^*(s_i) \qquad J_\theta(s) = \frac{1}{2\lambda n} \left\| X^\top s \right\|_2^2 - \left\langle X^\top s, \theta \right\rangle. \tag{C.3}$$

We applied FISTA to this dual problem, treating J_θ as the smooth part and G as the non-smooth proximable part. More precisely, we observe that, thanks to Asm. 4, for any $\theta \in \mathbb{R}^d$, J_θ is (R^2/λ)-Lipschitz smooth w.r.t. the ℓ_2 norm. As a matter of fact, for any $s \in \mathbb{R}^n$, we have

$$\nabla J_\theta(s) = \frac{1}{\lambda n} XX^\top s - X\theta \tag{C.4}$$

and, consequently, for any $s_1, s_2 \in \mathbb{R}^n$, we can write

$$\begin{aligned} \left\| \nabla J_\theta(s_1) - \nabla J_\theta(s_2) \right\|_2 = \frac{1}{\lambda n} \left\| XX^\top (s_1 - s_2) \right\|_2 &\leq \frac{1}{\lambda n} \left\| XX^\top \right\|_\infty \|s_1 - s_2\|_2 \\ &\leq \frac{R^2}{\lambda} \|s_1 - s_2\|_2, \end{aligned} \tag{C.5}$$

where, the first inequality derives from the definition of operator norm and the second inequality is a consequence of Asm. 4, implying $\|XX^\top\|_\infty \leq nR^2$. The term G plays the role of the non-smooth part and, thanks to Lemma 60 in App. A, for any step size $\gamma > 0$, any $i \in \{1, \dots, n\}$ and any $s \in \mathbb{R}^n$,

$$\Big(\operatorname{prox}_{\gamma G}(s)\Big)_i = \operatorname{prox}_{\gamma \ell_i^*}(s_i), \tag{C.6}$$

where, by Lemma 61 in App. A, for any $a \in \mathbb{R}$,

$$\operatorname{prox}_{\gamma \ell_i^*}(a) = a - \gamma \operatorname{prox}_{\ell_i/\gamma}(a/\gamma). \tag{C.7}$$

The primal variable was defined from the dual one by evaluating the KKT conditions in Eq. (5.22) at the current dual iteration. The algorithm is reported in Alg. 9. In the following result we study the objective accuracy reached by the last dual iteration of Alg. 9.

Lemma 64 (Approximate Dual Solution by FISTA, Bias)**.** *Let Asm. 1, Asm. 3 and Asm. 4 hold. For $\theta \in \mathbb{R}^d$, consider the output $s_{\theta,K}$ of Alg. 9, coinciding with FISTA applied to the objective $D_{n+1}(\cdot, \theta)$ of the within-task dual problem of Ex. 1 in Eq. (C.2). Then, $s_{\theta,K}$ is an $\hat{\epsilon}_\theta$-minimizer of $D_{n+1}(\cdot, \theta)$, where*

$$\hat{\epsilon}_\theta = \frac{2L^2 R^2 n}{\lambda (K+1)^2}. \tag{C.8}$$

Algorithm 9 Approximation of RERM for Ex. 1

Input $\lambda > 0$, $\theta \in \mathbb{R}^d$, $Z = (z_i)_{i=1}^n$, $R > 0$ as described in the text

Initialization $s_{\theta,0} = p_{\theta,1} = 0 \in \mathbb{R}^n$, $t_1 = 1$

For $k = 1$ to K

Update $s_{\theta,k} = \text{prox}_{\gamma G}\left(p_{\theta,k} - \gamma \nabla J_\theta(p_{\theta,k})\right)$, where $\gamma = \lambda/R^2$

Define $w_{\theta,k} = -\dfrac{1}{\lambda n} X^\top s_{\theta,k} + \theta$

Update $t_{k+1} = \dfrac{1 + \sqrt{1 + 4t_k^2}}{2}$

Update $p_{\theta,k+1} = s_{\theta,k} + \dfrac{t_k - 1}{t_{k+1}} \left(s_{\theta,k} - s_{\theta,k-1}\right)$

Return $w_{\theta,K}$, $s_{\theta,K}$

Proof. Combining the upper bound R^2/λ on the Lipschitz-smoothness constant of J_θ (see above) and the upper bound $\|\hat{s}_\theta\|_2 \leq L\sqrt{n}$ (see at the end of the proof of Lemma 17) with the convergence rate in objective value for FISTA in (Beck and Teboulle, 2009, Thm. 4.4), we get

$$D_{n+1}(s_{\theta,K}, \theta) - D_{n+1}(\hat{s}_\theta, \theta) \leq \hat{\epsilon}_\theta, \tag{C.9}$$

where $\hat{\epsilon}_\theta$ is the value in Eq. (C.8). By definition of ϵ-minimizer, this last inequality coincides with the desired statement. ■

C.2.2 Example 2. Feature Map

We start from recalling the (primal) RERM within-task problem in Eq. (3.9) for Ex. 2

$$\hat{w}_\theta = \operatorname*{argmin}_{w \in \mathbb{R}^d} P_{n+1}(w, \theta) \qquad P_{n+1}(w, \theta) = \sum_{i=1}^n \ell_i(\langle x_i, w \rangle) + \frac{\lambda n}{2} \langle w, \theta^\dagger w \rangle + \iota_{\text{Ran}(\theta)}(w) \tag{C.10}$$

and we rewrite its dual in Eq. (3.10) as follows

$$\hat{s}_\theta \in \operatorname*{argmin}_{s \in \mathbb{R}^n} D_{n+1}(s, \theta) \qquad D_{n+1}(s, \theta) = G(s) + J_\theta(s), \tag{C.11}$$

where we have introduced the following quantities

$$G(s) = \sum_{i=1}^{n} \ell_i^*(s_i) \qquad J_\theta(s) = \frac{1}{2\lambda n} \left\| \theta^{1/2} X^\top s \right\|_2^2. \tag{C.12}$$

Also in this case, we applied FISTA to this dual problem, treating J_θ as the smooth part and G as the non-smooth proximable part. More precisely, proceeding as described above for Ex. 1, one can immediately note that, thanks to Asm. 4, for any $\theta \in \mathcal{S}$, J_θ is $(\|\theta\|_\infty R^2/\lambda)$-Lipschitz smooth w.r.t. the ℓ_2 norm and, for any $s \in \mathbb{R}^n$, its gradient is given by

$$\nabla F_\theta(s) = \frac{1}{\lambda n} X \theta X^\top s. \tag{C.13}$$

The algorithm is reported in Alg. 10, where, as before, the primal variable is updated from the dual one by the KKT in Eq. (6.19). Also in this case, in the following lemma, we study the objective accuracy reached by the last dual iteration of Alg. 10.

Lemma 65 (Approximate Dual Solution by FISTA, Feature Map)**.** *Let Asm. 1, Asm. 3 and Asm. 4 hold. For $\theta \in \mathcal{S}$, consider the output $s_{\theta,K}$ of Alg. 10, coinciding with FISTA applied to the objective $D_{n+1}(\cdot, \theta)$ of the within-task dual problem of Ex. 2 in Eq. (C.11). Then, $s_{\theta,K}$ is an $\hat{\epsilon}_\theta$-minimizer of $D_{n+1}(\cdot, \theta)$, where*

$$\hat{\epsilon}_\theta = \frac{2L^2 R^2 n \|\theta\|_\infty}{\lambda (K+1)^2}. \tag{C.14}$$

Proof. Combining the upper bound $R^2\|\theta\|_\infty/\lambda$ on the Lipschitz-smoothness constant of J_θ (see above) and the upper bound $\|\hat{s}_\theta\|_2 \leq L\sqrt{n}$ (see at the end of the proof of Lemma 25) with the convergence rate in objective value for FISTA in (Beck and Teboulle, 2009, Thm. 4.4), we get

$$D_{n+1}(s_{\theta,K}, \theta) - D_{n+1}(\hat{s}_\theta, \theta) \leq \hat{\epsilon}_\theta, \tag{C.15}$$

where $\hat{\epsilon}_\theta$ is the value in Eq. (C.14). By definition of ϵ-minimizer, this last inequality coincides with the desired statement. ■

We now make some observations, regarding the above implementations.

The results in Lemma 64 and Lemma 65 guarantee that, by applying the procedure described in Prop. 6 in Sec. 3.3 with the last dual FISTA iteration $s_{\theta,K+1}$, one can compute an $(\hat{\epsilon}_\theta/n)$-subgradient of the meta-objective $\mathcal{L}_Z$ at the point θ, where $\hat{\epsilon}_\theta$ is the value specified above. Thus, the corresponding approximation error on the meta-subgradients can be made negligible by just

Algorithm 10 Approximation of RERM for Ex. 2 by FISTA

Input $\lambda > 0$, $\theta \in \mathcal{S}$, $Z = (z_i)_{i=1}^n$, $R > 0$ as described in the text

Initialization $s_{\theta,0} = p_{\theta,0} = 0 \in \mathbb{R}^n$, $t_1 = 1$

For $k = 1$ to K

Update $s_{\theta,k} = \text{prox}_{\gamma G}\left(p_{\theta,k} - \gamma \nabla J_\theta(p_{\theta,k})\right)$, where $\gamma = \lambda / (\|\theta\|_\infty R^2)$

Define $w_{\theta,k} = -\dfrac{1}{\lambda n}\, \theta X^\top s_{\theta,k}$

Update $t_{k+1} = \dfrac{1 + \sqrt{1 + 4t_k^2}}{2}$

Update $p_{\theta,k+1} = s_{\theta,k} + \dfrac{t_k - 1}{t_{k+1}} \left(s_{\theta,k} - s_{\theta,k-1}\right)$

Return $w_{\theta,K}$, $s_{\theta,K}$

increasing the number of iterations K. Specifically, in the experiments reported in the main body, we ran Alg. 9 and Alg. 10 until the duality gap was lower than a prescribed tolerance.

We also observe that the implementation of FISTA described above is based on the use of the standard ℓ_2 norm in the computation of the proximity operator and the Lipschitz-smoothness or strong-convexity constants. In particular, by standard primal-dual arguments (see e.g. (Villa et al., 2013, Eq. (6.5) and Rem. 5)), it is possible to show that the following relation holds

$$\frac{\sigma(\theta)\lambda n}{2} \left\| w_{\theta,K} - \hat{w}_\theta \right\|_2^2 \leq D_{n+1}(s_{\theta,K}, \theta) - D_{n+1}(\hat{s}_\theta, \theta), \tag{C.16}$$

where $\sigma(\theta)\lambda n$ is the strong-convexity parameter of the primal within-task objective w.r.t. the ℓ_2 norm. As a consequence, dividing by $\sigma(\theta)\lambda n$, the dual objective converge rates given in Lemma 64 and Lemma 65 automatically translate into a convergence rate on the square approximation error of the corresponding primal variables. However, in the setting outlined in Ex. 2, the strategy described above yields to a convergence rate for the primal iterates in which the minimum non-zero eigenvalue of the matrix θ (coinciding with the strong convexity parameter $\sigma(\theta)$) appears at the denominator. We guess that this tedious dependency in the bound may be removed relying on more sophisticated primal-dual tools, such as the ones in (Dünner et al., 2016), or investigating about the application of FISTA with generic norms, as done for our OWO Meta-Learning framework described in Chpt. 3. However, since the convergence properties we observed in practice revealed to be satisfactory and comparable to those returned by the convex solver CVX, we decided to leave the investigation of this technical aspect for the future.

C.3 Closed Forms for the Implementation

At last, we report the closed forms for the conjugate, the subdifferential and the proximity operator of the absolute and the hinge loss used in our experiments.

Example 3 (Absolute Loss for Regression and Binary Classification)**.** *Let $\mathcal{Y} \subseteq \mathbb{R}$ or $\mathcal{Y} = \{\pm 1\}$. For any $\hat{y}, y \in \mathcal{Y}$, let $\ell(\hat{y}, y) = |\hat{y} - y|$ and denote $\ell_y(\cdot) = \ell(\cdot, y)$. Then, we have*

$$\partial \ell_y(\hat{y}) = \begin{cases} \{1\} & \text{if } \hat{y} - y > 0 \\ \{-1\} & \text{if } \hat{y} - y < 0 \\ [-1, 1] & \text{if } \hat{y} - y = 0. \end{cases} \tag{C.17}$$

Moreover, for any $y \in \mathcal{Y}$, ℓ_y is 1-Lipschitz, and, for any $u, a \in \mathbb{R}$, $\gamma > 0$, we have that

$$\ell_y^*(u) = \iota_{[-1,1]}(u) + uy \tag{C.18}$$

$$\operatorname{prox}_{\ell_y/\gamma}(a) = \begin{cases} a - 1/\gamma & \text{if } a - y > 1/\gamma \\ y & \text{if } a - y \in \left[-1/\gamma, 1/\gamma\right] \\ a + 1/\gamma & \text{if } a - y < -1/\gamma. \end{cases} \tag{C.19}$$

Example 4 (Hinge Loss for Binary Classification)**.** *Let $\mathcal{Y} = \{\pm 1\}$. For any $\hat{y}, y \in \mathcal{Y}$, let $\ell(\hat{y}, y) = \max\{0, 1 - y\hat{y}\}$ and denote $\ell_y(\cdot) = \ell(\cdot, y)$. Then, we have*

$$\partial \ell_y(\hat{y}) = \begin{cases} \{-y\} & \text{if } 1 - y\hat{y} > 0 \\ \{0\} & \text{if } 1 - y\hat{y} < 0 \\ [-1, 1]\{-y\} & \text{if } 1 - y\hat{y} = 0. \end{cases} \tag{C.20}$$

Moreover, for any $y \in \mathcal{Y}$, ℓ_y is 1-Lipschitz, and, for any $u, a \in \mathbb{R}$, $\gamma > 0$, we have that

$$\ell_y^*(u) = \frac{u}{y} + \iota_{[-1,0]}\left(\frac{u}{y}\right) \tag{C.21}$$

$$\operatorname{prox}_{\ell_y/\gamma}(a) = \begin{cases} a + y/\gamma & \text{if } ya < 1 - y^2/\gamma \\ 1/y & \text{if } ya \in \left[1 - y^2/\gamma, 1\right] \\ a & \text{if } ya > 1. \end{cases} \tag{C.22}$$

Bibliography

Pierre Alquier, The Tien Mai, and Massimiliano Pontil. Regret bounds for lifelong learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 261–269, 2017.

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

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008a.

Andreas Argyriou, Andreas Maurer, and Massimiliano Pontil. An algorithm for transfer learning in a heterogeneous environment. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 71–85. Springer, 2008b.

Andreas Argyriou, Massimiliano Pontil, Yiming Ying, and Charles A Micchelli. A spectral regularization framework for multi-task structure learning. In *Advances in neural information processing systems*, pages 25–32, 2008c.

Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Efficient representations for lifelong learning and autoencoding. In *Conference on Learning Theory*, pages 191–210, 2015.

Maria-Florina Balcan, Mikhail Khodak, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning*, pages 424–433, 2019.

Heinz H Bauschke and Patrick L Combettes. *Convex Analysis and Monotone Operator theory in Hilbert Spaces*, volume 408. Springer, 2011.

Jonathan Baxter. Theoretical models of learning to learn. In *Learning to Learn*, pages 71–94. 1998.

Jonathan Baxter. A model of inductive bias learning. *J. Artif. Intell. Res.*, 12(149–198):3, 2000.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Dimitri P Bertsekas, Angelia Nedi, and Asuman Ozdaglar. *Convex analysis and optimization*. Athena Scientific, 2003.

Jonathan M Borwein and Adrian S Lewis. Convex analysis and nonlinear optimization: theory and examples. 2010.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Brian Bullins, Elad Hazan, Adam Kalai, and Roi Livni. Generalize across tasks: Efficient algorithms for linear representation learning. In *Algorithmic Learning Theory*, pages 235–246, 2019.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

Giovanni Cavallanti, Nicolò Cesa-Bianchi, and Claudio Gentile. Linear algorithms for online multitask classification. *Journal of Machine Learning Research*, 11:2901–2934, 2010.

Nicolò Cesa-Bianchi and Claudio Gentile. Improved risk tail bounds for on-line algorithms. In *Advances in Neural Information Processing Systems*, pages 195–202, 2006.

Nicolò Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Carlo Ciliberto, Youssef Mroueh, Tomaso Poggio, and Lorenzo Rosasco. Convex learning of multiple tasks and their structure. In *International Conference on Machine Learning*, pages 1548–1557, 2015.

Giulia Denevi, Sara Garbarino, and Alberto Sorrentino. Iterative algorithms for a non-linear inverse problem in atmospheric lidar. *Inverse Problems*, 33(8):085010, 2017.

Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Incremental learning-to-learn with statistical guarantees. In *Proc. 34th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018a.

Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Learning to learn around a common mean. In *Advances in Neural Information Processing Systems*, pages 10190–10200, 2018b.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning*, pages 1566–1575, 2019a.

Giulia Denevi, Dimitris Stamos, Carlo Ciliberto, and Massimiliano Pontil. Online-within-online meta-learning. In *Advances in Neural Information Processing Systems*, pages 13089–13099, 2019b.

Aymeric Dieuleveut, Nicolas Flammarion, and Francis Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *The Journal of Machine Learning Research*, 18 (1):3520–3570, 2017.

Celestine Dünner, Simone Forte, Martin Takáč, and Martin Jaggi. Primal-dual rates and certificates. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 783–792. JMLR. org, 2016.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, 2004.

Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning*, pages 1920–1930, 2019.

Rishi Gupta and Tim Roughgarden. A pac approach to application-specific algorithm selection. *SIAM Journal on Computing*, 46(3):992–1017, 2017.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2016.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Laurent Jacob, Jean-philippe Vert, and Francis R Bach. Clustered multi-task learning: A convex formulation. In *Advances in neural information processing systems*, pages 745–752, 2009.

Hiriart-Urruty Jean-Baptiste. *Convex analysis and minimization algorithms: advanced theory and bundle methods*. SPRINGER, 2010.

Sham Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf*, 2(1), 2009.

Sham M Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808, 2009.

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

Abhishek Kumar and Hal Daumé III. Learning task grouping and overlap in multi-task learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1723–1730. Omnipress, 2012.

Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning*, pages 942–950, 2013.

Ilja Kuzborskij and Francesco Orabona. Fast rates by transferring from auxiliary hypotheses. *Machine Learning*, 106(2):171–195, 2017.

Peter J Lenk, Wayne S DeSarbo, Paul E Green, and Martin R Young. Hierarchical bayes conjoint analysis: Recovery of partworth heterogeneity from reduced experimental designs. *Marketing Science*, 15(2):173–191, 1996.

Nick Littlestone. From on-line to batch learning. In *Proceedings of the second annual workshop on Computational learning theory*, pages 269–284, 1989.

Andreas Maurer. Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6:967–994, 2005.

Andreas Maurer. Bounds for linear multi-task learning. *Journal of Machine Learning Research*, 7(Jan):117–139, 2006.

Andreas Maurer. Transfer bounds for linear feature learning. *Machine Learning*, 75(3):327–350, 2009.

Andreas Maurer, Massi Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. In *International Conference on Machine Learning*, 2013.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. An inequality with applications to structured sparsity and multitask dictionary learning. In *Conference on Learning Theory*, pages 440–460, 2014.

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

Andrew M McDonald, Massimiliano Pontil, and Dimitris Stamos. New perspectives on k-support and cluster norms. *Journal of Machine Learning Research*, 17(155):1–38, 2016.

Brendan Mcmahan and Matthew Streeter. No-regret algorithms for unconstrained online convex optimization. In *Advances in neural information processing systems*, pages 2402–2410, 2012.

H Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Conference on Learning Theory*, pages 1020–1039, 2014.

Charles A Micchelli, Jean M Morales, and Massimiliano Pontil. Regularizers for structured sparsity. *Advances in Computational Mathematics*, 38(3):455–489, 2013.

Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2014.

Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems*, pages 577–585, 2016.

Anastasia Pentina and Christoph Lampert. A PAC-Bayesian bound for lifelong learning. In *International Conference on Machine Learning*, pages 991–999, 2014.

Anastasia Pentina and Ruth Urner. Lifelong learning with weighted majority votes. In *Advances in Neural Information Processing Systems*, pages 3612–3620, 2016.

Juan Peypouquet. *Convex optimization in normed spaces: theory, methods and examples*. Springer, 2015.

Massimiliano Pontil and Andreas Maurer. Excess risk bounds for multitask learning with trace norm regularization. In *Conference on Learning Theory*, pages 55–76, 2013.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *I5th International Conference on Learning Representations*, 2017.

Paul Ruvolo and Eric Eaton. Ella: An efficient lifelong learning algorithm. In *International Conference on Machine Learning*, pages 507–515, 2013.

Paul Ruvolo and Eric Eaton. Online multi-task learning via sparse dictionary optimization. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

S Shalev-Shwartz. Online learning: theory, algorithms, and applications [ph. d. thesis]. *Hebrew Univ., Jerusalem*, 2007.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Shai Shalev-Shwartz and Sham M Kakade. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *Advances in Neural Information Processing Systems*, pages 1457–1464, 2009.

Shai Shalev-Shwartz and Yoram Singer. Convex repeated games and fenchel duality. In *Advances in neural information processing systems*, pages 1265–1272, 2007a.

Shai Shalev-Shwartz and Yoram Singer. Logarithmic regret algorithms for strongly convex repeated games. *The Hebrew University*, 2007b.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, 2009.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pages 71–79, 2013.

Peter Stange. On the efficient update of the singular value decomposition. In *PAMM: Proceedings in Applied Mathematics and Mechanics*, volume 8, pages 10827–10828. Wiley Online Library, 2008.

Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46, 1995.

Sebastian Thrun and Lorien Pratt. *Learning to Learn*. Springer, 1998.

Silvia Villa, Saverio Salzo, Luca Baldassarre, and Alessandro Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633, 2013.

Zhenxun Zhuang, Ashok Cutkosky, and Francesco Orabona. Surrogate losses for online learning of stepsizes in stochastic non-convex optimization. In *International Conference on Machine Learning*, pages 7664–7672, 2019.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.