Dimitri Meunier

3^{rd} year at ENSAE, MVA master
Master's Thesis
2019-2020

Meta Learning Meets Variational Inference

Learning Priors with Guarantees

CREST
Palaiseau, France

Supervisor at RIKEN: Pierre Alquier
Team Leader at RIKEN: Emtiyaz Khan
May 11, 2020 — November 13, 2020

Abstract

In modern supervised learning, there are domains like medical imaging or robotics where a large number of tasks is available but many of them are associated with a small amount of data. With few datapoints per task, learning them in isolation would give poor results. In this thesis, we consider the problem of learning from a sequence of regression or classification tasks with small sample size. By exploiting their similarities we seek to design algorithms that can utilize previous experience to rapidly learn new skills or adapt to new environments.

Inspired by human ingenuity in solving new problems by leveraging prior experience, **Meta Learning** is a subfield of Machine learning whose goal is to automatically adapt a learning mechanism from past experiences to rapidly learn new tasks with little available data. Since it "learns the learning mechanism" it is also referred to as *learning to learn*. Numerous formulations exist for Meta Learning and we focus on the problem of *sequential Meta Learning* or *Online Meta Learning* where the tasks arrive one at a time and the goal is to efficiently transfer information from the previous tasks to the new ones such that we learn the new tasks fast. Each task is in turn processed *online*. To sum up, we have a stream of tasks and for each task a stream of observations.

For each online task, diverse well-established algorithms exist, Perceptron, Online Ridge regression, Online Mirror Descent, Follow-The-Regularized-Leader, Bayesian Inference etc. We refer to them as the *within-task* algorithms. The big challenge is to design a meta mechanism that uses past experiences to adapt a within-task algorithm to perform better on the next tasks. Placing ourselves in a Bayesian context, we treat each inner task as a Bayesian problem and solve it using a brand-new tool from the approximate Bayesian inference toolbox: **Generalized Variational Inference**. The Bayesian principle gives us a natural way to transfer information between the tasks, the meta algorithm learns to adapt the **prior**. For the Bayesian community, a prior is a representation of the information we have on a specific task before seeing the data. Having multiple datasets, once a new dataset arrives we incorporate all the information we have from the previous tasks into the prior that serves to treat the new task. In practice, depending on the choice of prior and Variational Inference algorithm, it boils down to a strategy that automatically learns some hyperparameters e.g. a starting point or a learning rate.

In contrast to the increasing attention received by *deep Meta Learning*, we focus on simpler models — that is, convex models — to derive a theoretically grounded approach of Meta Learning. Exploiting the connections between Online Convex Optimisation and Online Variational Inference we show how to get a meta learning strategy guided by Bayesian principles that comes with a regret bound.

Acknowledgments

This internship was a partnership between ENSAE and the Approximate Bayesian Inference team at RIKEN, Tokyo. In these unusual times, it was entirely carried out in ENSAE offices while working remotely with the ABI team.

I am grateful to Emtiyaz Khan and Pierre Alquier who made the remote work as enjoyable as possible under those uncanny circumstances. I hope I meet you all soon in person Peter, Happy, Fariz, Gian Ma, Dharmesh, Siddharth, Thomas, Erik, Alexandre, Evgenii in Tokyo or elsewhere during conferences once the situation settles down.

I am especially thankful to Pierre who made this work possible and inspired me to continue along the research path. I am thankful for his constant stream of new ideas to push the project further and for his thorough reading of the thesis.

I also would like to thank my friends Pierre Delanoue and Clément Guillo with whom I spent incredible years of study. Finally, a tender thought for my family who has always supported me and to whom I have never been closer than in these troubled times. Jérémi, Tomi, you are extraordinary people.

Mathematical Notations and Abbreviations

Symbols

$\mathcal{X} \subseteq \mathbb{R}^p$	input space
$\mathcal{Y} \subseteq \mathbb{R}$	label space
$\mathcal{W} \subseteq \mathbb{R}^d$	decision space in online learning, equipped with a $\sigma-$algebra $\mathcal{T}$
$\mathcal{P}(\mathcal{W})$	set of all probability distributions on $(\mathcal{W}, \mathcal{T})$
$\ell : \mathcal{W} \to \mathbb{R}_+$	a loss, indexed by t in online learning and by t, i in the meta version
$Z \in (\mathcal{X} \times \mathcal{Y})^n$	a dataset (also called a task when equipped with a loss)
$\mathcal{R}_T$	regret after T steps

Operators

f^*	Fenchel transformation also known as convex conjugate
$\|.\|_*$	Dual norm

Abbreviations

OCO	Online Convex Optimisation
EWA	Exponentially Weighted Aggregation
OGD	Online Gradient Descent
OMD	Online Mirror Descent
NGD	Natural Gradient Descent
FTRL	Follow-The-Regularized-Leader
VI	Variational Inference
GVI	Generalized Variational Inference
NGVI	Natural Gradient Variational Inference

Contents

Chapter 1

Introduction

The field of Machine Learning traditionally focuses on learning a single task (on a single dataset). When a new task arrives, it is common to train another algorithm from scratch. Humans, on the other hand, do not learn to perform tasks in isolation, they leverage past experiences in order to learn new tasks as efficiently as possible. The idea of **Learning to Learn** was born to alleviate this gap Baxter (1998), Thrun and Pratt (2012). As the definitions differ from one author to the other we start by giving a definition of Learning to Learn also known as **Meta Learning**. Given a family of tasks, Meta Learning is the process of improving a learning algorithm with the number of tasks. In contrast, given a task, standard Machine Learning methods design the process of improving model predictions over multiple data instances Hospedales et al. (2020). Hence the term "meta", it is a supplementary layer on top of standard Machine Learning. In this context, the algorithm involved in achieving good accuracy on a specific task is called a **within-task algorithm** while the algorithm involved in adapting the within-task algorithm to perform well on a set of tasks is called the **meta algorithm**. The main difficulty is to design a *meta objective* to train the meta algorithm, a minimum requirement should be that the within-task algorithms learn quicker than if they have learned the tasks independently, i.e., we have adaptation. The ideal area of application for Meta Learning is when one has access to different datasets, more or less similar, that contain few observations each. Learning them in isolation should be a dead end as the number of observations is low. Taking advantage of the similarity between the tasks, Meta Learning should help to improve the performance of the entire learning system. Overall, the goal of Meta Learning is to achieve rapid adaptation and flexibility. It is seen as a critical problem for the future of Machine Learning Chollet (2019).

Until recently, the majority of the research work focused on the **Multi-Task Learning** problem and the **batch Meta Learning** problem. With the former, a finite number of tasks are presented in one batch to the learner. The latter is slightly different, it assumes that some tasks are independent and identically distributed from a fixed *meta task distribution*. The tasks are also treated in batch. For researchers interested in the development of algorithms that can learn like living beings, it might be unsatisfying as living beings learn from a constant stream of data, we say that they learn **online**. Online learning offers an appealing theoretical framework that takes into account the sequential and non-stationary aspects of learning problems Shalev-Shwartz et al. (2012). The set of works on the theoretical study of Meta Learning in the online setting is more recent and is the heart of the project. We focus on what is called the **Online-Within-Online** setting Denevi et al. (2019b): a stream of tasks are presented one at a time to the learning system and the data within each task are also processed sequentially. An alternative approach is

Online-Within-Batch where the tasks arrive sequentially but the data inside the tasks are treated in batch.

In this study, we motivate and present a principled and theoretically grounded meta strategy, where the meta learner uses past experiences to sequentially learn a good prior that allows the within-task learner to quickly adapt to the next tasks. While the search for theoretical guarantees forces us to consider simple — i.e. convex — models and does not translate directly to state-of-the-art models such as deep neural networks, we believe that the results should hold for non-convex models. Theoretical guarantees (regret bounds) for Online Meta Learning have been introduced in Alquier et al. (2017) where they learn a two-parts hierarchical representation with an Exponential Weighted Aggregation strategy. EWA can learn a posterior distribution in an online fashion but it requires to store all the datasets. Moreover, the convergence of the Monte Carlo algorithms used to compute it in practice is slow, especially in high dimension. It is therefore not practical for large models. A frequentist approach has been suggested by Denevi et al. (2019b) (as a generalisation of Denevi et al. (2019a)). Exploiting a Primal-Dual view of Online Mirror Descent they propose a bi-level online optimisation scheme that is fast and computationally cheap. The approach is close to Finn et al. (2019) where the authors suggest a strategy to learn a good starting point for a gradient descent algorithm with Meta Learning, this is also called *fine-tuning*.

The aim of the thesis is to use an approximation of EWA called **Generalized Variational Inference** to obtain a strategy that will combine the computational efficiency of Denevi et al. (2019b) and the advantages of Bayesian posterior as in Alquier et al. (2017) and this with theoretical guarantees. Variational Inference Khan and Nielsen (2018), Khan and Rue (2019) is known to perform approximate Bayesian Inference at scale and works extremely well in practice Khan et al. (2018). It led to the first scaling of Bayesian principles to state-of-the-art neural networks Osawa et al. (2019). The difficulty is to show that we can adapt and extend the analysis of Denevi et al. (2019b) to the variational setting. By employing a Variational approach instead of a frequentist one we learn a distribution over the inner parameters and a prior common to the tasks, we can therefore perform uncertainty estimations. Intuitively, with Bayesian Inference, a prior is used to incorporate all the information we have on the problem before seeing the data. In Meta Learning, we have several tasks and it is natural to ask how we could incorporate this knowledge as prior information for the new tasks. Learning to calibrate the prior with Meta Learning helps the inner algorithms to converge faster, especially if we have a large number of tasks with few observations. In practice, our method is more general than the fine-tuning approach as it can learn other hyperparameters such as learning rates.

We notice that there are slightly different learning settings close to Meta Learning. The term **few-shot learning** can be equivalent to Meta-Learning but is usually associated with deep Computer Vision of Neural Language Processing tasks. The goal is to discriminate between N classes with K examples of each, K is typically small. In the extreme case of **one-shot learning**, $K = 1$. A recent example of large scale few-shot learning experiment is the GPT-3 model trained with 175 billion parameters Brown et al. (2020). Meta Learning mainly focuses on learning new tasks fast but can suffer from catastrophic forgetting Kirkpatrick et al. (2017), the goal of **continual learning** Pan et al. (2020) is to overcome this limitation and train algorithms that can maintain expertise on tasks which they have not experienced for a long time. Finally, Meta Learning also shares similarities with **transfer learning**, **lifelong learning** and **federated learning**.

1.1 Notations for Online Learning and Meta Learning

In this section, we introduce the Online Convex Optimisation (OCO) framework, the notion of regret and its meta generalisation. The main advantages of online learning over standard statistical inference is that it does not make any assumption on the data such as independence or stationarity. It relies on convex optimisation and therefore makes instead assumptions on smoothness and strong convexity. The obtained regret bounds can be converted to statistical guarantees under the standard statistical assumptions with the *online-to-batch* technique.

Convex Online Optimisation The goal of online learning is to learn how to make good decisions given a stream of data. The stream of data is denoted as $Z = (z_i)_{i=1}^n$. We consider a loss $\ell(z_i, w) \in \mathbb{R}^+, w \in \mathcal{W}$ where $\mathcal{W} \subseteq \mathbb{R}^d$ is a convex decision space. We use the shorthand $\ell_i : w \mapsto \ell(z_i, w)$ and assume that ℓ_i is convex for all $i \in \{1, \dots, n\}$. At each step, the decision is denoted by w_i and is a function of $(z_j)_{j=1}^{i-1}$. $\ell_i(w_i)$ measures how good is the decision w_i at step i. For example, in classification or regression, we consider an input space $\mathcal{X}$, an output space $\mathcal{Y}$, and $Z = (x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ is the set of input-label pairs. The decision space is the parameter space of a prediction function f_w. In the former $\ell_i(w) = (1 - y_i f_w(x_i))_+$ is the hinge loss and in the latter $\ell_i(w) = ||y_i - f_w(x_i)||_2^2$ is the mean squared error loss. We can consider other settings such as Prediction With Expert Advice, Online ranking, unsupervised learning etc. The dataset Z is processed *sequentially*, at each step $i \in \{1, \dots, n\}$:

(a) a datapoint z_i is observed

(b) the algorithm outputs a decision $w_i \in \mathcal{W}$

(c) the learner incurs the error $\ell_i(w_i)$

We refer to the algorithm used to solve this problem as a **within-task algorithm**. In online learning, the notion of generalisation error is not valid, the natural metric to assess the performance of an algorithm is the **regret**:

$$\mathcal{R}_{inner}(Z) := \sum_{i=1}^{n} \ell_i(w_i) - \inf_{w \in \mathcal{W}} \sum_{i=1}^{n} \ell_i(w). \tag{1.1}$$

It is called a "regret" as it measures how much we regret not having followed the best decision in hindsight. The first term in the right-hand side is the cumulative error incurred by the algorithm. The algorithm should be chosen so that it admits a low regret on the task, i.e. a regret in $O(n)$.

Online Meta Learning Online Meta Learning is a two layers generalisation of online learning. We consider a series of tasks indexed by $t = 1, \cdots, T$: $Z_t = (z_{t,i})_{i=1}^n$. For each dataset, the input space, the output space and the loss are the same. We also assume

for simplicity that each dataset contains the same number of points. To solve each task, we select a within-task algorithm among a family of algorithms parametrised by a meta parameter $\theta \in \Theta$. The goal is to adapt θ to the sequence of learning tasks. The sequential process mimics the standard online learning process where at each step, the loss is the cumulative loss suffered by the current within-task algorithm. We use the shorthand $\ell_{t,i} : w \mapsto \ell(z_{t,i}, w)$ to highlight the link with the task t and the piece of information i inside this task. For each task $t \in \{1, \dots, T\}$,

(a) the meta-learner incrementally receives a task dataset Z_t

(b) it runs the within-task online algorithm with meta-parameter θ_t on Z_t, returning the predictor vectors $\left(w_i^{\theta_t}\right)_{i=1}^n$

(c) it incrementally incurs the errors $\ell_{t,i}\left(w_i^{\theta_t}\right)$, $i = 1, \dots, n$

(d) the meta-parameter (and consequently, the within-task algorithm) is updated to θ_{t+1}

The overall process is illustrated in Figure 1.1. The algorithm we use to select the meta parameter is called the **meta algorithm**. Its performance is evaluated by the **meta regret**:

$$\mathcal{R}_{meta}\left((Z)_{t=1}^T\right) := \sum_{t=1}^T \sum_{i=1}^n \ell_{t,i}\left(w_i^{\theta_t}\right) - \inf_{w_1,\dots,w_T \in \mathcal{W}} \sum_{t=1}^T \sum_{i=1}^n \ell_{t,i}(w_t). \tag{1.2}$$

For illustration, let us consider a simple example, we have a sequence of linear regression tasks with true parameters $\theta_1^*, \dots, \theta_T^*$ and their parameters are close to each other, e.g. there exists an Euclidean ball with center θ^* and radius r such that for all $t \in [T]$, $\theta_t^* \in B(\theta^*, r)$. The smaller the radius, the closer the regression tasks are to each other. It might be easier to learn a common "bias vector" i.e. the center θ^* than learning the regression tasks individually. If we use a Stochastic Gradient Descent to solve the regression tasks, we can initiate the descent at the bias vector and it should converge faster. This bias vector can be seen as a hyperparameter for the linear regression solver and the question we want to answer is: how to learn θ^*? Solving this problem in high dimension in the context of personalised feature selection for medicine setups has been recently used Yamada et al. (2017).

1.2 Outline of the Thesis

The remainder is organised as follows. In Chapter 2, we present the class of within-task algorithms we use inside the meta process, the Generalized Variational Inference (GVI) family. The necessary background on Variational Inference (VI) and Online Convex Optimisation (OCO) is introduced. We present GVI and connect it to the OCO framework to derive a first regret bound for the within-task algorithms. In Chapter 3, we derive a general meta strategy and its meta regret bound. We formalise the idea of learning a prior and present a solution based on a Primal-Dual analysis of Online Learning. Chapter 4 introduces a practical algorithm in the Gaussian setting which boils down to a bi-level gradient descent where the meta level learns the starting point and the learning rate (coordinate-wise) for the inner descent. We discuss future work and conclude the thesis in Chapter 5.

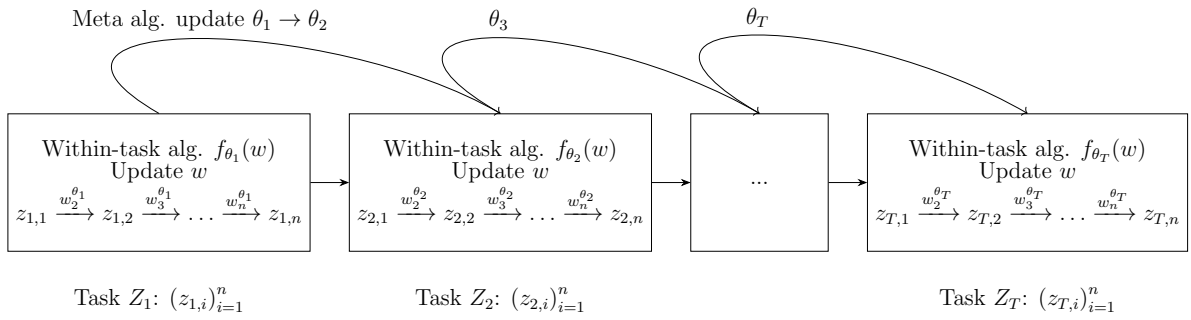

Figure 1.1: Illustration of the meta process for regression or classification. $\{f_\theta(w) : \theta \in \Theta, w \in \mathcal{W}\}$ is a class of within-task predictors parametrised with meta parameters θ and inner parameters w. For all $\theta \in \Theta$ and $w \in \mathcal{W}$, $f_\theta(w) : \mathcal{X} \to \mathcal{Y}$. The meta learner updates the meta parameters $\theta \in \Theta$ such that the inner parameters can be trained to quickly achieve good performance on any task.

.

Chapter 2

Generalized Variational Inference for the within-tasks

Many algorithms are available to solve a machine learning task in the online setting: Exponentially Weighted Aggregation (EWA), Online Gradient Descent (OGD), Follow-The-Regularized-Leader (FTRL), Online Mirror Descent (OMD) and Variational Inference. EWA has the advantage that it can sequentially build a posterior distribution over the parameters of the model rather than just computing a point estimate. However, it is not tractable for non-finite parameter spaces and necessitates the use of MCMC algorithms that can be slow in high dimension or when the sample size is large. Variational Inference is an approximation of EWA that can scale well with the dimension while retaining convergence guarantees in both the batch setting Alquier et al. (2016), Alquier et al. (2020) and the online setting Chérief-Abdellatif et al. (2019), Alquier (2020). For those reasons, scalability and computation of an approximate posterior distribution, VI will be the main algorithm of study in this thesis and this chapter is devoted to its introduction. We will show how we can build a regret bound for Online VI by treating it is a special case of Follow-The-Regularized-Leader and studying its Online Mirror Descent counterpart as done in Shalev-Shwartz et al. (2012).

In the first section, we start by introducing the EWA strategy as a generalization of the standard Bayesian posterior in statistical modelling. Then, we present Variational Inference as a tractable approximation of EWA and how it can be further extended by allowing another regularisation than the Kullback-Leibler divergence. This last version has been recently investigated under the name Generalized Variational Inference (GVI) Knoblauch et al. (2019), Alquier (2020) and is the one used in our algorithms.

In the second section, we connect GVI to two standard algorithms in Online Convex Optimisation (OCO): FTRL and OMD. This connection was done in Chérief-Abdellatif et al. (2019) and led to the first regret bound for Online Variational Inference. We conclude with the inner algorithm that we will apply on the within-tasks and give its regret bound. The meta-algorithm will be the focus of the next chapter.

2.1 Generalized Variational Inference

2.1.1 General Bayes (EWA)

In this chapter, we consider a single task and drop the dependency in t in all the manipulated variables. We consider a sequence of losses $\ell_i : \mathcal{W} \to \mathbb{R}$ for $i = 1, \dots, n$ as in the introductory chapter. Instead of learning a fix parameter w_i at each step we are interested in the problem of learning a posterior distribution ρ_i. Our prediction can then be made drawing $w_i \sim \rho_i$. The **Exponentially Weighted Aggregation** (EWA) strategy starts from some prior π on the parameter space $\mathbb{R}^d$ (for simplicity we assume that $\mathcal{W} = \mathbb{R}^d$) and updates a probability distribution at each online round,

$$\begin{aligned} \rho_1(dw) &= \pi(dw) \\ \rho_i(dw) &= \frac{\exp\left(-\frac{1}{\lambda}\sum_{j=1}^{i-1}\ell_j(w)\right)\pi(dw)}{\int_{\mathbb{R}^d}\exp\left(-\frac{1}{\lambda}\sum_{j=1}^{i-1}\ell_j(v)\right)\pi(dv)}, \quad i = 2, \dots, n. \end{aligned} \tag{2.1}$$

This strategy is also called *Generalized Bayes* and ρ_i a *generalized posterior distribution* (also called *Gibbs posterior*). It is called "generalized" as we do not assume the existence of an underlying statistical model. Therefore, ρ_i might not be a posterior distribution as defined in the standard Bayesian framework. Given a statistical model, i.e. a family of densities $\{p_w, w \in \mathbb{R}^d\}$ with respect to some reference measure ν on some space $\mathcal{X}$, and i.i.d random variables $X_1, X_2, \dots$, drawn from some probability distribution p on $\mathcal{X}$[1], if $\lambda = 1$[2] and the loss functions are $\ell_i(w) = -\log p_w(X_i)$, ρ_i is the posterior distribution of w given $X_1, ..., X_{i-1}$ used in Bayesian statistics:

$$\rho_i(w) \propto \prod_{j=1}^{i-1} p_w(X_j)\pi(w), \quad i = 2, \dots, n. \tag{2.2}$$

While we do not assume that such a statistical model exists we still refer to π as a "prior distribution", to ρ_i as a "posterior distribution" and to our approach as "Bayesian". The expected regret of EWA is in $O(\sqrt{n})$ under the assumption that the losses ℓ_i are uniformly bounded. If in addition the ℓ_i are convex we get a uniform regret bound also in $O(\sqrt{n})$. For the sake of completeness, both bounds are stated in the Appendix A.2, Theorem A.1. Despite this guarantee, in practice, computing the normalising constant in eq. (2.1) can be challenging. For large models, it necessitates to run MCMC algorithms known to be slow in high dimension. The Online Meta Learning problem has been previously treated with just EWA in Alquier et al. (2017) but they suffer a high computational cost and running the algorithm requires to store all the previous datasets.

[1] If p is not in $\{p_w, w \in \mathbb{R}^d\}$ the model is mis-specified but the likelihood still exists.

[2] If $\lambda \neq 1$, it is called a tempered likelihood, and it might be more robust to mis-specification than the standard likelihood Bhattacharya et al. (2019).

2.1.2 From EWA to Variational Inference

We saw in the previous section that EWA is a first solution to approximate a posterior distribution in the online setting. The main difficulty being the computation of a potentially high dimensional integral. The next lemma shows how to convert this integration problem into an optimisation one Catoni (2004), Guedj (2019).

Lemma 2.1. *(Donsker-Varadhan representation) Equation (2.1) can be rewritten as,*

$$\rho_i = \underset{\rho \in \mathcal{P}(\mathbb{R}^d)}{\operatorname{argmin}} \left\{ \sum_{j=1}^{i-1} \mathbb{E}_{w\sim\rho} \left[\ell_j(w)\right] + \lambda KL(\rho, \pi) \right\}, \tag{2.3}$$

where $\mathcal{P}(\mathbb{R}^d)$ is the set of all probabilities on $\mathbb{R}^d$ and KL is the Kullback-Leibler divergence (Definition A.5).

Proof. Let $\rho \in \mathcal{P}(\mathbb{R}^d)$ such that ρ is absolutely continuous with respect to π,

$$\begin{aligned}
\mathbb{E}_{w\sim\rho}\left[\sum_{j=1}^{i-1}\ell_j(w)\right] + \lambda KL(\rho,\pi) &= \lambda \mathbb{E}_{w\sim\rho}\left[\frac{1}{\lambda}\sum_{j=1}^{i-1}\ell_j(w) + \log\left(\frac{\rho(w)}{\pi(w)}\right)\right] \\
&= \lambda \mathbb{E}_{w\sim\rho}\left[\log\left(\frac{\rho(w)}{\pi(w)e^{-\frac{1}{\lambda}\sum_{j=1}^{i-1}\ell_j(w)}}\right)\right] \\
&= \lambda \mathbb{E}_{w\sim\rho}\left[\log\left(\frac{\rho(w)}{\rho_i(w)}\right)\right] - \underbrace{\log\int_{\mathbb{R}^d} e^{-\frac{1}{\lambda}\sum_{j=1}^{i-1}\ell_j(v)}\pi(dv)}_{:=c} \\
&= \lambda KL(\rho,\rho_i) - c.
\end{aligned}$$

Where c does not depend on ρ. Since $KL(\rho,\rho_i) \geq 0$ and $KL(\rho,\rho_i) = 0$ if and only if $\rho = \rho_i$ (Proposition A.5), the unique distribution which achieves the minimum of the right-hand side in eq. (2.3) is the Gibbs posterior given by eq. (2.1). □

Unfortunately, solving the optimisation problem in eq. (2.3) is usually as infeasible as computing the normalisation constant in eq. (2.1). The idea of **Variational Inference** is to approximate ρ_i by constraining eq. (2.3) to a tractable set of probability distributions $\mathcal{Q} \subset \mathcal{P}(\mathbb{R}^d)$. Then, $\hat{\rho}_i \approx \rho_i$ where,

$$\hat{\rho}_i = \underset{\rho \in \mathcal{Q}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{i-1} \mathbb{E}_{w\sim\rho} \left[\ell_j(w)\right] + \lambda KL(\rho, \pi) \right\}. \tag{2.4}$$

The richer the family, the closest it is to the full EWA update. If ρ_i belongs to the variational family we retrieve the same update. If the family of distributions is "simple" in the sense that the problem is tractable and the derived algorithms lead to closed form updates we can efficiently perform approximate Bayesian updates. Different choices for $\mathcal{Q}$ lead to a wide range of different VI variants. The two most used are *mean field VI* and *parametric VI* (see Chapter 10 in Bishop (2006)). In the former, we assume that the

distributions in $\mathcal{Q}$ factorize with respect to predefined groups, in the latter, we consider an identifiable parametric family $\mathcal{Q} = \{q_\mu, \mu \in \mathcal{M} \subseteq \mathbb{R}^p\}$ and we approximate ρ_i with q_{μ_i} where,

$$\mu_i = \underset{\mu \in \mathcal{M}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{i-1} \mathbb{E}_{w \sim q_\mu} \left[\ell_j(w) \right] + \lambda KL(q_\mu, \pi) \right\}. \tag{2.5}$$

In Chapter 4, we will consider a combination of both: *Gaussian Mean Field VI*. For the remainder of the thesis we only consider parametric VI eq. (2.5). Despite the approximation, for parametric VI under convexity assumptions, we will see later that a regret bound similar than the one for EWA can be proved.

We have already taken two steps away from standard Bayesian inference: first, the ℓ_i's do not necessarily come from a statistical model, secondly, we do not minimize on the entire set of probability distributions. In the next section, we take a third step away by considering other divergences than the Kullback-Leibler divergence.

2.1.3 Generalized Variational Inference

The use of a different divergence than the Kullback-Leibler divergence for EWA and its variational approximation has been suggested at several places, Reid et al. (2015), Alquier and Guedj (2018), Knoblauch et al. (2019), and recently studied in the context of Online Learning in Alquier (2020). If there is no underlying statistical model, there is no obvious reason to stick to the Kullback-Leibler divergence. For example, in the standard EWA algorithm, the regret bound holds under the assumption that the losses are uniformly bounded, but using a chi-2 divergence, this restriction is substituted with the assumption that the second moments under π of the ℓ_i's are uniformly bounded, which is far less restrictive (Example 2.2 in Alquier (2020)). For us, the use of a different divergence will be necessary to get a practical algorithm in Chapter 4 as the KL divergence does not satisfy all the required hypothesis regarding convexity.

Instead of the KL term we consider any divergence D, that is, a function of ρ and π with values in $[0, +\infty]$, satisfying: $D(\rho\|\pi) \geq 0$ with equality if and only if $\rho = \pi$, and when ρ is not absolutely continuous with respect to π, $D(\rho\|\pi) = +\infty$. The following generalization of the EWA strategy,

$$\rho_i = \underset{\rho \in \mathcal{P}(\mathbb{R}^d)}{\operatorname{argmin}} \left\{ \sum_{j=1}^{i-1} \mathbb{E}_{w \sim \rho} \left[\ell_j(w) \right] + \lambda D(\rho, \pi) \right\}, \tag{2.6}$$

admits a regret bound if D satisfies a strongly convex condition and the ℓ_i's a Lipschitz condition in expectation (Alquier (2020), Theorem 2.1). In a similar way to what has been done previously with Variational Inference, we consider a **Generalised Variational**

Inference (GVI) update where we replace the KL term by any divergence D.

$$\mu_i = \underset{\mu \in \mathcal{M}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{i-1} \mathbb{E}_{w \sim q_\mu} \left[\ell_j(w) \right] + \lambda D(q_\mu, \pi) \right\}. \tag{2.7}$$

We stress that the three parts that now differ from standard Bayes are the divergence D which might not be KL, the loss ℓ which might not be the negative log-likelihood of a statistical model and the set $\mathcal{Q}$ over which we minimise. For a fixed prior and a fixed sequence of losses, the algorithm is fully characterised by the triplet $(\ell, \mathcal{Q}, D)$. This idea of Generalised Variational Inference can also be found in Knoblauch et al. (2019) where they call it the *rule of three*. The different combinations are summarized in fig. 2.1. Even if we are not doing standard Bayesian inference, we continue to call π a prior and q_{μ_i} a posterior distribution.

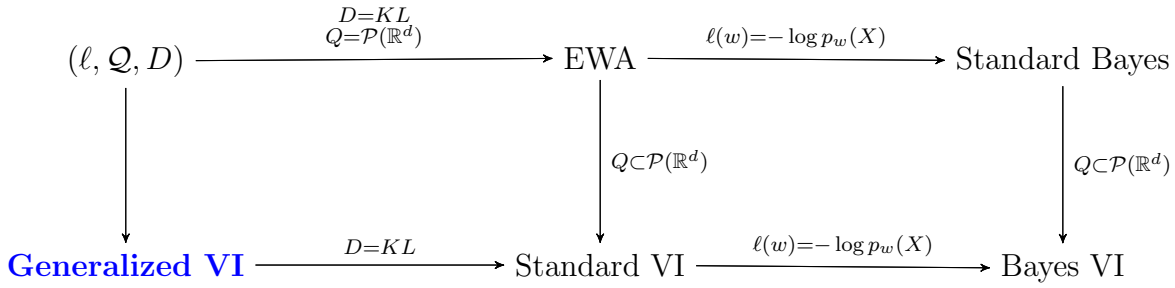

Figure 2.1: Summary of the different combinations of the rule of three $(\ell, \mathcal{Q}, D)$. Inspired by Knoblauch et al. (2019).

As a last generalisation, instead of the expectation operator we consider any convex operator L_i that acts on the distribution μ and depends on the loss ℓ_i.

$$\mu_i = \underset{\mu \in \mathcal{M}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{i-1} L_j(\mu) + \lambda D(q_\mu, \pi) \right\}. \tag{2.8}$$

This will allow us to consider either $L_i(\mu) = \mathbb{E}_{w \sim q_\mu} \left[\ell_i(w) \right]$ or $L_i(\mu) = \ell_i \left(\mathbb{E}_{w \sim q_\mu} \left[w \right] \right)$ in Chapter 4. Keep in mind that at the end, we want to apply the tools of Online Convex Optimisation to our problem, therefore, the sequence of losses $(\ell_i)_{i \geq 1}$ is replaced by the sequence $(L_i)_{i \geq 1}$, and instead of predicting $(w_i)_{i \geq 1}$, we get $(\rho_i)_{i \geq 1}$ and then predict $w_i \sim \rho_i$. Comparatively, knowing ρ_i, we can assess the underlying uncertainty of the process. The goal of the next section is to formally connect Generalized Variational Inference to standard algorithms in Online Convex Optimisation. This connection leads to a practical algorithm to solve eq. (2.8) and a regret bound.

2.2 GVI through the lenses of Online Convex Optimisation

We briefly review the Follow-The-Regularized-Leader (FTRL) algorithm in the Online Convex Optimisation setting (OCO) and how it is linked to Online Mirror Descent (OMD)

through linearisation following the monograph Shalev-Shwartz et al. (2012). For more details about Mirror Descent in the batch setting we refer to the monographs Bubeck (2015), Vishnoi (2020) and for FTRL and Online Mirror Descent in the OCO setting we refer to the monographs Hazan (2019), Orabona (2019). Those tools are important because Generalized Variational Inference can be casted as a special case of FTRL and therefore inherits its theoretical analysis.

2.2.1 FTRL and GVI

We consider an arbitrary sequence of **convex** losses $g_1, g_2 \ldots; g_i : \mathcal{V} \to \mathbb{R}_+$ on an Euclidean space $\mathcal{V}$. It does not matter what is behind g_i as long as it is convex. We can have $g_i = \ell_i$, $g_i = L_i$ as in GVI or it can be a meta loss as we will see in the next chapter. Given a regulariser $r : \mathcal{V} \to \mathbb{R}_+$, the FTRL strategy solves for each $i \geq 1$,

$$v_i = \arg\min_{v \in \mathcal{V}} \sum_{j=1}^{i-1} g_i(v) + \lambda r(v). \tag{2.9}$$

Without the regularized term r, this strategy is called Follow-The-Leader. It is a natural learning rule that predicts at any online round the parameter which has minimal loss on all past rounds. However, this strategy can fail if the updates are too unstable (Example 2.2 Shalev-Shwartz et al. (2012)). Therefore, the regularization term is here to stabilize the solution. To do so, r has to be strongly convex (definition A.1).

The similarity with eq. (2.8) is striking. GVI is a special case of FTRL where $g_i = L_i$ and $r : \mu \to D(q_\mu, \pi)$. Therefore, a regret bound for FTRL is valid for GVI. The main difficulty is to show under which conditions L_i is convex and r is strongly convex. Chérief-Abdellatif et al. (2019) shows that it holds for location-scale variational families and the Kullback-Leibler divergence if the ℓ_i are convex (see also Domke (2019)). Regret bounds for FTRL are well known but solving eq. (2.9) requires to solve an optimisation program at **each step** and might be intractable in practice. In the next section, we show that we can simplify the problem by linearising it. The resulting algorithm is equivalent to an Online Mirror Descent. It leads to both a practical algorithm and a regret bound for GVI.

2.2.2 Link to Online Mirror Descent through linearisation

If the g_i are differentiable we have, for all $(v, v') \in \mathcal{V}^2$,

$$g_i(v) = g_i(v') + \langle v - v', \nabla_{v=v'} g_i(v) \rangle + o\left(v - v'\right). \tag{2.10}$$

Keeping only the linear part that depends on v, the suggested linearized update is[3]

$$v_i = \underset{v \in \mathcal{V}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{i-1} \left\langle v, \nabla_{v=v_j} g_j(v) \right\rangle + \lambda r(v) \right\}. \tag{2.11}$$

We refer to this update as **Linear-FTRL**. It is still a FTRL strategy but the convex losses $(g_i)_{i\geq 1}$ are substituted with the linear losses $\hat{g}_i(v) = \langle v, \nabla_{v=v_i} g_i(v) \rangle$. In other words, each g_i is approximated by a linear loss that is its Taylor expansion of order 1 at the prediction point v_i. Since the g_i's are convex, for all $i \geq 1$, and for all $v \in \mathcal{V}$,

$$g_i(v_i) - g_i(v) \leq \langle v_i, \nabla_{v=v_i} g_i(v) \rangle - \langle v, \nabla_{v=v_i} g_i(v) \rangle = \hat{g}_i(v_i) - \hat{g}_i(v).$$

Therefore, the regret produced by running Linear-FTRL on the sequence of losses $(\hat{g}_i)_{i\geq 1}$ upper bounds the regret on the sequence $(g_i)_{i\geq 1}$. Let us note that if the g_i are just convex but not differentiable the argument still holds if we take subgradients (Definition A.2).

Another important point that motivates the use of Linear-FTRL is that it is equivalent to **Online Mirror Descent**. Indeed, from the definition of the convex conjugate (Definition A.3),

$$\begin{aligned} v_i &= \underset{v \in \mathcal{V}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{i-1} \left\langle v, \nabla_{v=v_j} g_j(v) \right\rangle + \lambda r(v) \right\} \\ &= \underset{v \in \mathcal{V}}{\operatorname{argmax}} \left\{ \left\langle v, -\frac{1}{\lambda} \sum_{j=1}^{i-1} \nabla_{v=v_j} g_j(v) \right\rangle - r(v) \right\} \\ &= \nabla r^* \left(-\frac{1}{\lambda} \sum_{j=1}^{i-1} \nabla_{v=v_j} g_j(v) \right). \end{aligned} \tag{2.12}$$

In the last equality we used the fact that r is strongly convex (hence strictly convex) and Proposition A.3. It can we re-written as follows, $v_1 = \nabla r^*(0)$ and for $i = 1, \ldots,$

$$\begin{cases} \alpha_i = \nabla_{v=v_i} g_i(v) \\ v_{i+1} = \nabla r^* \left(-\frac{1}{\lambda} \sum_{j=1}^{i} \alpha_j \right) \end{cases}, \tag{2.13}$$

or equivalently, $\alpha_0 = 0$, $v_1 = \nabla r^*(0)$ and for $i = 1, \ldots,$

$$\begin{cases} \alpha_i = \alpha_{i-1} - \frac{1}{\lambda} \nabla_{v=v_i} g_i(v) \\ v_{i+1} = \nabla r^* \left(\alpha_i \right) \end{cases}. \tag{2.14}$$

It shows that Linear-FTRL is equivalent to Online Mirror Descent where the mirror map that goes from the primal space to the dual space is ∇r. $(v_i)_{i\geq 1}$ is the primal sequence and $(\alpha_i)_{i\geq 1}$ is the dual sequence. We stress that there are other ways to introduce Online

[3] We still call the update v_i so as not to make the notations more cumbersome but be aware that v_i in eq. (2.11) is not the same as in eq. (2.9).

Mirror Descent, for example with Bregman divergences or Legendre functions. They lead to slightly different versions. The version presented here is specifically called *lazy Online Mirror Descent*

Strategy for GVI. Applying lazy OMD for GVI eq. (2.8) gives us a practical algorithm. Once we know ∇r^*, at each online round, it only requires the computation of a gradient (or a subgradient if the losses are not differentiable). The resulting strategy is given in Algorithm 1. Let us emphasize that it does not solve eq. (2.8) directly but instead it solves a linearized version of it that is equivalent to Online Mirror Descent. Regret bound for Online Mirror Descent (i.e. Linear-FTRL) are well known and can be found in Shalev-Shwartz et al. (2012). In Chérief-Abdellatif et al. (2019) they were the first to apply the regret bounds from Online Mirror Descent to Variational Inference by treating it as an Online Convex Optimisation problem. More recently, Alquier (2020) generalised this analysis to Generalized Variational Inference as stated in the following theorem.

Theorem 2.1 (adaptation of Theorem 4.2 Alquier (2020))**.** *Let $\|\cdot\|$ be a norm on $\mathbb{R}^d$. If each L_i is convex and L-Lipschitz with respect to $\|\cdot\|$, if $\psi : \mu \mapsto D\left(q_\mu \| \pi\right)$ is α-strongly convex with respect to $\|\cdot\|$ (it implies that ψ^* is differentiable), then:*

$$\sum_{i=1}^{n} L_i(\mu_i) \leq \inf_{\mu \in \mathcal{M}} \left\{ \sum_{i=1}^{n} L_i(\mu) + \frac{nL^2}{\lambda \alpha} + \lambda D\left(q_\mu, \pi\right) \right\}, \tag{2.15}$$

where $(\mu_i)_{i=1}^n$ is the output of Algorithm 1. If $L_i(\mu) = \mathbb{E}_{w \sim q_\mu}\left[\ell_i(w)\right]$ and ℓ_i is convex for all $i \geq 1$, the cumulative regret has the following upper bound when $\hat{w}_i = \mathbb{E}_{w \sim q_{\mu_i}}[w]$ is the posterior mean,

$$\sum_{i=1}^{n} \ell_i\left(\hat{w}_i\right) \leq \inf_{\mu \in \mathcal{M}} \left\{ \sum_{i=1}^{n} \mathbb{E}_{w \sim q_\mu}\left[\ell_i(w)\right] + \frac{nL^2}{\lambda \alpha} + \lambda D(q_\mu, \pi) \right\}. \tag{2.16}$$

Algorithm 1: Lazy OMD (a.k.a Linear-FTRL) for Generalized Variational Inference

Init: $\psi : \mu \mapsto D\left(q_\mu \| \pi\right)$, $\mu_1 = \nabla \psi^*(0)$;

Loop for $i \leftarrow 1, \ldots, n$ do

Receive L_i, suffer $L_i(\mu_i)$

$\alpha_i \in \partial L_i(\mu_i)$

$\mu_{i+1} = \nabla \psi^* \left(-\frac{1}{\lambda} \sum_{j=1}^{i} \alpha_j \right)$

Return: $(\mu_i)_{i=1}^{n+1}$

Conclusion of the chapter. We have introduced Generalized Variational Inference for the within-tasks and how it can be solved with a particular Online Mirror Descent scheme. In the convex setting, Online Mirror Descent has a known regret bound. In the next chapter, we will see how Online Mirror Descent can be adapted to run a meta algorithm. To achieve this goal, we will present a different view of Online Mirror Descent based on the Fenchel-Rockafellar duality.

Chapter 3

Primal-Dual Online Learning and application to the Meta task

In the online Meta Learning framework introduced in Chapter 1, the role of the meta algorithm is to select a within-task algorithm for each new task. Our choice for the within-task algorithms is the Generalized Variational Inference family introduced in the last chapter. For a GVI strategy, once the variational family and the divergence are fixed the remaining degree of freedom is the choice of the prior. Therefore, we suggest a meta algorithm that will select a prior for each task. It is natural to incorporate information from one task to the other through a prior. For the Bayesian community, a prior is a representation of the information we have on a specific task before seeing the data. With Meta Learning, we have multiple datasets, and once a new dataset arrives we should incorporate all the information we have from the previous datasets into the prior that serves to treat the new task.

In the first section, we formalise the idea of learning a prior with a meta algorithm. In the second section, we present a Primal-Dual view of Online Learning that unifies many algorithms such as FTRL and OMD, it serves as a building block to get our meta strategy. Finally, in the last section, we give the general meta algorithm and its regret bound.

3.1 Meta Variational Inference

To derive a meta algorithm we start with a parametric family of priors $\{\pi_\theta, \theta \in \Theta\}$. Our strategy is to apply a second online algorithm at the meta level to sequentially learn θ. To achieve this goal, we define a sequence of meta losses $(\mathcal{L}_t : \Theta \to \mathbb{R}_+)_{t\geq 1}$. We introduce again the double index $\{t, i\}$ where t refers to the task t and i the datapoint inside that task[1]. Our meta objective is the following,

$$\mathcal{L}_t(\theta) = \min_{\mu\in\mathcal{M}} \frac{1}{n}\sum_{i=1}^{n} L_{t,i}(\mu) + \lambda D(q_\mu, \pi_\theta), \qquad \theta \in \Theta. \tag{3.1}$$

[1]For simplicity we have assumed that each task contains the same number of points n.

$\mathcal{L}_t(\theta)$ is the best regularized variational regret loss we could get after solving the problem in hindsight with prior π_θ. It measures how good is π_θ for task t. The use of the "min" operator will be properly justified in the next section. The meta loss is inspired by the meta learning "bias" framework suggested in Denevi et al. (2019b), that is in turn inspired by previous work on Multi-Task Learning. Our objective is to apply an online algorithm (specifically another Online Mirror Descent) to adapt θ to the sequence of losses $(\mathcal{L}_t)_{t\geq 1}$.

In the remainder we will answer the following question: when is it possible to run Online Mirror Descent on $(\mathcal{L}_t)_{t\geq 1}$? There are two difficulties, preserving the convexity of $\mathcal{L}_t$ and building subgradients.

(a) The theoretical guarantees for Online Mirror Descent lie on convexity. While not obvious to get for $\mathcal{L}_t$, a sufficient condition is that the divergence in jointly convex in (μ, θ).

(b) To run Online Mirror Descent we need a first-order oracle i.e. we need a procedure that given any point $\theta \in \Theta$ and step $t \geq 1$ outputs a subgradient of $\mathcal{L}_t$ at θ. We do not have access to a zeroth order oracle since the within-task algorithm might not achieve the min. However, following the analysis done in Denevi et al. (2019b), we show that it is possible to build **approximate subgradients** using the "dual" outputs of the within-task algorithm. The approximation is known in closed form and incorporated into the meta regret bound. The strategy exploits a different analysis of Online Mirror Descent than the one presented in the last section. It is based on the Fenchel-Rockafellar duality and is presented in the next section.

Let us stress that eq. (3.1) does not defined a variational program, there is no expectation. We do not learn a "meta distribution" but instead directly treat the parameters of the prior as meta parameters we want to learn.

3.2 Primal-Dual Online Learning: unified view of OMD and FTRL

In this section, we take a step back from the Meta Learning problem and present an analysis of Online Convex Optimisation through the prism of duality: **Primal-Dual Online Learning**. Due to Shalev-Shwartz and Singer (2007c), Shalev-Shwartz and Kakade (2009), Shalev-Shwartz and Singer (2007b), Shalev-Shwartz and Singer (2007a), it has been adapted by Denevi et al. (2019b) to suggest a frequentist meta learning strategy. Our contribution is to show that it can also be adapted for Generalized Variational Inference. In comparison to the linearization approach of the last chapter, the advantages of the Primal-Dual approach are the following,

(a) it relates multiple algorithms such as Follow-The-Regularized-Leader and Online Mirror Descent through the notion of aggressiveness under a single algorithm with known regret bound. While not essential for our framework as we will still use Online Mirror Descent at the end, it leaves the door open for more aggressive optimisation schemes in between FTRL and OMD.

(b) it allows the use of approximate subgradients instead of subgradients. This is important as we do not have access to exact subgradients for the meta losses. Let us note that it should be easy to add this feature to the linearized approach too.

(c) it provides dual gap bounds from which regret bounds like Theorem 2.1 are a consequence. The dual gap bound is the key element for the construction of approximate subgradients of the meta losses and is the main reason why we cannot only rely on the approach of the last chapter.

Since the analysis is not specific to Variational Inference we introduce an arbitrary OCO problem, using the similar notations as eq. (2.9): $g_1, \dots, g_M$ is a sequence of losses indexed by m and defined over an Euclidean space $\mathcal{V}$, c_m is the regularisation strength at step m and r is a regularizer. For all $m \geq 1$,

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

The associated Follow-The-Regularized-Leader update is $v_{m+1} = \arg\min_{v \in \mathcal{V}} \hat{P}_{m+1}(v)$. The corresponding Fenchel dual to eq. (3.2) is,

$$D_{m+1} = \inf_{\alpha \in \mathcal{V}^m} \hat{D}_{m+1}(\alpha) \quad \hat{D}_{m+1}(\alpha) = \sum_{j=1}^{m} g_j^*(\alpha_j) + c_m r^* \left(-\frac{1}{c_m} \sum_{j=1}^{m} \alpha_j \right). \tag{3.3}$$

We call the two program the *instantaneous Primal-Dual pair at step m*. Weak duality always holds: $P_m + D_m \geq 0$ for all $m \geq 1$. For more details about Fenchel duality, we refer the reader to Peypouquet (2015). We make the following standard assumptions where $\|\cdot\|$ is a norm on $\mathcal{V}$,

Assumption 1: $c_m > 0$, r is a s-strongly convex ($s > 0$), proper and closed function on $\mathcal{V}$ w.r.t. $\|\cdot\|$ such that $\inf_{v \in \mathcal{V}} r(v) = 0$.

Assumption 2: for $m \in \{1, \dots, M\}$, g_m is a closed, convex and real-valued function on $\mathcal{V}$.

Theorem 3.1 characterises strong duality for the instantaneous Primal-Dual pair.

Theorem 3.1 (Denevi et al. (2019b) Prop. 32 and Lemma 34)**.** *Under Assumptions 1 and 2, both the primal and the dual admit a solution and strong duality holds, for all $m \geq 1$: $P_m + D_m = 0$. We denote by $\hat{v}_m \in \mathcal{V}$ the primal variable solution of P_m and by $\hat{\alpha}_m \in \mathcal{V}^{m-1}$ the dual variable solution of D_m. Strong duality implies that*

$$P_m = \sum_{j=1}^{m-1} g_j(\hat{v}_m) + c_m r(\hat{v}_m) = \sum_{j=1}^{m-1} g_j^*(\hat{\alpha}_{m,j}) + c_m r^* \left(-\frac{1}{c_m} \sum_{j=1}^{m-1} \hat{\alpha}_{m,j} \right) = D_m. \tag{3.4}$$

$\hat{v}_m$ and $\hat{\alpha}_m$ are related through the following optimality conditions,

$$\hat{v}_m = \nabla r^* \left(-\frac{1}{c_m} \sum_{j=1}^{m-1} \hat{\alpha}_{m,j} \right) \qquad \hat{\alpha}_m \in \partial \left(G_{m-1} \right)(A_{m-1}(\hat{v}_m)) \,, \tag{3.5}$$

where,

$$G_{m-1}(\alpha) = \sum_{j=1}^{m-1} g_j \left(\alpha_j \right), \quad \forall \alpha \in \mathcal{V}^{m-1} \text{ and } A_{m-1}(v) = (v, \dots, v) \in \mathcal{V}^{m-1} \quad \forall v \in \mathcal{V}$$

.

Proof. (eq. (3.2) - eq. (3.3)) is an example of Fenchel-Rockafellar primal-dual problem. A classical result (Theorem 3.51 in Peypouquet (2015)) states that if the primal admits a minimiser $\hat{v}_m$ and there exists $v \in dom(r)$ such that the g_m's are continuous at v, then the dual admits a minimiser $\hat{\alpha}_m$, strong duality holds and the minimisers are related through the conditions:

$$\hat{v}_m \in \partial r^* \left(-\frac{1}{c_m} \sum_{j=1}^{m-1} \alpha_{m,j} \right) \qquad \hat{\alpha}_m \in \partial \left(G_{m-1} \right)(A_{m-1}(\hat{v}_m)) \,. \tag{3.6}$$

Since strong convexity of r (Assumption 1) implies strict convexity of r which implies that r^* is differentiable (by proposition A.3) we get eq. (3.5) by replacing the subdifferential in the left term.

Let us prove that the hypothesis for the theorem on Fenchel-Rockafellar duality hold. Strong convexity of r (Assumption 1) and convexity of g_m (Assumption 2) ensure that the primal objective is strongly convex thus coercive and strictly convex therefore it admits a unique solution. Any convex function is continuous on the interior of its domain, but the g_m's are real-valued on $\mathcal{V}$ which implies $dom(g_m) = \mathcal{V}$, therefore they are continuous on the entire space $\mathcal{V}$, which concludes the proof. □

Before the introduction of the general class of Primal-Dual Online algorithms, we recall the definition of approximate subgradients.

Definition 3.1 ($\epsilon-$subdifferential, $\epsilon > 0$)**.** *Let f be a convex function on $\mathcal{V}$,*

$$\partial_\epsilon f(v) = \{\alpha \in \mathcal{V} : f\left(v'\right) \geq f(v) + \langle \alpha, v' - v \rangle - \epsilon, \text{ for any } v' \in \mathrm{Dom} f\} \,.$$

If $u \in \partial_\epsilon f(v)$, u is called an $\epsilon-$**subgradient of** f at v. From the definition we get that v is an $\epsilon-$minimiser of f if and only if $0 \in \partial_\epsilon f(v)$. If $\epsilon = 0$ we retrieve standard subgradients (Definition A.2).

We are now ready to introduce the general algorithm. As it might be too expensive to find $\hat{v}_m$ or equivalently $\hat{\alpha}_m$ at each step m — i.e. to apply Follow-The-Regularized-Leader

— we would like to take less **aggressive** updates. We consider any strategy that outputs a sequence of dual variables $\alpha_1, \alpha_2, \ldots$ ($\alpha_m \in \mathcal{V}^{m-1}$) that satisfies the following property,

$$\boxed{\begin{aligned} \epsilon_m \geq 0, \qquad \alpha'_m \in \partial_{\epsilon_m} g_m(v_m) \\ \hat{D}_{m+1}(\alpha_{m+1}) \leq \hat{D}_{m+1}(\underbrace{\alpha_{m,1}, \ldots, \alpha_{m,m-1}}_{=\alpha_m}, \alpha'_m) = \hat{D}_{m+1}(\alpha_m, \alpha'_m), \end{aligned}} \tag{3.7}$$

where v_m is related to α_m with the same optimality condition as in Theorem 3.1,

$$v_m = \nabla r^* \left(-\frac{1}{c_m} \sum_{j=1}^{m-1} \alpha_{m,j} \right). \tag{3.8}$$

We call the set of strategies that satisfy this rule the Primal-Dual Online family. The aggressiveness of a sequence of dual updates $\alpha_1, \alpha_2, \ldots$ is measured by how large is the inequality in eq. (3.7) at each step. If $\epsilon_m = 0$, we see that Online Mirror Descent is the least aggressive strategy as it saturates the inequality at each step. Indeed, Online Mirror Descent greedily update only the last coordinate of α_m with the update rule $\alpha_{m+1} = (\alpha_m, \alpha'_m)$, $\alpha'_m \in \partial g_m(v_m)$, therefore,

$$\hat{D}_{m+1}(\alpha_{m+1}) = \hat{D}_{m+1}(\alpha_m, \alpha'_m). \tag{3.9}$$

On the other hand of the aggressiveness spectrum, Follow-The-Regularized-Leader updates all the dual coordinates and maximise the size of the inequality. Indeed, for FTRL, we take $\alpha_{m+1} \in \partial(G_m)(A_m(v_{m+1}))$ where $v_{m+1} = \arg\min_{v \in \mathcal{V}} \hat{P}_{m+1}(v)$, therefore, by Theorem 3.1, $\alpha_{m+1} = \arg\min_{\alpha \in \mathcal{V}^m} \hat{D}_{m+1}(\alpha)$, hence, for $\alpha'_m \in \partial g_m(v_m)$,

$$D_{m+1}(\alpha_m, \alpha'_m) - \hat{D}_{m+1}(\alpha_{m+1}) = \max_{\alpha \in \mathcal{V}^m} D_{m+1}(\alpha_m, \alpha'_m) - \hat{D}_{m+1}(\alpha). \tag{3.10}$$

A plethora of intermediate strategies are worth considering, for example, doing a hard optimisation of only the last coordinate or a group of the last coordinates. When $\epsilon_m > 0$, we take into account algorithms with imperfect access to subgradients, the errors will accumulate in the dual bound. The general algorithm for this family is given in Algorithm 2 and the associated dual gap bounds are given in Theorem 3.2.

Theorem 3.2 (Dual Bounds, Thm. 1 Denevi et al. (2019b))**.** *Let $(v_m)_{m=1}^M$ be the primal iterates outputted by Algorithm 2 and let $\Delta_{Dual} = \hat{D}_{M+1}(\alpha_{M+1}) - D_{M+1}$ be the corresponding (non-negative) dual optimality gap at the last dual iterate of the algorithm. If Assumptions 1 and 2 are satisfied,*

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

$$\Delta_{Dual} \leq -\sum_{m=1}^{M} g_m(v_m) + P_{M+1} + \frac{1}{2s} \sum_{m=1}^{M} \frac{1}{c_m} \|\alpha'_m\|_*^2 + \sum_{m=1}^{M} \epsilon_m.$$

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

$$\Delta_{Dual} \leq -\sum_{m=1}^{M} \{g_m(v_m) + \lambda_m r(v_m)\} + P_{M+1} + \frac{1}{2s} \sum_{m=1}^{M} \frac{1}{c_m} \|\alpha'_m\|_*^2 + \sum_{m=1}^{M} \epsilon_m,$$

where for all $m \geq 1$, $\alpha'_m \in \partial g_m(v_m)$ *and* $\|.\|_*$ *is the dual norm of* $\|.\|$.

Algorithm 2: General Primal-Dual algorithm

Init: $v_0 = \nabla r^*(0)$;
Loop for $m \leftarrow 1, \dots, M$ do

Receive g_m, c_m, ϵ_m

suffer $g_m(v_m)$,

take any update α_{m+1} satisfying eq. (3.7)

$v_{m+1} = \nabla r^* \left(-\frac{1}{c_m} \sum_{j=1}^{m} \alpha_{m+1,j} \right)$

Return: $(\alpha_m)_{m=2}^{M+1}$, $(v_m)_{m=1}^{M+1}$

Remark. The dual gap bounds are valid for any algorithm that satisfies eq. (3.7) from the least aggressive (OMD) to the most aggressive (FTRL). However, keep in mind that OMD is simple to run as we only need access to subgradients while more aggressive schemes necessitate more involved optimisation steps. If we assume more structure on the underlying problem and that a more aggressive algorithm than OMD is computable it might be more effective in practice.

3.3 Meta strategy from Primal-Dual Online Learning

In this section we show how we can apply Theorem 3.2 to the within-task problem as well as the meta layer and present the meta regret bound.

3.3.1 Within-task algorithm

We recall the definition of the meta objective,

$$\mathcal{L}_t(\theta) = \min_{\mu \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} L_{t,i}(\mu) + \lambda D(q_\mu, \pi_\theta), \qquad \theta \in \Theta. \tag{3.11}$$

It motivates the use of a within-task online algorithm that mimics this batch objective. We apply a Primal-Dual Online learning strategy on the following Primal-Dual pair where

the primal is the non-normalized within-task problem associated with eq. (3.11). We fix $\theta \in \Theta$ and $t \geq 1$. We introduce the notation $\psi_\theta : \mu \mapsto D(q_\mu, \pi_\theta)$, for all $i \geq 1$,[2]

$$\begin{aligned} P^\theta_{t,i+1} &:= \inf_{\mu \in \mathcal{M}} \hat{P}_{t,i+1}(\mu, \theta) \qquad \hat{P}_{t,i+1}(\mu, \theta) := \sum_{j=1}^{i} L_{t,j}(\mu) + n\lambda\psi_\theta(\mu) \\ D^\theta_{t,i+1} &:= \inf_{\alpha \in \mathcal{M}^i} \hat{D}_{t,i+1}(\alpha, \theta) \qquad \hat{D}_{t,i+1}(\alpha, \theta) := \sum_{j=1}^{i} L^*_{t,j}(\alpha_j) + n\lambda\psi^*_\theta\left(-\frac{1}{n\lambda}\sum_{j=1}^{i}\alpha_j\right). \end{aligned} \tag{3.12}$$

The strategy we choose is Online Mirror Descent but it is also possible to use a more aggressive strategy as mentioned in the last section. For all $i \geq 2$, we take $\mu_{t,i} \in \mathcal{M}$, $\alpha'_{t,i} \in \mathcal{M}^{i-1}$ where we greedily update only the last coordinate of $\alpha'_{t,i}$ with the update rule $\alpha'_{t,i} = (\alpha'_{t,i-1}, \alpha_{t,i-1})$, $\alpha_{t,i-1} \in \partial L_{t,i-1}(\mu_{t,i-1})$ and relates $\mu_{t,i}$ to $\alpha_{t,i}$ through the optimality condition $\mu_{t,i} = \nabla\psi^*\left(-\frac{1}{n\lambda}\sum_{j=1}^{i-1}\alpha'_{t,i,j}\right) = \nabla\psi^*\left(-\frac{1}{n\lambda}\sum_{j=1}^{i-1}\alpha_{t,j}\right)$. The algorithm is given in Algorithm 3. By identification to (eq. (3.2) - eq. (3.3)) with $i \leftarrow m$, $n \leftarrow M$, $n\lambda \leftarrow c_m$, $\mathcal{M} \leftarrow \mathcal{V}$, $\psi_\theta \leftarrow r$, $L_{t,j} \leftarrow g_j$ and $\epsilon_j = 0$ (no approximation we have access to the subgradients of $L_{t,i}$), we apply the second point in Theorem 3.2 and get the following dual bound.

Proposition 3.1 (Inner regularized duality gap bound)**.** *Let $\theta \in \Theta$ be fixed, $(\mu_{t,i})_{i=1}^n$ are the primal iterates and $\alpha'_{t,n+1} = (\alpha_{t,i})_{i=1}^n$ is the last dual iterate returned by Algorithm 3 with meta parameter θ on dataset Z_t, for all $i \geq 1$, $\alpha_{t,i} \in \partial L_{t,i}(\mu_{t,i})$. $\Delta_{Dual} = \hat{D}_{t,n+1}\left(\alpha'_{t,n+1}, \theta\right) - D^\theta_{t,n+1}$ is the corresponding (non-negative) dual optimality gap at the last dual iterate. If $L_{t,i}$ satisfies Assumption 2 and ψ_θ Assumption 1 with respect to a norm $\|.\|_\theta$ and strongly convex parameter s_θ, then,*

$$\Delta_{Dual} \leq -\sum_{i=1}^{n}\left\{L_{t,i}(\mu_{t,i}) + \lambda\psi_\theta(\mu_{t,i})\right\} + n\mathcal{L}_t(\theta) + \frac{1}{2s_\theta\lambda}\sum_{i=1}^{n}\frac{1}{i}\|\alpha_{t,i}\|^2_{\theta,*} := \epsilon_\theta. \tag{3.13}$$

Remark 1. The strongly convex parameter for ψ_θ can depends on θ and similarly Assumptions 1 and 2 are with respect to a norm that can depends on θ. That is why we use the notations s_θ and $\|.\|_\theta$.

Remark 2. Assumption 1 for ψ_θ and Assumption 2 for $L_{t,i}$ are not trivial. For example, when ψ_θ is the Kullback-Leibler divergence it is known that it is jointly convex *with respect to the distributions* but it might not stay convex *with respect to the parameters* of those distributions (e.g with respect to the mean and the variance of two Gaussian distributions). For Assumption 2, when $L_{t,i}(\mu) = \mathbb{E}_{w\sim q_\mu}\left[\ell_{t,i}(w)\right]$, if the underlying losses $\ell_{t,i}$ are convex and $\{q_\mu, \mu \in \mathcal{M} \subseteq \mathbb{R}^p\}$ is a location-scale family (e.g. Gaussian, Student, Laplace etc), it can be shown that $L_{t,i}$ is convex. Those assertions will be discussed and proved in Chapter 4.

[2] notice that since $\mathcal{L}_t$ is a regularized empirical error, in comparison to Chapter 2, the regularization strength is $n\lambda$ instead of λ, it leads to slightly different bounds.

From the bound, we see that the last dual iterate $(\alpha_{t,i})_{i=1}^n$ is an $\epsilon_\theta-$minimiser of the dual problem $\hat{D}_{t,n+1}$ where ϵ_θ is the right hand side in eq. (3.13). In the next section, we show that this fact can be used to build approximate subgradients for the meta objective.

3.3.2 Meta subgradient

For any meta parameter $\theta \in \Theta$ and task t, Proposition 3.2 connects the dual gap in eq. (3.13) to the construction of an approximate subgradient $\nabla'_\theta \in \partial_{\epsilon_\theta/n}\mathcal{L}_t(\theta)$.

Proposition 3.2 (ϵ-subgradient for $\mathcal{L}_t$ - Prop 3 Denevi et al. (2019b))**.** *Let $\bar{\alpha}_\theta := (\alpha_{t,i})_{i=1}^n \in \mathcal{M}^n$ be the dual output of Algorithm 3 with meta parameter θ on dataset Z_t. Let $\nabla_\theta \in \partial\left\{-\hat{D}_{t,n+1}\left(\bar{\alpha}_\theta, \cdot\right)\right\}(\theta)$, where*

$$\hat{D}_{t,n+1}(\alpha, \theta) = \sum_{i=1}^n L_{t,i}^*\left(\alpha_i\right) + \lambda n \psi_\theta^*\left(-\frac{1}{\lambda n}\sum_{i=1}^n \alpha_i\right) \quad \alpha \in \mathcal{M}^n,$$

is the dual of the non-normalized inner problem associated to task t. Under the same assumptions that in Proposition 3.1, $\nabla'_\theta := \nabla_\theta/n \in \partial_{\epsilon_\theta/n}\mathcal{L}_t(\theta)$, where ϵ_θ is defined in Proposition 3.1.

Proof. Assumptions 1 and 2 ensure that we can apply Theorem 3.1 to the inner algorithm. By strong duality, since $\hat{D}_{t,n+1}$ is the dual problem associated to the non-normalized within-task problem, for all $\theta' \in \Theta$, it holds that

$$n\mathcal{L}_t(\theta') = -\min_{\alpha \in \mathcal{M}^n} \hat{D}_{t,n+1}(\alpha, \theta'),$$

i.e.

$$\mathcal{L}_t(\theta') = \max_{\alpha \in \mathcal{M}^n} \tilde{D}_{t,n+1}(\alpha, \theta') \quad \tilde{D}_{t,n+1}(\alpha, \theta') := -\frac{1}{n}\hat{D}_{t,n+1}(\alpha, \theta').$$

From Proposition 3.1 we know that $\bar{\alpha}_\theta$ is an ϵ_θ minimiser of $\hat{D}_{t,n+1}(., \theta)$, therefore it is an ϵ_θ/n maximiser of $\tilde{D}_{t,n+1}(., \theta)$. For any $\theta' \in \Theta$, we have,

$$\begin{aligned}
\mathcal{L}_t\left(\theta'\right) &= \max_{\alpha \in \mathcal{M}^n} \tilde{D}_{t,n+1}\left(\alpha, \theta'\right) \\
&\geq \tilde{D}_{t,n+1}\left(\bar{\alpha}_\theta, \theta'\right) \\
&\geq \tilde{D}_{t,n+1}\left(\bar{\alpha}_\theta, \theta\right) + \left\langle \frac{\nabla_\theta}{n}, \theta' - \theta \right\rangle \\
&\geq \max_{\alpha \in \mathcal{M}^n} \tilde{D}_{t,n+1}\left(\alpha, \theta\right) - \frac{\epsilon_\theta}{n} + \left\langle \frac{\nabla_\theta}{n},\ \theta' - \theta \right\rangle \\
&= \mathcal{L}_t(\theta) + \left\langle \frac{\nabla_\theta}{n}, \theta' - \theta \right\rangle - \frac{\epsilon_\theta}{n}.
\end{aligned} \tag{3.14}$$

In the second inequality we use that by definition $\nabla_\theta/n \in \partial\left\{\tilde{D}_{t,n+1}\left(\bar{\alpha}_\theta, \cdot\right)\right\}(\theta)$, and in the third inequality we use that $\bar{\alpha}_\theta$ is an ϵ_θ/n maximiser of $\tilde{D}_{t,n+1}(., \theta)$. It shows that $\nabla_\theta/n \in \partial_{\epsilon_\theta/n}\mathcal{L}_t(\theta)$. □

It gives a simple rule to build approximate subgradients. The only requirement is to know the gradient *with respect to the meta parameter* of the conjugate function ψ_θ^* and to plug the output of the inner algorithm.

3.3.3 Meta Algorithm and dual gap bound

It the last sections, we proved a dual bound for the inner algorithm and a way to build approximate subgradients of the meta loss using only Assumptions 1 and 2 *on the inner losses*. Introducing a meta regularization function $\phi : \Theta \to \mathbb{R}_+$ and regularization strength η we are thus able to run approximate Online Mirror Descent on the meta losses. Notations for the within-task OMD and meta OMD are summarized in Table 3.1. To build a dual gap bound we need Assumption 1 to hold for ϕ with respect to a norm $|||.|||$ and strongly convex parameter $\tilde{s}$, and Assumption 2 for $(\mathcal{L}_t)_{t\geq 1}$. As it is not obvious that $\mathcal{L}_t$ is convex, we introduce a sufficient condition.

Assumption 3 $\psi_\theta(\mu)$ is jointly convex in (μ, θ) on $\mathcal{M} \times \Theta$.

Assumption 3 implies that $\mathcal{L}_t$ is convex on Θ by Proposition A.2. By identification to (eq. (3.2) - eq. (3.3)) with $t \leftarrow m$, $T \leftarrow M$, $\eta \leftarrow c_m$, $\Theta \leftarrow \mathcal{V}$, $\phi \leftarrow r$, $\mathcal{L}_t \leftarrow g_j$ running Online Mirror Descent gives Algorithm 4. As the reader might feel overwhelmed by the heavy notations used up to this place we provide a summary of the notations and assumptions in Table 3.1 and Table 3.2. We apply Theorem 3.2 point 1 to get the following duality gap bound.

Proposition 3.3 (Meta regularized duality gap bound)**.** *Run Algorithm 4 (which itself runs Algorithm 3 on each task) that outputs $(\mu_{t,i})_{i=1}^n$ for all $t \geq 1$ and $(\theta_t)_{t=1}^T$. If the $L_{t,i}$ satisfy Assumption 2, ψ_θ Assumption 1 with strongly convex coefficient s_θ and norm $||.||_\theta$ and Assumption 3 (it ensures that $\mathcal{L}_t$ satisfies Assumption 2), and ϕ satisfies Assumption 2 with strongly coefficient $\tilde{s}$ and norm $|||.|||$, then, using the fact that the dual gap is always positive we get,*

$$0 \leq -\sum_{t=1}^T \mathcal{L}_t(\theta_t) + \min_{\theta\in\Theta} \sum_{t=1}^T \{\mathcal{L}_t(\theta) + \eta\phi(\theta)\} + \frac{1}{2\eta\tilde{s}} \sum_{t=1}^T |||\nabla_{\theta_t}|||_*^2 + \sum_{t=1}^T \epsilon_{\theta_t}. \tag{3.15}$$

Plugging the value of ϵ_{θ_t} and rearranging the term, we get,

$$\begin{aligned}\sum_{t=1}^T \sum_{i=1}^n L_{t,i}(\mu_{t,i}) + \lambda\psi_{\theta_t}(\mu_{t,i}) \leq \inf_{\substack{\theta\in\Theta \\ (\bar{\mu}_t)_{t=1}^T \in \mathcal{M}^T}} nT\Bigg\{ &\frac{\eta}{T}\phi(\theta) + \frac{\lambda}{T}\sum_{t=1}^T \psi_\theta(\bar{\mu}_t) + \\ &\frac{1}{nT}\sum_{t=1}^T\sum_{i=1}^n L_{t,i}(\bar{\mu}_t) + \frac{1}{2\lambda nT}\sum_{t=1}^T \frac{1}{s_{\theta_t}} \sum_{i=1}^n \frac{||\alpha_{t,i}||_{\theta_t,*}^2}{i} + \frac{1}{2\tilde{s}\eta T}\sum_{t=1}^T |||\nabla_{\theta_t}|||_*^2\Bigg\}.\end{aligned} \tag{3.16}$$

Remark. To get a more explicit bound with noticeable convergence rates, we need:

(i) the exact form of ϕ and ψ_θ

(ii) to precise the variational family and the prior family

(iii) a control over the norm of the inner subgradients and the meta subgradients. The inner subgradients can usually be bounded by a constant under a Lipschitz hypothesis. The bound on $|||\nabla_{\theta_t}|||_*^2$ on the other hand might hide a dependency with respect to λ and n which is non-trivial to analyse. In the next chapter, we give an example in the Gaussian Mean Field setting and derive the corresponding meta regret bound.

Table 3.1: Summary of the notations for the within-task and meta OMD

	Within-Task Algorithm	**Meta Algorithm**
Algorithm	GVI-OMD	Approximate-OMD
Parameter	$\mu \in \mathcal{M}$	$\theta \in \Theta$
Regularizer	ψ_θ	ϕ
Learning rate	$i\lambda$	η
Underlying family of distributions	q_μ	π_θ

Table 3.2: Summary of the assumptions for the within-task and meta OMD

	Assumptions
ψ_θ	s_θ-strongly convex, proper and closed on $\mathcal{M}$ w.r.t. $\|\cdot\|_\theta$, $\inf_{\mu\in\mathcal{M}} \psi_\theta(\mu) = 0$ jointly convex as a function of (μ, θ) on $\mathcal{M} \times \Theta$
$L_{t,i}$	closed, convex and real-valued on $\mathcal{M}$
ϕ	$\tilde{s}$-strongly convex, proper and closed on Θ w.r.t. $\|\|\|\cdot\|\|\|$, $\inf_{\theta\in\Theta} \phi(\theta) = 0$

Algorithm 3: Within-task (GVI)

Input: $\theta \in \Theta$, $\lambda > 0$, task Z_t

Init: $\mu_{t,1} = \nabla\psi_\theta^*(0)$;

Loop for $i \leftarrow 1, \dots, n$ do

Receive $\ell_{t,i}$, $\hat{w}_{t,i} = \mathbb{E}_{w\sim q_{\mu_{t,i}}}[\ell_{t,i}(w)]$

suffer $\ell_{t,i}(\hat{w}_{t,i})$,

$\alpha_{t,i} \in \partial L_{t,i}(\mu_{t,i})$

$\mu_{t,i+1} = \nabla\psi_\theta^*\left(-\frac{1}{i\lambda}\sum_{j=1}^{i}\alpha_{t,j}\right)$

Return: $(\alpha_i)_{i=2}^{n+1}$, $(\mu_i)_{i=1}^{n+1}$

Algorithm 4: Meta algorithm

Input: $\eta > 0$

Init: $\theta_1 = \nabla\phi^*(0)$;

Loop for $t \leftarrow 1, \dots, T$ do

Receive incrementally the dataset Z_t

run Algorithm 3 with θ_t over Z_t

Compute ∇_*^t as in Proposition 3.2

$\theta_{t+1} = \nabla\phi^*\left(-\frac{1}{\eta}\sum_{j=1}^{i}\nabla_*^j\right)$

Return: $(\theta_t)_{t=1}^{T+1}$

Conclusion of the chapter. We have introduced the sequence of meta losses and the Primal-Dual Online Learning framework that allows to theoretically analyse a meta Online Mirror Descent on the meta losses. We provided the within-task and meta abstract Online Mirror Descent derived from this framework. The within-task OMD is standard and the subgradients for the meta OMD are approximated with the dual outputs of the within-task OMD. Despite this approximation, a meta regret bound for the entire procedure is given in Proposition 3.3. In the next chapter, we specialise those results in a practical setting with a Gaussian Mean Field variational family.

Chapter 4

A Practical Algorithm in the Gaussian Mean Field setting

In this chapter, we derive a practical algorithm and its regret bound to solve the online Meta Learning problem. We use a Gaussian Mean Field family for both the variational family and the prior family. We only learn the mean at the inner level, we refer to this algorithm as 2OGD for "Double Online Gradient Descent". 2OGD can be seen as an extension of the strategy presented in Denevi et al. (2019b) where, by learning the meta variance, we adaptively learn the gradient step size coordinate by coordinate for the inner OGD.

4.1 Setting

As in the preceding chapters we iteratively receive a sequence of sequence of losses indexed by $\{t, i\}$: $\ell_{t,i}$, that we assume convex for all $t \in [T]$ and $i \in [n]$. We specify an inner and a meta Online Mirror Descent that satisfies the hypothesis summed up in Table 3.1 and Table 3.2 in the Gaussian Mean field setting. Let $\sigma^2 \in \mathbb{R}^d_{++}$ be a fixed variance parameter. The Gaussian Mean Field variational family is the most common for large scale applications of Variational Inference such as deep neural networks Graves (2011), Blundell et al. (2015). The Gaussian family usually leads to simple updates and by learning only a diagonal variance-covariance matrix, it stays memory efficient. The choice of variational family is

$$\{q_m, m \in \mathbb{R}^d\} = \{\mathcal{N}(m, \mathrm{Diag}(\sigma^2)), m \in \mathbb{R}^d\}.$$

In this setting, the operator we take for the loss is $L_{t,i}(m) = \ell_{t,i}(\mathbb{E}_{w\sim q_m}[w]) = \ell_{t,i}(m)$. $L_{t,i}$ does not depend on σ^2, therefore we can just ignore this parameter and set it to 1_d. A more involved algorithm can use $L_{t,i}(m, \sigma^2) = \mathbb{E}_{w\sim\mathcal{N}(m,\mathrm{Diag}(\sigma^2))}[\ell_{t,i}(w)]$ to learn both m and σ^2. We start with the simple case where we only learn m. For the prior family we take a Gaussian family parametrised with a diagonal variance-covariance matrix,

$$\{\pi_{\mu,v}, (\mu, v) \in \mathbb{R}^d \times \mathbb{R}^d_{++}\} = \{\mathcal{N}(\mu, \mathrm{Diag}(v)), (\mu, v) \in \mathbb{R}^d \times \mathbb{R}^d_{++}\}.$$

For the inner and meta regularisation functions we take two "Euclidean-like" functions,

- $\psi_{\mu,v}(m) = \sum_{i=1}^{d} \left\{ \frac{(m_i-\mu_i)^2}{2v_i} + v_i \right\}$
- $\phi(\mu, v) = \frac{1}{2s^2}||\mu||_2^2 + \frac{1}{2s^2}||v - s^2 1_d||_2^2$

s^2 can be seen as the variance of a meta prior $\mathcal{N}(0, s^2 I_d)$. In comparison to Table 3.1, the choices are summarized in Table 4.1,

Table 4.1: Summary of the choices for the Gaussian Mean Field 2OGD

	Within-Task Algorithm	**Meta Algorithm**
Algorithm	GVI-OMD	Approximate-OMD
Parameter	$m \in \mathbb{R}^d$	$(\mu, v) \in \mathbb{R}^d \times \mathbb{R}^d_{++}$
Regularizer	$\psi_{\mu,v}(m) = \sum_{i=1}^{d} \frac{(m_i-\mu_i)^2}{2v_i} + v_i$	$\phi(\mu, v) = \frac{1}{2s^2}\|\|\mu\|\|_2^2 + \frac{1}{2s^2}\|\|v - s^2 1_d\|\|_2^2$
Learning rate	$i\lambda$	η
Underlying distributions	$q_m \rightsquigarrow \mathcal{N}_d(m, I_d)$	$\pi_{\mu,v} \rightsquigarrow \mathcal{N}(\mu, \mathrm{Diag}(v))$

We need to check that this set up satisfies the functional specifications given in Table 3.2. The convexity of the $L_{t,i}$ follows from the convexity of the $\ell_{t,i}$ since $L_{t,i}(m) = \ell_{t,i}(m)$, the rest is specified in Proposition 4.1.

Proposition 4.1. *$\psi_{\mu,v}$ is $\|v\|_\infty^{-1}-$strongly convex in m on $\mathbb{R}^d$ with respect to the Euclidean norm and ϕ is $s^{-2}-$strongly convex in (μ, v) on $\mathbb{R}^d \times \mathbb{R}^d_{++}$ with respect to the Euclidean norm. Furthermore, $\psi_{\mu,v}(m)$ is jointly convex in (m, μ, v) and $\inf_{(\mu,v)\in\mathbb{R}^d\times\mathbb{R}^d_{++}} \phi(\mu, v) = 0$, $\inf_{m\in\mathbb{R}^d} \psi_{\mu,v}(m) = 0$.*

Proof. $\nabla^2_m \psi_{\mu,v}(m) = \mathrm{Diag}(v^{-1})$ and $\nabla^2_{\mu,v}\phi(\mu, v) = s^{-2} I_{2d}$. Their smallest eigenvalues are respectively $(\max_i v_i)^{-1} > 0$ and $s^{-2} > 0$ and give the coefficients of strong convexity by Proposition A.1. For the joint convexity of $\psi_{\mu,v}(m)$ in (m, μ, v), notice that $(x, t) \in \mathbb{R} \times \mathbb{R}_{++} \mapsto \frac{x^2}{t}$ is convex in (x, t) as it is the perspective function of the convex function $x \mapsto x^2$ (see section 3.2.6 in Boyd et al. (2004)). Therefore $\psi_{\mu,v}(m)$ is convex in (m, μ, v) as a sum of jointly convex functions and a composition with a linear transformation. □

Comments on the regularisation functions. The choices for the regularisation functions may seem odd. Indeed if we use only Gaussian distributions, why not using two Kullback-Leibler divergences ? The reason is that it does not satisfy the requirements given by Table 3.2. For the inner regularisation, the choice $\psi_{\mu,v}(m) = KL(\mathcal{N}(m, I_d), \mathcal{N}(\mu, \mathrm{Diag}(v)))$ is not jointly convex in (m, μ, v) (Proposition A.6 and Proposition A.7). With our choice we cut off the part of the KL divergence that is not convex in v. Let us notice that we parametrise the prior family with the variance v and not with a standard deviation $\hat{v} = \sqrt{v}$ because otherwise we would have used $\frac{1}{2}\sum_{i=1}^{d} \frac{(m_i-\mu_i)^2}{\hat{v}^2}$ which is not jointly convex either (the function $(x, t) \in \mathbb{R} \times \mathbb{R}_{++} \mapsto \frac{x^2}{t^2}$ is not jointly convex). For this reason, we also did not choose a Kullback Leibler divergence at the meta level, $\phi(\mu, v) = KL(\mathcal{N}(\mu, \mathrm{Diag}(v)), \mathcal{N}(0, s^2 I_d))$, because it is not strongly convex when we parametrise with the variance (it is only strictly convex see Proposition A.7). To overcome this, we can still use a KL penalisation if we restrict

the set of v to a compact $K \subset \mathbb{R}^d_{++}$ or if we add a strongly convex term, for example, $\phi(\mu, v) = KL(\mathcal{N}(\mu, \text{Diag}(v)), \mathcal{N}(0, s^2 I_d)) + ||v||_2^2$. For simplicity, we stick to ϕ as defined in Table 4.1.

4.2 Within-task Algorithm

In this part, we derive the within-task algorithm by applying the generic Algorithm 3 to the Gaussian Mean Field setting. The algorithm we obtain is reported in Algorithm 5. We fix the meta variable $(\mu, v) \in \mathbb{R}^d \times \mathbb{R}^d_{++}$ and dataset Z_t but omit the index t for readability. To derive Algorithm 5 from Algorithm 3, we need to compute $\nabla\psi^*_{\mu,v}$ as the update is given by,

$$\begin{aligned} m_1 &= \nabla\psi^*_{\mu,v}(0) \\ \forall i \geq 2 \quad m_{i+1} &= \nabla\psi^*_{\mu,v}\left(-\frac{1}{i\lambda}\sum_{j=1}^{i}\alpha_j\right), \qquad \forall j \quad \alpha_j \in \nabla\ell_j(m_j). \end{aligned} \tag{4.1}$$

Proposition 4.2. $\psi_{\mu,v}(m) = \frac{1}{2}\sum_{i=1}^d \frac{(m_i-\mu_i)^2}{v_i} + \sum_{i=1}^d v_i$. *For all* $\alpha \in \mathbb{R}^d$,

$$\begin{aligned} \psi^*_{\mu,v}(\alpha) &= \langle\alpha, \mu\rangle + \frac{1}{2}\langle\alpha^2, v\rangle - \sum_{i=1}^d v_i \\ \nabla\psi^*_{\mu,v}(\alpha) &= (\mu_1 + v_1\alpha_1, \ldots, \mu_d + v_d\alpha_d), \end{aligned} \tag{4.2}$$

where α^2 *is the vector* α *with each coordinate raised to the power 2.*

Proof. Let us study the conjugate function of $g_{y,z}(x) = \frac{(x-y)^2}{2z}$, where $x, y, z \in \mathbb{R}\times\mathbb{R}\times\mathbb{R}_{++}$. $g_{y,z}$ is strictly convex in x, therefore, by Proposition A.3, $g^*_{y,z}$ is differentiable and for all $\alpha \in \mathbb{R}$,

$$\begin{aligned} \nabla g^*_{y,z}(\alpha) &= \arg\max_{x\in\mathbb{R}} \alpha x - g_{y,z}(x) \\ &= \arg\max_{x\in\mathbb{R}} z\alpha x - \frac{x^2}{2} + xy \\ &= \arg\max_{x\in\mathbb{R}} -\frac{1}{2}(y + z\alpha - x)^2 \\ &= y + z\alpha. \end{aligned}$$

By plugging back this solution in the definition of $g^*_{y,z}$, we get,

$$\begin{aligned} g^*_{y,z}(\alpha) &= \sup_{x\in\mathbb{R}} \alpha x - g_{y,z}(x) \\ &= \alpha(y + z\alpha) - g_{y,z}(y + z\alpha) \\ &= \alpha(y + z\alpha) - \frac{z\alpha^2}{2} \\ &= \alpha y + \frac{z\alpha^2}{2}. \end{aligned}$$

For any function g on $\mathbb{R}^d$ that admits the decomposition $g(x) = \sum_{i=1}^d g_i(x_i)$ its Fenchel transform admits the decomposition $g^*(\alpha) = \sum_{i=1}^d g_i^*(\alpha_i)$. Thus, if the g_i^* are differentiable $\nabla g^*(\alpha) = (\nabla g_1^*(\alpha_1), \dots, \nabla g_d^*(\alpha_d))$. Furthermore, if $h(x) = f(x) + C$ for all $x \in \mathbb{R}^d$ where C does not depend on x, then $h^*(\alpha) = f^*(\alpha) - C$ for all $\alpha \in \mathbb{R}^d$. Here, $\psi_{\mu,v}(m) = \sum_{i=1}^d g_{\mu_i,v_i}(m_i) + C$ ($C := \sum_{i=1}^d v_i$), therefore,

$$\begin{aligned} \psi_{\mu,v}^*(\alpha) &= \sum_{i=1}^d g_{\mu_i,v_i}^*(\alpha_i) - \sum_{i=1}^d v_i = \sum_{i=1}^d \alpha_i\mu_i + \frac{v_i}{2}\alpha_i^2 - \sum_{i=1}^d v_i \\ \nabla\psi_{\mu,v}^*(\alpha) &= (g_{\mu_1,v_1}^*(m_1), \dots, \nabla g_{\mu_d,v_d}^*(m_d)) = (\mu_1 + v_1\alpha_1, \dots, \mu_d + v_d\alpha_d). \end{aligned} \tag{4.3}$$

□

The update for m_i is obtained by plugging the expression of $\nabla\psi_{\mu,v}^*(\alpha)$ in *eq.* (4.1). Proposition 4.3 shows that it boils down to a simple online gradient descent on the *regularized* loss functions. The resulting within-task algorithm is given in Algorithm 5.

Proposition 4.3 (Recursive update)**.** *eq.* (4.1) *can be written recursively as,*

$$m_{i+1} = m_i - \underbrace{\frac{v}{\lambda i}}_{:=\eta_i}\underbrace{\left(\alpha_i + \frac{\lambda}{v}(m_i - m_1)\right)}_{:=g_i} = m_i - \eta_i g_i. \tag{4.4}$$

Additions, multiplications and divisions between vectors are to be understood component-wise.

Proof. From eq. (4.1) and Proposition 4.2, $m_1 = \nabla\psi_{\mu,v}^*(0) = \mu$, and for all $i \geq 2$,

$$\begin{aligned} m_{i+1} = \underbrace{\mu}_{=m_1} - \frac{v}{\lambda i}\sum_{j=1}^i \alpha_j &= m_1 - \frac{v}{\lambda i}\left(\sum_{j=1}^{i-1}\alpha_j + \alpha_i\right) \\ &= m_1 - \frac{v}{\lambda i}\left(\frac{\lambda(i-1)}{v}(m_1 - m_i) + \alpha_i\right) && \text{by def of } m_i \\ &= m_1 - \frac{v}{\lambda i}\left(\left(\frac{\lambda i - \lambda}{v}\right)(m_1 - m_i) + \alpha_i\right) \\ &= m_i - \frac{v}{\lambda i}\left(\alpha_i + \frac{\lambda}{v}(m_i - \mu)\right). \end{aligned}$$

□

Comments on Algorithm 5. Since $g_i \in \partial\left(\ell_i(m_i) + \lambda\psi_{\mu,v}(m_i)\right)$, the update can be seen as an Online Gradient Descent on the *regularized losses*: $\ell_i(.) + \lambda\psi_{\mu,v}(.)$, therefore we call it **Regularized-OGD**. The starting point of the descent is the meta parameter μ and the step-size η_i depends on the meta parameter v. Therefore, by learning (μ, v) at the meta level, we learn both the starting point and the **step-size for each coordinate** for the within-task algorithm. This is the main improvement over the methods suggested in

Denevi et al. (2019a) and Denevi et al. (2019b) where they only learn the starting point μ. Finally, even if it returns the full primal and dual updates sequences $(\alpha_i)_{i=1}^n \in \mathbb{R}^{d\times n}$, $(m_i)_{i=1}^n \in \mathbb{R}^{d\times n}$ which can be memory inefficient if n and d are large, we will see that to run the full algorithm we only have to store a summary statistics of them of dimension d, so that the full algorithm stays memory efficient.

4.3 Meta algorithm

The sequence of losses we feed to the meta algorithm is $(\mathcal{L}_t)_{t\geq 1}$, where,

$$\begin{aligned}\mathcal{L}_t(\mu, v) &= \min_{m\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \ell_{t,i}(m) + \lambda\psi_{\mu,v}(m) \\ &= \min_{m\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \ell_{t,i}(m) + \lambda\sum_{i=1}^d \left\{\frac{(m_i-\mu_i)^2}{2v_i} + v_i\right\}.\end{aligned} \tag{4.5}$$

To run an Online Mirror Descent on $(\mathcal{L}_t)_{t\geq 1}$ we need (approximate) subgradients. The next proposition is a consequence of Proposition 3.2 that shows how to build approximate subgradients for $\mathcal{L}_t$.

Proposition 4.4 (Approximate subgradient for 2OGD)**.** *Let $(\alpha_{t,1}, \dots, \alpha_{t,n}) \in (\mathbb{R}^d)^n$ be the sequence of dual updates obtained after running Algorithm 5 on Z_t with meta parameters (μ, v), and $(m_{t,1}, \dots, m_{t,n}) \in (\mathbb{R}^d)^n$ is the primal sequence. Then ∇_*^t is an $(\epsilon_{\mu,v}/n)$-subgradient of $\mathcal{L}_t(\mu, v)$ where,*

$$\nabla_*^t = (\nabla_\mu^t, \nabla_v^t), \quad \nabla_\mu^t = \frac{1}{n}\sum_{i=1}^n \alpha_{t,i}, \quad \nabla_v^t = -\frac{1}{2\lambda}\left(\nabla_\mu^t\right)^2 + \lambda 1_d, \tag{4.6}$$

where the square is applied component-wise and

$$\epsilon_{\mu,v} = -\sum_{i=1}^n \{\ell_{t,i}(m_{t,i}) + \lambda\psi_{\mu,v}(m_{t,i})\} + n\mathcal{L}_t(\mu, v) + \frac{\|v\|_\infty}{2\lambda}\sum_{i=1}^n \frac{1}{i}\|\alpha_{t,i}\|_2^2. \tag{4.7}$$

Under the assumption that $\ell_{t,i}$ are $L-$Lipschitz w.r.t Euclidean norm, the norm of the meta subgradient is bounded by a constant (that depends on λ),

$$\|\nabla_*^t\|_2^2 = \|\nabla_\mu^t\|_2^2 + \|\nabla_v^t\|_2^2 \leq L^2 + \left(\frac{L}{2\lambda} + \lambda\sqrt{d}\right)^2. \tag{4.8}$$

Proof. Let $(\mu, v) \in \mathbb{R}^d \times \mathbb{R}^d_{++}$, we apply Proposition 3.2 to build a sub-gradient of $\mathcal{L}_t$ at (μ, v). For $\alpha \in \mathbb{R}^{d\times n}$ we denote by α_i its $i-$th column $i \in \{1, \dots, n\}$. If 1_n is the vector of $\mathbb{R}^n$ full of ones, $\sum_{i=1}^n \alpha_i = \alpha 1_n$. For a vector x of $\mathbb{R}^d$, the notation x^2 refers to the

vector y such that $y_i = x_i^2$, $i \in [d]$ (i.e. the square is applied component by component). By Proposition 4.2, for all $\alpha \in \mathbb{R}^{d\times n}$,

$$\begin{aligned}\hat{D}_{t,n+1}(\alpha,(\mu,v)) &= \sum_{i=1}^{n} \ell_{t,i}^*(\alpha_i) + \lambda n \psi_{\mu,v}^*\left(-\frac{1}{\lambda n}\sum_{i=1}^{n}\alpha_i\right) \\ &= \sum_{i=1}^{n} \ell_{t,i}^*(\alpha_i) + \frac{1}{2\lambda n}\left\langle (\alpha 1_n)^2, v\right\rangle - \langle \alpha 1_n, \mu\rangle - \lambda n \sum_{i=1}^{n} v_i.\end{aligned} \tag{4.9}$$

$D_{n+1}(\alpha,(\mu,v))$ is differentiable in (μ,v),

$$\begin{aligned}\nabla_\mu &= \nabla_\mu \left\{-\hat{D}_{t,n+1}(\alpha,(\mu,v))\right\}/n = \frac{\alpha 1_n}{n} \\ \nabla_v &= \nabla_v \left\{-\hat{D}_{t,n+1}(\alpha,(\mu,v))\right\}/n = -\frac{(\alpha 1_n)^2}{2\lambda n^2} = -\frac{1}{2\lambda}(\nabla_\mu)^2 + \lambda 1_d.\end{aligned} \tag{4.10}$$

The square is applied component-wise on $\alpha 1_n$. We get the approximate meta subgradients by Proposition 3.2. The expression for $\epsilon_{\mu,v}$ follows from *Proposition* 3.1, in the Gaussian Mean Field setting $\psi_{\mu,v}$ is $\left(\max_{j=1,\dots,d} v_j\right)^{-1}$−strongly convex with respect to the Euclidean norm (Proposition 4.1). If for all $t \geq 1$ and $i \geq 1$, $\ell_{t,i}$ is L−Lipschitz with respect to the Euclidean norm, it implies $\|\alpha_{t,i}\|_2 \leq L$. To bound the subgradient of μ, we use the triangular inequality,

$$\|\nabla_\mu\|_2 = \frac{1}{n}\left\|\sum_{i=1}^{n}\alpha_{t,i}\right\|_2 \leq \frac{1}{n}\sum_{i=1}^{n}\|\alpha_{t,i}\|_2 = L. \tag{4.11}$$

To bound the subgradient of v, notice that for any two vectors $x, y \in \mathbb{R}^d$, such that $y_i = x_i^2$ for all $i \in [d]$, $\|y\|_1 = \|x\|_2^2$, therefore using the triangular inequality and $\|.\|_2 \leq \|.\|_1$,

$$\begin{aligned}\|\nabla_v\|_2 \leq \frac{1}{2\lambda}\|(\nabla_\mu)^2\|_2 + \lambda\sqrt{d} &\leq \frac{1}{2\lambda}\|(\nabla_\mu)^2\|_1 + \lambda\sqrt{d} \\ &= \frac{1}{2\lambda}\|\nabla_\mu\|_2^2 + \lambda\sqrt{d} \\ &\leq \frac{L^2}{2\lambda} + \lambda\sqrt{d}.\end{aligned} \tag{4.12}$$

□

Comments on the meta subgradient and its bound. To build the meta subgradients we only need to store the sum of the dual iterations of the within-task algorithm $\sum_{i=1}^{n}\alpha_{t,i}$. Therefore we only have to store d parameters which is not too memory hungry. Let us note that the derivation of the meta subgradient does not depend on the choice of the meta regularisation function ϕ, it only depends on the within-task dual iterates and the inner regularisation function $\psi_{\mu,v}$. We also stress that the bound of the meta gradient depends on λ.

Meta algorithm. The second ingredient for the meta Online Mirror Descent is the gradient of the Fenchel transform of the regularisation function ϕ. We apply Proposition A.4,

$$\begin{aligned}
&\phi^*(\alpha,\beta) = \frac{s^2}{2}||\alpha||_2^2 + \frac{s^2}{2}||\beta||_2^2 + s^2\sum_{j=1}^{d}\beta_j \\
&\nabla_\alpha\phi^*(\alpha,\beta) = s^2\alpha \qquad \nabla_\beta\phi^*(\alpha,\beta) = 1_d s^2 + s^2\beta \\
&\nabla_\alpha\phi^*(0) = 0 \qquad \nabla_\beta\phi^*(0) = s^2 1_d
\end{aligned} \tag{4.13}$$

The update is given by,

$$\begin{aligned}
(\mu_1, v_1) &= \nabla\phi^*(0,0) = (0, s^2 1_d) \\
(\mu_{t+1}, v_{t+1}) &= \nabla\phi^*\left(-\frac{1}{\eta}\sum_{j=1}^{t}(\nabla_\mu^j, \nabla_v^j)\right) \\
&= \left(-\frac{s^2}{\eta}\sum_{j=1}^{t}\nabla_\mu^j, \quad s^2 1_d - \frac{s^2}{\eta}\sum_{j=1}^{t}\nabla_v^j\right) \\
&= \left(\mu_t - \frac{s^2}{\eta}\nabla_\mu^t, v_t - \frac{s^2}{\eta}\nabla_v^t\right) \qquad \forall t \geq 2
\end{aligned} \tag{4.14}$$

We use a double Euclidean regularisation for the parameters μ and ν, therefore the algorithm is an Online Gradient Descent on both μ and v. The within-task algorithm is also an Online Gradient Descent (on the regularized losses but it is still OGD), therefore we name the full algorithm 2OGD for *Double Online Gradient Descent*. For each dataset Z_t, μ_t sets the starting point of the within-task descent and v_t sets the stepsize of the within task descent coordinate by coordinate. The meta algorithm is given in Algorithm 6.

Algorithm 5: Within-task for GMF

Input: $(\mu, v) \in \mathbb{R}^d \times \mathbb{R}^d_{++}$, $\lambda > 0$, task Z_t

Init: $m_1 = \mu$

Loop for $i \leftarrow 1, \ldots, n$ do

Receive ℓ_i, suffer $\ell_i(m_i)$

receive a subgradient $\alpha_i \in \partial\ell_i(m_i) \subset \mathbb{R}^d$

$m_{i+1} = m_i - \frac{v}{\lambda i}\left(\alpha_i + \frac{\lambda}{v}(m_i - \mu)\right)$

Return: $\bar{\alpha} = (\alpha_i)_{i=1}^n \in \mathbb{R}^{d\times n}$, $(m_i)_{i=1}^n \in \mathbb{R}^{d\times n}$

Algorithm 6: Meta algorithm for GMF

Input: $s > 0$, $\eta > 0$

Init: $\mu^1 = 0$, $v^1 = s^2 1_d$;

Loop for $t \leftarrow 1, \ldots, T$ do

Receive incrementally the dataset Z_t

run Algorithm 5 with (μ^t, v^t) over Z_t

Compute $\nabla_*^t = (\nabla_{\mu^t}, \nabla_{v^t})$ as in Proposition 4.4

$\mu^{t+1} = \mu^t - \frac{s^2}{\eta}\nabla_{\mu^t}$

$v^{t+1} = v^t - \frac{s^2}{\eta}\nabla_{v^t}$

Return: $(\mu^t, v^t)_{t=1}^T$

4.4 Regret Bound

Theorem 4.1 (Regret bound for 2OGD)**.** *Under the assumptions stated in Table 3.2 and the assumption that the $\ell_{t,i}$ are $L-$Lipschitz w.r.t Euclidean norm and if $\sum_{t=1}^{T} \|v_t\|_\infty = O(T)$, there exists C_1, C_2 and C_3 constants such that,*

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n} \ell_{t,i}(m_{t,i}) \leq \inf_{\substack{\bar{v}\in\mathbb{R}_{++} \\ (\bar{m}_t)_{t=1}^T \in (\mathbb{R})^T}} \sum_{t=1}^{T}\sum_{i=1}^{n} \ell_{t,i}(\bar{m}_t) + C_1 n T^{3/4}\left\{\|\bar{\bar{m}}\|_2^2 + \bar{v}^2\right\} \\ + C_2 T\sqrt{n\log(n)}\left(\frac{\sigma_{\bar{m}}^2}{\bar{v}} + \bar{v}\right) + C_3 T\sqrt{n\log(n)}.\end{aligned} \tag{4.15}$$

where $\bar{\bar{m}} := \sum_{t=1}^{T} \frac{\bar{m}_t}{T}$ and $\sigma_{\bar{m}}^2 := \sum_{t=1}^{T} \frac{\|\bar{m}_t - \bar{\bar{m}}\|_2^2}{T}$. As a result, if $(\bar{m}_t)_{t=1}^T \in (\mathbb{R})^T$ minimises the term $\sum_{t=1}^{T}\sum_{i=1}^{n} \ell_{t,i}(\bar{m}_t)$, we have the regret bound,

$$\mathcal{R}_{meta}\left((Z)_{t=1}^T\right) \leq O(nT^{3/4}) + O(T\sqrt{n\log(n)}) + \sigma_{\bar{m}}^2 O(T\sqrt{n\log(n)}). \tag{4.16}$$

Proof. To get the regret bound of the full algorithm, we apply Proposition 3.3. ϕ is $s^{-2}-$strongly convex with respect to the Euclidean norm and $\psi_{\mu,v}$ is $\|v\|_\infty^{-1}$ strongly convex with respect to the Euclidean norm. Thus,

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n} \ell_{t,i}(m_{t,i}) + \lambda\psi_{\mu_t,v_t}(m_{t,i}) \leq \inf_{\substack{(\mu,v)\in\mathbb{R}^d\times\mathbb{R}^d_{++} \\ (\bar{m}_t)_{t=1}^T \in \mathbb{R}^{d\times T}}} nT\Bigg\{\frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n} \ell_{t,i}(\bar{m}_t) + \\ \frac{\eta}{T}\phi(\mu,v) + \frac{\lambda}{T}\sum_{t=1}^{T}\psi_{\mu,v}(\bar{m}_t) + \frac{1}{2n\lambda T}\sum_{t=1}^{T}\|v_t\|_\infty \sum_{i=1}^{n}\frac{||\alpha_{t,i}||_2^2}{i} + \frac{s^2}{2\eta T}\sum_{t=1}^{T}||\nabla_*^t||_2^2\Bigg\}.\end{aligned} \tag{4.17}$$

Since ψ_{μ_t,v_t} is always positive we can withdraw it from the left hand side. Then, the $\ell_{t,i}$ are L-Lipschitz which implies $\|\alpha_{t,i}\|_2 \leq L$, for all $(t,i) \in [T]\times[n]$, therefore,

$$\sum_{i=1}^{n}\frac{||\alpha_{t,i}||_2^2}{i} \leq L^2(1+\log(n)).$$

And in Proposition 4.4 we proved $\|\nabla_*^t\|_2^2 \leq L^2 + \left(\frac{L}{\lambda} + \lambda\sqrt{d}\right)^2$. By plugging the gradient

bounds and the values of the regularisation functions, it leads to,

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n}\ell_{t,i}(m_{t,i}) \leq \inf_{\substack{(\mu,v)\in\mathbb{R}^d\times\mathbb{R}^d_{++}\\(\bar{m}_t)_{t=1}^T\in\mathbb{R}^{d\times T}}} &\sum_{t=1}^{T}\sum_{i=1}^{n}\ell_{t,i}(\bar{m}_t) + \frac{n\eta}{2s^2}\left\{\|\mu\|_2^2 + \|v - s^2 1_d\|_2^2\right\}\\ &+ n\lambda\sum_{t=1}^{T}\left\{\sum_{i=1}^{d}\frac{(\bar{m}_{t,i}-\mu_i)^2}{2v_i} + v_i\right\} + \frac{L^2(1+\log(n))}{2\lambda}\sum_{t=1}^{T}\|v_t\|_\infty\\ &+ \frac{nTs^2}{2\eta}\left(L^2 + \left(\frac{L}{\lambda} + \lambda\sqrt{d}\right)^2\right).\end{aligned} \tag{4.18}$$

Take $\mu = \sum_{t=1}^{T}\frac{\bar{m}_t}{T} =: \bar{\bar{m}}$, $v = \bar{v}1_d$, $\bar{v}\in\mathbb{R}_{++}$ and use the notation $\sigma^2_{\bar{m}} := \sum_{t=1}^{T}\frac{\|\bar{m}_t - \bar{\bar{m}}\|_2^2}{T}$,

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n}\ell_{t,i}(m_{t,i}) \leq \inf_{\substack{\bar{v}\in\mathbb{R}_{++}\\(\bar{m}_t)_{t=1}^T\in(\mathbb{R})^T}} &\sum_{t=1}^{T}\sum_{i=1}^{n}\ell_{t,i}(\bar{m}_t) + \frac{n\eta}{2s^2}\left\{\|\bar{\bar{m}}\|_2^2 + d(\bar{v}-s^2)^2\right\}\\ &+ n\lambda T\left(\frac{\sigma^2_{\bar{m}}}{2\bar{v}} + dv\right) + \frac{L^2(1+\log(n))}{2\lambda}\sum_{t=1}^{T}\|v_t\|_\infty + \frac{nTs^2}{2\eta}\left(L^2 + \left(\frac{L}{\lambda} + \lambda\sqrt{d}\right)^2\right).\end{aligned} \tag{4.19}$$

We take $\lambda = \max\left(T^{-1/4}, \sqrt{\frac{\log(n)}{n}}\right)$ and $\eta = \frac{\sqrt{T}}{\lambda}$. Notice that $\lambda^{-1} \leq \min\left(T^{1/4}, \sqrt{\frac{n}{\log(n)}}\right)$. We make the hypothesis that $\sum_{t=1}^{T}\|v_t\|_\infty = O(T)$, so that we can find $K \geq 0$ such that $\sum_{t=1}^{T}\|v_t\|_\infty \leq KT$. It leads to,

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n}\ell_{t,i}(m_{t,i}) \leq \inf_{\substack{\bar{v}\in\mathbb{R}_{++}\\(\bar{m}_t)_{t=1}^T\in(\mathbb{R})^T}} &\sum_{t=1}^{T}\sum_{i=1}^{n}\ell_{t,i}(\bar{m}_t)\\ &+ \frac{1}{2s^2}\left\{\|\bar{\bar{m}}\|_2^2 + d(\bar{v}-s^2)^2\right\}\min\left(nT^{3/4}, n\sqrt{\frac{Tn}{\log(n)}}\right)\\ &+ \left(\frac{\sigma^2_{\bar{m}}}{2\bar{v}} + d\bar{v}\right)\max\left(nT^{3/4}, T\sqrt{n\log(n)}\right)\\ &+ \frac{KL^2}{2}\min\left(T^{5/4}\log(n), T\sqrt{n\log(n)}\right)\\ &+ Cs^2\min\left(nT^{3/4}, n\sqrt{\frac{Tn}{\log(n)}}\right).\end{aligned} \tag{4.20}$$

where C is a constant that depends only on L and d. There are now two ways to formulate the bound. We place oursevles in the regime where $n \ll T$, C_1, C_2 and C_3 are quantities

that do not depend on n, T, $\bar{v}$ and $(\bar{m}_t)_{t=1}^T$,

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n}\ell_{t,i}(m_{t,i}) \leq \inf_{\substack{\bar{v}\in\mathbb{R}_{++}\\(\bar{m}_t)_{t=1}^T\in(\mathbb{R})^T}} \sum_{t=1}^{T}\sum_{i=1}^{n}\ell_{t,i}(\bar{m}_t) + C_1 nT^{3/4}\left\{\|\bar{\bar{m}}\|_2^2 + \bar{v}^2\right\} \\ + C_2 T\sqrt{n\log(n)}\left(\frac{\sigma_{\bar{m}}^2}{\bar{v}} + \bar{v}\right) + C_3 T\sqrt{n\log(n)}.\end{aligned} \tag{4.21}$$

Take $(\bar{m}_t)_{t=1}^T \in (\mathbb{R})^T$ such that they minimise the term $\sum_{t=1}^T\sum_{i=1}^n \ell_{t,i}(\bar{m}_t)$ and take any $\bar{v}\in\mathbb{R}_{++}$,

$$\mathcal{R}_{meta}\left((Z)_{t=1}^T\right) \leq O(nT^{3/4}) + O(T\sqrt{n\log(n)}) + \sigma_{\bar{m}}^2 O(T\sqrt{n\log(n)}). \tag{4.22}$$

□

Comment on the bound. The meta regret bound is to compare with Independant Task Learning (ITL) where each task is trained in isolation. Running an Online algorithm for each task generates a regret in $O(\sqrt{n})$. Therefore training on T tasks in isolation gives a regret in $O(T\sqrt{n})$. Our bound is sublinear in NT but admits a worse regret than $O(T\sqrt{n})$ showing no improvement over ITL). This source of suboptimality could either come from the choice of regularisation functions (as we did not use Kullback Leibler divergenes that is natural to compare distributions) or the proof technique.

Chapter 5

Conclusion

Our objective was to present a theoretically grounded framework for Meta Learning in the online setting. We successfully introduced a general algorithm in Chapter 3 for variational learning as an extension of an existing frequentist approach. Its current practical implementation is suboptimal as it leads to a worse regret bound than the Independent Task Learning. However, we stress that it does not mean that the underlying algorithm is not worth considering. It is likely that the source of suboptimality in the bound comes from the proof technique and we conjecture that it can be improved.

It lefts the door open for many improvements. For example, the choice of the regularisation functions is subject to discussion, it is still not clear what should be the best configuration in the Gaussian setting. We have two options to improve them, either we find a better configuration or we find a way to use the Kullback-Leibler divergence despite the violated convex assumptions. Another interesting direction is the Natural Gradient formulation of Variational Inference. When the variational family is exponential, the parametrisation with the natural parameter leads to a Natural Gradient Descent which is a Gradient Descent locally pre-conditioned by the inverse of the Fisher Information Matrix. The Fisher Information Matrix is hard to get in high dimension but a recent breakthrough was to use the expectation parametrisation instead of the natural parametrisation which allows to avoid the computation of the Fisher Information Matrix and also connects the process to a Mirror Descent Khan and Nielsen (2018). It scales well and gives better results than standard Variational Inference. Unfortunately, the expectation parametrisation for an exponential family is almost never convex so its theoretical analysis is still an open question.

Among the questions we did not address was the question of when Meta Learning is guaranteed to be helpful? For linear regression tasks, it can be shown that if their true parameters lie close to each other, Meta Learning will win over Independent Task Learning. But for more complex models it is less clear. The regret for Independent Task Learning is in $O(T\sqrt{N})$. Our dream regret bound for Meta Learning has the form $S \times O(f(N,T)) + O(N\sqrt{T})$ where S is a coefficient that goes to zero when the tasks share strong similarities and $O(f(N,T))$ is potentially worse than $O(T\sqrt{N})$. Therefore, in the setting where the tasks share strong similarities (i.e. $S \approx 0$) and $N \ll T$ we have an advantage with Meta Learning. The difficulty is to define S.

Last but not least, no numerical experiment has been introduced in the thesis. The remaining month of the internship will be entirely devoted to them and to improve the bound. We will experiment with 2OGD on simulated and real datasets. Hopefully, it will lead to new hindsights to improve the algorithm.

Bibliography

Pierre Alquier. Non-exponentially weighted aggregation: regret bounds for unbounded loss functions. *arXiv preprint arXiv:2009.03017*, 2020.

Pierre Alquier and Benjamin Guedj. Simpler pac-bayesian bounds for hostile data. *Machine Learning*, 107(5):887–902, 2018.

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of gibbs posteriors. *Journal of Machine Learning Research*, 17(236): 1–41, 2016.

Pierre Alquier, Massimiliano Pontil, et al. Regret bounds for lifelong learning. In *Artificial Intelligence and Statistics*, pages 261–269, 2017.

Pierre Alquier, James Ridgway, et al. Concentration of tempered posteriors and of their variational approximations. *Annals of Statistics*, 48(3):1475–1497, 2020.

Jonathan Baxter. Theoretical models of learning to learn. In *Learning to learn*, pages 71–94. Springer, 1998.

Anirban Bhattacharya, Debdeep Pati, Yun Yang, et al. Bayesian fractional posteriors. *The Annals of Statistics*, 47(1):39–66, 2019.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

Olivier Catoni. *Statistical learning theory and stochastic optimization: Ecole d'Eté de Probabilités de Saint-Flour XXXI-2001*. Springer, 2004.

Badr-Eddine Chérief-Abdellatif, Pierre Alquier, and Mohammad Emtiyaz Khan. A generalization bound for online variational inference. In *Proceedings of The Eleventh Asian Conference on Machine Learning*, volume 101 of *Proceedings of Machine Learning Research*, pages 662–677, 2019.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 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.

Justin Domke. Provable smoothness guarantees for black-box variational inference. *arXiv preprint arXiv:1901.08431*, 2019.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. *arXiv preprint arXiv:1902.08438*, 2019.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

Benjamin Guedj. A primer on pac-bayesian learning. In *Proceedings of the second congress of the French Mathematical Society*, 2019.

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

M.E. Khan and H. Rue. Learning-algorithms from bayesian principles. *draft*, 2019. URL https://emtiyaz.github.io/papers/learning_from_bayes.pdf.

Mohammad Emtiyaz Khan and Didrik Nielsen. Fast yet simple natural-gradient descent for variational inference in complex models. In *2018 International Symposium on Information Theory and Its Applications (ISITA)*, pages 31–35. IEEE, 2018.

Mohammad Emtiyaz Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*, 2018.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*, 2019.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. In *Advances in neural information processing systems*, pages 4287–4299, 2019.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard E Turner, and Mohammad Emtiyaz Khan. Continual deep learning by functional regularisation of memorable past. *arXiv preprint arXiv:2004.14070*, 2020.

Juan Peypouquet. *Convex optimization in normed spaces: theory, methods and examples*. Springer, 2015.

Mark D Reid, Rafael M Frongillo, Robert C Williamson, and Nishant Mehta. Generalized mixability via entropic duality. In *Conference on Learning Theory*, pages 1501–1522, 2015.

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 and Yoram Singer. Online learning: Theory, algorithms, and applications. *PhD Thesis*, 2007c.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Nisheeth K. Vishnoi. Lecture notes algorithms for convex optimization., 2020.

Makoto Yamada, Takeuchi Koh, Tomoharu Iwata, John Shawe-Taylor, and Samuel Kaski. Localized lasso for high-dimensional regression. In *Artificial Intelligence and Statistics*, pages 325–333. PMLR, 2017.

Appendix A

Technical Results

A.1 Tools

A.1.1 Convexity Results and definitions

Definition A.1. *(strong convexity) Let $\mu > 0$ and $||.||$ an arbitrary norm on $\mathbb{R}^d$, $f : \mathbb{R}^d \to \mathbb{R}$ is $\mu-$strongly convex w.r.t to $||.||$ if and only if*

$$f(x) \geq f(y) + g^T(x-y) + \frac{\mu}{2}||x-y||^2 \qquad \forall x, y \in \mathbb{R}^d, g \in \partial f(y).$$

The following is also equivalent,

$$f(\alpha x + (1-\alpha)v) \leq \alpha f(x) + (1-\alpha)f(y) - \frac{\sigma}{2}\alpha(1-\alpha)||x-y||^2, \qquad \forall x, y \in \mathbb{R}^d, \alpha \in (0,1).$$

Proposition A.1. *$\mu > 0$, $f : \mathbb{R}^d \to \mathbb{R}$ is $\mu-$strongly convex w.r.t to the Euclidean norm if and only if $g(x) = f(x) - \frac{\mu}{2}\|x\|_2^2$ is convex $\forall x$. Hence, if f is twice differentiable, f is $\mu-$strongly convex w.r.t to the Euclidean norm if and only if $\lambda_{min}(\nabla^2 f(x)) \geq \mu$.*

Definition A.2 (Subdifferential)**.** *$f : \mathbb{R}^d \to \mathbb{R}$, $x \in \mathbb{R}^d$,*

$$\partial f(x) = \left\{ g \in \mathbb{R}^d : f(x') \geq f(x) + \langle g, x' - x \rangle, \text{ for any } x' \in \mathrm{Dom} f \right\}.$$

Proposition A.2. *If $g(x,y)$ is jointly convex on $\mathcal{X} \times \mathcal{Y}$ then $h(x) = \inf_{y \in \mathcal{Y}} g(x,y)$ is convex on $\mathcal{X}$.*

Proof. Let $x, x' \in Dom(h)$, $x'' := \lambda x + (1-\lambda)x'$, $\lambda \in (0,1)$, we show that $h(x'') \leq \lambda h(x) + (1-\lambda)h(x')$. By definition of the infimum, for all $\epsilon > 0$, $\exists y, y'$ such that $g(x,y) \leq h(x) + \epsilon$ and $g(x',y) \leq h(x') + \epsilon$. Then,

$$\begin{aligned} h(x'') &= \inf_{y \in \mathcal{Y}} g(x'', y) \\ &\leq g(x'', \lambda y + (1-\lambda)y') \\ &\leq \lambda g(x,y) + (1-\lambda) g(x', y') \qquad \text{by joint convexity} \\ &\leq \lambda h(x) + (1-\lambda)h(x') + \epsilon \end{aligned} \tag{A.1}$$

□

A.1.2 Fenchel Transform

Definition A.3 (Convex conjugate also known as the Fenchel transform)**.** *The Fenchel transform of a mapping* $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ *is*

$$f^*(\alpha) = \sup_{x \in \mathbb{R}^d} \langle \alpha, x \rangle - f(x), \qquad \forall \alpha \in \mathbb{R}^d. \tag{A.2}$$

Proposition A.3. *If* f *is strictly convex, then* f^* *is differentiable and*

$$\nabla f^*(\alpha) = \arg\max_{x \in \mathbb{R}^d} \alpha^T x - f(x), \qquad \forall \alpha \in \mathbb{R}^d. \tag{A.3}$$

Proposition A.4 (Example: translated and rescaled Euclidean Norm)**.**

$$\begin{aligned} f :& \mathbb{R}^d \to \mathbb{R}, \qquad x \mapsto \frac{c}{2}\|x - y\|_2^2, \qquad c > 0, y \in \mathbb{R}^d \\ f^* :& \mathbb{R}^d \to \mathbb{R}, \qquad \alpha \mapsto \frac{1}{2c}\|\alpha\|_2^2 + \langle \alpha, y \rangle \end{aligned}$$

Proof.

$$\begin{aligned} \arg\max_{x \in \mathbb{R}^d} \alpha^T x - f_(x) &= \arg\max_{x \in \mathbb{R}^d} \alpha^T x - \frac{c}{2}||x||_2^2 + cx^T y \\ &= \arg\max_{x \in \mathbb{R}^d} \left(\frac{\alpha}{c}\right)^T x - \frac{1}{2}||x||_2^2 + x^T y \\ &= \arg\max_{x \in \mathbb{R}^d} -\frac{1}{2}||y + \frac{\alpha}{c} - x||_2^2 = y + \frac{\alpha}{c} \end{aligned}$$

Thus,

$$\begin{aligned} f^*(\alpha) = \sup_{x \in \mathbb{R}^d} \alpha^T x - f(x) &= (y + \frac{\alpha}{c})^T \alpha - \frac{1}{2c}||\alpha||_2^2 \\ &= \frac{1}{2c}||\alpha||_2^2 + \alpha^T y \end{aligned}$$

□

Taking $c = 1$ and $y = 0$, it shows that the Euclidean norm is self-dual.

A.1.3 Duality

A.1.4 f-Divergences and Kullback-Leibler Divergence

Divergences are common tools to compare probability distributions and the family of $f-$divergences covers a large part of them.

Definition A.4 ($f-$divergences)**.** *If $(\mathcal{X}, \mathcal{F})$ is an arbitrary measurable space and P and Q are two probability distributions on $(\mathcal{X}, \mathcal{F})$ such that $P \ll Q$ (i.e. P is absolutely continuous with respect to Q, $\forall A \in \mathcal{F}, Q(A) = 0 \implies P(A) = 0$). We define by $\mathcal{P}(\mathcal{X}, \mathcal{F})$ the set of probability measures on $(\mathcal{X}, \mathcal{F})$. Then, for a convex function f such that $f(1) = 0$, the f-divergence of P from Q is defined as,*

$$D_f(P \parallel Q) \equiv \int_{\mathcal{X}} f\left(\frac{dP}{dQ}\right) dQ$$

Where $\frac{dP}{dQ}$ is the Radon-Nikodym derivative of P with respect to Q that exists since $P \ll Q$. If P is not absolutely continuous with respect to Q, we set $D_f(P \parallel Q) = +\infty$.

Proposition A.5 (Positive definiteness of f-divergences)**.** $\forall P, Q \in \mathcal{P}(\mathcal{X}, \mathcal{F})$
$D_f(P||Q) \geq 0$ *and* $D_f(P||Q) = 0 \iff P = Q$

Proof. if P is not absolutely continuous with respect to Q, $P \neq Q$ and $D_f(P||Q) = +\infty$, therefore the statement is true. If we assume $P \ll Q$, since f is convex, and Q is a probability distribution by Jensen's inequality,

$$D_f(P||Q) = \mathbb{E}_{X\sim Q}\left[f\left(\frac{dP}{dQ}(X)\right)\right] \geq f\left(\mathbb{E}_Q\left[\frac{dP}{dQ}(X)\right]\right) = f\left(\int_{\mathcal{X}} \frac{dP}{dQ} dQ\right) = f(\underbrace{P(\mathcal{X})}_{=1}) = 0$$

where we use the definition of the Radon Nikodym derivative: if $P \ll Q$, $\frac{dP}{dQ}$ is the ($Q-$a.e) unique function such as $P(A) = \int_{\mathcal{A}} \frac{dP}{dQ} dQ \quad \forall A \in \mathcal{F}$.

If $D_f(P||Q) = 0$, the Jensen's inequality is saturated, therefore $\frac{dP}{dQ}$ is equal to a constant Q-a.e., integrating over $\mathcal{X}$ we get that the constant is equal to 1. Thus, $\forall A \in \mathcal{F}$, $P(A) = \int_A \frac{dP}{dQ} dQ = \int_A dQ = Q(A)$. It proves that $P = Q$.

If $P = Q$, it is straightforward to see that $\frac{dP}{dQ} = 1$ Q-a.e satisfies the Radon Nikodym definition and therefore since $f(1) = 0$, we have $D_f(P||Q) = 0$. □

Definition A.5 (Kullback-Leibler Divergence)**.** *When f is the mapping $x \mapsto x\log(x)$, $D_f(P||Q) := D_{KL}(P \parallel Q)$ is the KL divergence*

$$KL(P||Q) \equiv \begin{cases} \int \frac{dP}{dQ} \log\left(\frac{dP}{dQ}\right) dQ & \text{if } P \ll Q \\ +\infty & \text{otherwise} \end{cases}$$

Proposition A.6 (Kullback Leibler Divergence between Gaussian Distributions)**.** *If $X \sim \mathcal{N}_d(\mu_1, \Sigma_1)$, $Y \sim \mathcal{N}_d(\mu_2, \Sigma_2)$ and Σ_1, Σ_2 are non singular, then*

$$KL(X||Y) = \frac{1}{2}\left(\log\frac{|\Sigma_2|}{|\Sigma_1|} - d + Tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T\Sigma_2^{-1}(\mu_2 - \mu_1)\right)$$

Thus if $X \sim \mathcal{N}_d(\mu_1, Diag(\sigma_1^2))$, $Y \sim \mathcal{N}_d(\mu_2, Diag(\sigma_2^2))$, $\sigma_1 \in \mathbb{R}^d_{++}$, $\sigma_2 \in \mathbb{R}^d_{++}$

$$KL(X||Y) = \frac{1}{2}\sum_{i=1}^{d}\left(\ln\left(\frac{\sigma_{2,i}^2}{\sigma_{1,i}^2}\right) - 1 + \frac{\sigma_{1,i}^2}{\sigma_{2,i}^2} + \frac{(\mu_{2,i} - \mu_{1,i})^2}{\sigma_{2,i}^2}\right)$$

Proposition A.7 (Convexity for the Gaussian KL divergence)**.** *$X \sim \mathcal{N}_d(\mu_1, Diag(\sigma_1^2))$, $Y \sim \mathcal{N}_d(\mu_2, Diag(\sigma_2^2))$, $\sigma_1 \in \mathbb{R}^d_{++}$, $\sigma_2 \in \mathbb{R}^d_{++}$. $KL(X||Y)$ is*

(i) strictly convex as a function (μ_1, σ_1^2) (but not strongly convex)

(ii) $\|\sigma_2^2\|_\infty^{-1}$-strongly convex as a function (μ_1, σ_1)

(iii) not jointly convex as a function of $(\mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$ or $(\mu_1, \sigma_1, \mu_2, \sigma_2)$ as it is not convex in σ_2 of σ_2^2

A.2 Theorems

Theorem A.1 (Regret bound EWA, Theorem 1 in Chérief-Abdellatif et al. (2019))**.** *Assuming that the loss is bounded, i.e., $0 \leq \ell_i(w) \leq B, \forall i \geq 1, w \in \mathbb{R}^d$, the expected regret has the following upper bound*

$$\sum_{i=1}^{n} \mathbb{E}_{w\sim\rho_i}\left[\ell_i\left(w\right)\right] \leq \inf_{\rho\in\mathcal{P}(\mathbb{R}^d)} \left\{ \mathbb{E}_{w\sim p}\left[\sum_{i=1}^{n} \ell_i(w)\right] + \frac{nB^2}{8\lambda} + \lambda KL(\rho,\pi) \right\}$$

If in addition ℓ_i is convex for all $i \geq 1$, the cumulative regret has the following upper bound when $\hat{w}_i = \mathbb{E}_{w\sim\rho_i}[w]$ is the posterior mean,

$$\sum_{i=1}^{n} \ell_i\left(\hat{w}_i\right) \leq \inf_{\rho\in\mathcal{P}(\mathbb{R}^d)} \left\{ \mathbb{E}_{w\sim p}\left[\sum_{i=1}^{n} \ell_i(w)\right] + \frac{nB^2}{8\lambda} + \lambda KL(\rho,\pi) \right\}$$