Learning from Dependent Data

by

Alexander Zimin

September, 2018

A thesis presented to the
Graduate School
of the
Institute of Science and Technology Austria, Klosterneuburg, Austria
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Institute of Science and Technology

Abstract

The most common assumption made in statistical learning theory is the assumption of the independent and identically distributed (i.i.d.) data. While being very convenient mathematically, it is often very clearly violated in practice. This disparity between the machine learning theory and applications underlies a growing demand in the development of algorithms that learn from dependent data and theory that can provide generalization guarantees similar to the independent situations.

This thesis is dedicated to two variants of dependencies that can arise in practice. One is a dependence on the level of samples in a single learning task. Another dependency type arises in the multi-task setting when the tasks are dependent on each other even though the data for them can be i.i.d. In both cases we model the data (samples or tasks) as stochastic processes and introduce new algorithms for both settings that take into account and exploit the resulting dependencies. We prove the theoretical guarantees on the performance of the introduced algorithms under different evaluation criteria and, in addition, we compliment the theoretical study by the empirical one, where we evaluate some of the algorithms on two real world datasets to highlight their practical applicability.

Acknowledgments

First, I want to thank Christoph for making all this possible, for encouragement to explore research topics and for keeping my side during the ups and downs of publishing process. Thank you for tolerance with my articles and for all the late hours we spent editing the submissions over and over.

I would like to thank my committee members, Jan and Liva. Thank you Liva for agreeing to come for the qualifying exam and keeping in touch afterwards. Thank you Jan for agreeing to be the last minute substitute and then actually staying with me for the rest of my PhD.

A lot of great memories are shared with fellow students, post-docs and interns at IST including, but not limited to Asya, Amélie, Csaba, Emilie, Georg, Harald, Ilja, Jan, Kristóf, Mary, Nathaniel, Nikola, Tomas, Victoria and all of the members of IST football team that I spent so much time playing with. Of course, special thanks to Alex and Michal, you have been the best office mates and I miss all of the things we did together.

I want to give my warmest thanks to my parents, whose vision and unconditional support made me so far.

My greatest gratitude goes to Anastasiia who shared this toughest journey with me. I am so lucky to have met such a wonderful and supportive woman.

This thesis was partially funded by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 308036. From 2013 to 2016 I have been an OMV scholar.

About the Author

Alexander Zimin received a Bachelor degree from Yaroslavl State University in Russia. Afterwards, he obtained his Master degree from Central European University in 2013 under supervision of Laszlo Gyorfi and Gergely Neu. Now he is a PhD student at IST Austria working in the group of Christoph Lampert. His research is focused on various machine learning scenarios, which involve dependent data. Previously he has also worked on online learning and reinforcement learning.

Table of Contents

List of Figures

1 Introduction

In the course of its development, the area of machine learning heavily relied on theoretical analysis of learning problems. It helps us to understand the limits, to explain the behavior of existing algorithms and even to motivate new algorithms. In the basis of every theoretical study are the assumptions about the nature of the real world. The most common assumption made in statistical learning theory is the assumption of the independent and identically distributed (i.i.d.) data. While very convenient mathematically, it is often very clearly violated in practice. We can observe this even in the famous textbook example of classifying e-mails into *ham* or *spam*. For example, when multiple emails are exchanged with the same writer, the contents of later emails depends on the contents of earlier ones. This disparity between the theory and applications underlies a growing demand in the development of algorithms that learn from dependent data and theory that can provide generalization guarantees similar to the independent situations.

This thesis is dedicated to two variants of dependencies that can arise in practice. One is a dependence on the level of samples, like in the example of e-mail conversations. Another dependency type arises in the multi-task setting when the task are dependent on each other even though the data within each task can be i.i.d. For example, different e-mail users sometimes are modelled as different learning tasks and there can be a group of users whose e-mails are interdependent: these users can belong to the same mailing lists and receive the same newsletters or they can send e-mails with similar content to the same addresses (e.g. registration e-mails for some events).

From a statistical point of view, in both cases we model the data (samples or tasks) as *stochastic processes*, i.e. data sources that have a notion of time and usually are observed in a sequential manner. Many practical problems, even spam classification, are inherently online and a lot of structure can be observed by taking time into consideration. Contrary to the

most prior research, we attempt to analyze the problems for general processes without making a specific assumptions like stationarity or ergodicity. Note that i.i.d. data sources are trivial stochastic processes and hence are automatically included in the studies.

In Chapter 3 we use stochastic processes as a model for the samples in the training set. We turn our attention to the mostly unstudied problem of the conditional risk minimization, where the goal is to optimize the performance of the learner on each step conditioned on the observations seen so far. This is contrary to more well studied marginal learning, where the goal is typically to optimize the performance on the average over all possible realizations of the process. As the set of the observations we condition on changes at every step of the learning process, the existing theoretical framework of statistical learning theory does not fit the problem. Thus, we develop a new framework that focuses on a new notion of conditional learnability that requires the performance of the algorithm to improve with the amount of the observed data regardless the changing goals. One of our main insights is the crucial role of the individual discrepancies, a notion of a distance between the conditional distributions of the process. We provide a dichotomy of the problem based on the behavior of these discrepancies. In the convergent case, summarized in Theorem 3.7.2, we show that the Empirical Risk Minimization (ERM) algorithm is sufficient to achieve learnability, thus generalizing the existing results for i.i.d. data. In the non-convergent case, in the situations when we have an additional information in the form of the upper bounds on the discrepancies, we introduce two new algorithms: weighted empirical risk minimization (WERM) and MACRO. In Theorem 3.8.4 we prove that WERM is able to achieve conditional learnability in the limit for wide range of stochastic processes. However, WERM has not very favorable computational complexity that motivates us to introduce another algorithm, MACRO, that has a linear runtime complexity in the size of the dataset. Moreover, in Theorems 3.8.8, 3.8.9 and 3.8.10 we prove that MACRO achieves a modified conditional learnability under weaker assumptions than WERM. The results of this chapter are based on two papers in collaboration with Christoph Lampert: "Learning Theory for Conditional Risk Minimization", appeared in AISTATS 2017, and "MACRO: A Meta-Algorithm for Conditional Risk Minimization".

Chapter 4 shows how the principles of the conditional risk minimization can be applied in practice. We conduct experiments with two datasets for the MACRO algorithm studied theoretically in the previous chapter. We compare the performance of MACRO to two algorithms that represent the traditional approaches to learning: statistical learning theory and

online learning. For both datasets we show that MACRO is able to improve the classification accuracy on the given tasks for a wide range of hyper-parameters. In addition, we study empirically two versions of MACRO algorithm that choose the hyper-parameters on the fly and show that both are viable strategies when the necessary computational resources are available. The results of this chapter are based on "MACRO: A Meta-Algorithm for Conditional Risk Minimization", a joint work with Christoph Lampert.

In Chapter 5 we turn our attention to multi-task setting where stochastic processes are used to model a sequence of tasks that a learner is faced with. We present a new algorithm, MTLAB, that is based on the idea of running an online learner on the data of all tasks combined and then performing a specific online-to-batch conversion for each task. In Theorem 5.2.1, utilizing the PAC-Bayes framework, we prove a regret bound with respect to the best fixed Gibbs predictor chosen in hindsight. In addition, in Theorems 5.6.4 and 5.6.6 we show that by estimating the discrepancies between the tasks (either from labelled or un-labelled data), one can use a MACRO-style algorithm that utilizes MTLAB and prove strong performance guarantees for each individual task. The results presented in this chapter are based on the joint work with Christoph Lampert: "Tasks Without Borders: A New Approach to Online Multi-Task Learning".

2 Background

In this chapter we introduce all the necessary background on machine learning and stochastic processes.

2.1 PAC learning

In this section we introduce the standard framework for the statistical learning theory. For this let us go back to the example of the introduction and formalize its components. We want to find a predictor that can classify e-mails into spam or ham. Therefore, the input space, that we denote as $\mathcal{X}$, is considered to be the space of all e-mails. Each message $x \in \mathcal{X}$ has an associated label $y \in \mathcal{Y}$ from some label space $\mathcal{Y}$, that can be $\{0, 1\}$ in the spam/ham case. The joint input-label space will be denoted as $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The predictor (or hypothesis) h is a function that takes an e-mail x as an input and produces an output in its own output space $\mathcal{D}$, i.e. $h : \mathcal{X} \to \mathcal{D}$. In the example, $\mathcal{D}$ can be taken to be $[0, 1]$. All of the possible predictors constitute a space of possible hypotheses $\mathcal{H} \subseteq \{h : \mathcal{X} \to \mathcal{D}\}$. The quality of the prediction is assessed by a loss function $\ell : \mathcal{D} \times \mathcal{Y} \to [0, 1]$. There is a number of options for the loss functions and often this is a design choice of the problem to choose an appropriate loss function. The popular choices are l_2-loss $\ell(d, y) = (d - y)^2$, log-loss $\ell(d, y) = -y \log d - (1 - y) \log(1 - d)$ and, when $\mathcal{D} = \{0, 1\}$, 0-1 loss $\ell(d, y) = \mathbb{I}[d \neq y]$. We will however use a shorthand notation for the loss of the hypothesis and write $\ell(h, z) = \ell(h(x), y)$ for a hypothesis h and a sample $z = (x, y)$. In addition, we will often work with the induced function space $\mathcal{L}(\mathcal{H}) = \{\ell(h, \cdot), \forall h \in \mathcal{H}\}$.

The standard statistical learning theory assumes that there is an underlying distribution D that all images (and their labels) are sampled from. Moreover, the standard assumptions is that the images are sampled independently. The goal of a learner is to find a hypothesis

$h \in \mathcal{H}$ that will have a small loss on a newly sampled data point from the same distribution D. This is formalized by the notion of a *risk*:

$$R(h, D) = \mathbb{E}_{z \sim D} \left[\ell(h, z) \right]. \tag{2.1}$$

Hence, the ultimate task for the learner is to perform the following optimization:

$$\min_{h \in \mathcal{H}} R(h, D). \tag{2.2}$$

However, the learner does not have an access to the distribution D. Rather, we have an access to a dataset S of images sampled from D: $S = \{z_i\}_{i=1}^n$. Then any learning algorithm $\mathcal{A}$ is defined as a function that takes a dataset S and produces a predictor $h_{\mathcal{A}}$.

The central notion that statistical learning theory studies is learnability of different hypotheses classes. The following definition, introduced in [Valiant, 1984], specifies what learnability means exactly.

Definition 2.1.1 (PAC-learnability). *A class of hypotheses $\mathcal{H}$ is called a* Probably Approximately Correct *(PAC) learnable if there exists an algorithm $\mathcal{A}$ and a function $n_{\mathcal{A}} : (0,1) \times (0,1) \to \mathbb{N}$, such that for any $\varepsilon > 0$ and $\delta > 0$, all distributions D and all $n \geq n_{\mathcal{A}}(\varepsilon, \delta)$, with probability of $1 - \delta$ over the sampling of $S \sim D^n$*

$$R(h_{\mathcal{A}}, D) \leq \inf_{h \in \mathcal{H}} R(h, D) + \varepsilon. \tag{2.3}$$

Such an algorithm $\mathcal{A}$ is called a PAC-learner.

A typical algorithm that is usually used to prove learnability of a particular hypotheses class is an Empirical Risk Minimization (ERM). Given a dataset S, ERM learner chooses a hypothesis that minimizes the empirical risk:

$$h_{\text{ERM}} = \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{n} \sum_{z \in S} \ell(h, z). \tag{2.4}$$

The standard tool for studying the learnability by ERM learner is a uniform convergence of the empirical risk from the true risk that is characterized by

$$\mathcal{U}_n(\mathcal{H}) = \sup_{h \in \mathcal{H}} \left| R(h, D) - \frac{1}{n} \sum_{z \in S} \ell(h, z) \right|. \tag{2.5}$$

In fact, the convergence rate of $\mathcal{U}_n(\mathcal{H})$ controls the convergence rate of ERM learner that can be seen from the following result.

Lemma 2.1.2. *For any hypotheses class $\mathcal{H}$ and any sample $S \sim D^n$*

$$R(h_{\mathcal{A}}, D) - \inf_{h \in \mathcal{H}} R(h, D) \leq 2\mathcal{U}_n(\mathcal{H}). \tag{2.6}$$

The answer to the question whether a uniform convergence for a particular class $\mathcal{H}$ can be achieved or not depends on the "richness" of $\mathcal{H}$. There have been a number of ways to describe richness of a class of functions and we will discuss some of them in the next section.

2.2 Complexity measures

The standard way to show uniform convergence of a certain class of hypotheses relies on two main technical ingredients: concentration inequalities and finite approximations of the hypotheses class. In this section we describe the complexity measures that control the quality of different approximations of $\mathcal{H}$. We start with covering numbers.

Definition 2.2.1. *For a function class $\mathcal{F} \subseteq \{f : \mathcal{Z} \to \mathbb{R}\}$ and a metric Δ on $\mathcal{F}$, every finite collection of functions $f_1, \ldots, f_N : \mathcal{Z} \to \mathbb{R}$ with the property that for every $f \in \mathcal{F}$, there is a $j \in [N]$ such that*

$$\Delta(f, f_j) \leq \varepsilon \tag{2.7}$$

is called an ε-cover of $\mathcal{F}$ with respect to Δ.

The ε-covering number of $\mathcal{F}$ with respect to Δ is denoted as $\mathcal{N}_\Delta(\mathcal{F}, \varepsilon)$ and is equal to the size of the smallest ε-cover of $\mathcal{F}$.

In learning theory the most useful metrics are the empirical (pseudo-)metrics based on the sample. For a sample S of size n we define

$$\Delta_1^S(f, g) = \frac{1}{n} \sum_{z \in S} |f(z) - g(z)| \tag{2.8}$$

and

$$\Delta_\infty^S(f, g) = \sup_{z \in S} |f(z) - g(z)| . \tag{2.9}$$

We will denote by $\mathcal{N}_1(\mathcal{F}, \varepsilon, S)$ and $\mathcal{N}_\infty(\mathcal{F}, \varepsilon, S)$ the covering numbers with respect to Δ_1^S and Δ_∞^S respectively. An example of a bound that can be proven utilizing the covering numbers is the following theorem, that follows, for example, from Theorem 9.1 in [Györfi *et al.*, 2002].

Theorem 2.2.2. *For any $\delta > 0$, let $d = \mathbb{E}\mathcal{N}_1(\mathcal{L}(\mathcal{H}), 16\sqrt{\frac{2\log\frac{1}{\delta}}{n}}, S)$. Then, with probability $1-\delta$,*

$$\mathcal{U}_n(\mathcal{H}) \leq 8\sqrt{\frac{2\log 8d + 2\log\frac{1}{\delta}}{n}}, \tag{2.10}$$

The covering number $\mathcal{N}_1(\mathcal{F}, \varepsilon, S)$ can be related to more standard complexity measures like VC-dimension and fat-shattering dimension. We will start with the former.

Definition 2.2.3 ([Vapnik and Chervonenkis, 1971]). *A set of functions $\mathcal{F} \subseteq \{f : \mathcal{Z} \to \{0,1\}\}$ is said to shatter a set $S \in 2^{\mathcal{Z}}$ if the image of S under $\mathcal{H}$ is all possible $2^{|S|}$ labellings of S.*

The size of the largest set S that can be shattered by $\mathcal{H}$ is called a Vapnik-Chervonenkis (VC) dimension of $\mathcal{H}$ and is denoted as $vc(\mathcal{H})$.

The VC-dimension can be used directly to show uniform convergence for classes with the finite VC-dimension using direct argument. We, however, will give a bound on the covering number of $\mathcal{L}(\mathcal{H})$ in terms of VC-dimension of $\mathcal{H}$. First, note that if we use 0-1 loss and the decision space is $\mathcal{D} = \{0,1\}$, then we have $\mathcal{N}_1(\mathcal{L}(\mathcal{H}), \varepsilon, S) \leq \mathcal{N}_1(\mathcal{H}, \varepsilon, S)$. Second, the following lemma, that is a corollary of Vapnik-Chervonenkis-Sauer-Shelah lemma, [Sauer, 1972; Shelah, 1972; Vapnik and Chervonenkis, 1971], gives us the bound on the covering numbers of $\mathcal{H}$.

Lemma 2.2.4. *Let $\mathcal{H} \subseteq \{h : \mathcal{X} \to \{0,1\}\}$, then for any $\varepsilon \in (0,1)$ and any $n \in \mathbb{N}$*

$$\mathcal{N}_1(\mathcal{H}, \varepsilon, S) \leq (n+1)^{vc(\mathcal{H})}. \tag{2.11}$$

Additionally, for $n > vc(\mathcal{H})$

$$\mathcal{N}_1(\mathcal{H}, \varepsilon, S) \leq \left(\frac{en}{vc(\mathcal{H})}\right)^{vc(\mathcal{H})}. \tag{2.12}$$

VC-dimension works only for the classification tasks. For the real-valued predictions, the corresponding analog is a fat-shattering dimension.

Definition 2.2.5 ([Alon *et al.*, 1997; Bartlett *et al.*, 1996]). *A set of functions $\mathcal{F} \subseteq \{f : \mathcal{Z} \to [0,1]\}$ is said to γ-shatter a set $S \in 2^{\mathcal{Z}}$ for $\gamma > 0$ if there exist a function $s : \mathcal{Z} \to [0,1]$ such that for every subset $E \subseteq S$, there exists a function $f_E \in \mathcal{H}$ satisfying*

$$f_E(x) \leq s(x) - \gamma \text{ for } x \in S \setminus E \tag{2.13}$$

$$f_E(x) \geq s(x) + \gamma \text{ for } x \in E. \tag{2.14}$$

The size of the largest set S that can be γ-shattered by $\mathcal{F}$ is called a fat-shattering dimension of $\mathcal{F}$ at scale γ and is denoted as $fat_\gamma(\mathcal{F})$.

Similarly to Lemma 2.2.4, we can provide the bounds on the covering numbers in terms of the fat-shattering dimension.

Lemma 2.2.6 (Lemma 3.5 from [Alon *et al.*, 1997]). *For any $\varepsilon \in (0,1)$ let $d = fat_{\varepsilon/4}(\mathcal{F})$, then we have*

$$\mathcal{N}_\infty(\mathcal{F}, \varepsilon, S) \leq 2\left(\frac{2n}{\varepsilon^2}\right)^{d\log(2en/(d\varepsilon))} \tag{2.15}$$

Of course, covering numbers are not the only way to measure the complexity of a function class. Another measure that gives tighter bounds is a Rademacher complexity.

Definition 2.2.7. *Let $\sigma = \{\sigma_1, \ldots, \sigma_n\}$ be independent uniform random variables that take values in $\{-1, +1\}$. For a function class $\mathcal{F} \subseteq \{f : \mathcal{Z} \to [0,1]\}$ and a sample $S = \{z_1, \ldots, z_n\}$, the* Empirical Rademacher Complexity *of $\mathcal{F}$ with respect to S is defined as*

$$\hat{\mathfrak{R}}_S(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(z_i)\right]. \tag{2.16}$$

For any integer n, the Rademacher complexity *of $\mathcal{F}$ is defined as*

$$\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_{S\sim D^n}\left[\hat{\mathfrak{R}}_S(\mathcal{F})\right]. \tag{2.17}$$

The following result was first shown in [Koltchinskii, 2001].

Theorem 2.2.8. *For any $\delta > 0$, with probability $1-\delta$ we have*

$$\mathcal{U}_n(\mathcal{H}) \leq 2\mathfrak{R}_n(\mathcal{L}(\mathcal{H})) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}} \tag{2.18}$$

and

$$\mathcal{U}_n(\mathcal{H}) \leq 2\hat{\mathfrak{R}}_S(\mathcal{L}(\mathcal{H})) + 3\sqrt{\frac{\log\frac{1}{\delta}}{n}}. \tag{2.19}$$

The Rademacher complexity can be connected back to the covering numbers in various ways, e.g. using Dudley's theorem. As these connections are not relevant for the discussion in the present thesis, we refer the reader to [Bartlett and Mendelson, 2002] for further information.

2.3 PAC-Bayes framework

In this section, we consider a theory that is parallel to PAC-learning and that studies probabilistic predictors. A probabilistic Gibbs predictor is defined by a probability distribution Q over the hypotheses class. Upon receiving a new data point x, the Gibbs predictor samples a hypothesis $h \sim Q$ and outputs $h(x)$. The risk of such a predictor is

$$R(Q, D) = \mathbb{E}_{h\sim Q}\left[R(h, D)\right]. \tag{2.20}$$

The PAC-Bayesian theory concerns itself with the uniform bounds on R. The following result, e.g. from [McAllester, 1999], is an example of such a bound.

Theorem 2.3.1. *Let P be a fixed prior distribution over $\mathcal{H}$ chosen independently of data. For any $\delta > 0$, with probability $1 - \delta$ over the sampling of $S \sim D^n$, it holds uniformly over all Q*

$$R(Q, D) \leq \frac{1}{n}\sum_{z\in S}\mathbb{E}_{h\sim Q}\left[\ell(h, z)\right] + \frac{KL(Q|P) + \log\frac{1}{\delta}}{\sqrt{n}}. \tag{2.21}$$

Beside being used to provide a performance guarantee for ERM algorithm, Theorem 2.3.1 can be used directly to derive an algorithm that optimizes the right-hand side of the bound.

It may seem on the first sight that the bound of Theorem 2.3.1 lacks the dependence on the complexity of $\mathcal{H}$, but this is misleading. In fact, the complexity is hidden in the KL-divergence term $\mathrm{KL}(Q|P)$. To see that, consider a case of a finite hypotheses class $\mathcal{H}$ and a uniform prior distribution: $P(h) = \frac{1}{|\mathcal{H}|}, \forall h \in \mathcal{H}$. Then for any Q

$$\mathrm{KL}(Q|P) \leq 2\log|\mathcal{H}|. \tag{2.22}$$

Therefore, the dependence is same as the one in Theorem 2.2.2 for a finite hypothesis class.

2.4 Stochastic processes

As announced in the introduction, we plan to model the generation process of $z_{1:n} = \{z_1, \ldots, z_n\}$ using stochastic processes. While there are different ways to look at and define stochastic processes, in this thesis we adopt a generative view. We think of a stochastic process as a sequence of conditional distributions $D_t = \mathbb{P}\left[\cdot \middle| z_1, \ldots, z_{t-1}\right]$. Then the sample is generated by subsequent sampling from these distributions conditioned on the already sampled points,

i.e. z_1 is sampled from the initial distribution D_1, z_2 from $D_2 = \mathbb{P}[\cdot | z_1]$ and so on. Technically, a sequence of conditional distributions completely defines the process by Ionescu Tulcea Extension Theorem. To simplify the notations, for any time step t, we denote by $\mathbb{E}_t[f(z_{t+1})]$ the expectation of a function f with respect to D_{t+1}, i.e. $\mathbb{E}[f(z_{t+1}) | z_{1:t}]$. This expectation also equivalent to the expectation with respect to the sigma algebra Σ_t that is generated by $z_{1:t}$.

As mentioned above, a sequence of conditional distributions completely specifies the joint distribution of the process. However, there are also other ways we can look at the resulting joint distribution. Sometimes the marginal distributions are of interest, in which case we denote them by M_t, i.e. $M_t(A) = \mathbb{P}[z_t \in A]$ for a set A.

The process is called *stationary* if the vector $(z_{t_1+\tau}, \dots, z_{t_k+\tau})$ has the same distribution for all $\tau \geq 0$ and for any indices $t_1, \dots, t_k$ and any $k \in \mathbb{N}$. If the process is stationary, then the marginals M_t are all the same for all steps t and we denote it by M in this case. We will sometimes mention *ergodic* processes. Their discussion goes beyond the scope of the thesis, we just mention that, informally, a stochastic process is said to be ergodic if its statistical properties can be deduced from a single, sufficiently long, random sample of the process. We refer to [Klenke, 2013] for a formal introduction and definitions.

A very useful class of processes that is helpful for analysis is a class of martingale differences.

Definition 2.4.1. *A real valued process $x_1, \dots, x_t, \dots$ is called a martingale difference (MD) sequence with respect to a filtration $\{\Theta_t\}$ if for all t:*

$$\mathbb{E}[x_t | \Theta_{t-1}] = 0. \tag{2.23}$$

MD sequences seem to be a very special type of processes. However, it appears naturally in the structure of any other stochastic process. In particular, for an arbitrary process $z_{1:\infty}$ and a function $f : \mathcal{Z} \to \mathbb{R}$, a sequence $x_t = f(z_t) - \mathbb{E}_{t-1}[f(z_t)]$ forms a natural martingale difference with respect to Σ_t. Martingale differences exhibit convergence properties similar to i.i.d. sequences in the form of Azuma-Hoeffding inequality. This similarity motivates us to study the uniform deviations similar to $\mathcal{U}_n(\mathcal{H})$. To this end we will be interested in the behavior of the following quantity:

$$\mathcal{V}_n(\mathcal{F}, w) = \sup_{f \in \mathcal{F}} \left| \sum_{t=1}^{n} w_t (f(z_t) - \mathbb{E}_{t-1}[f(z_t)]) \right|, \tag{2.24}$$

where $\mathcal{F} \subseteq \{f : \mathcal{Z} \to [0,1]\}$ and $w \in \mathbb{R}^n$. Similarly to i.i.d. situation, the convergence of $\mathcal{V}_n(\mathcal{F}, w)$ is controlled by a complexity measure of $\mathcal{F}$. However, due to different underlying process, this measures have to be adjusted appropriately and we discuss these adjustments in the next section.

2.5 Sequential complexity measures

The uniform convergence of martingale difference sequences relies on the sequential complexity measures that we introduce in this section. In turn, the sequential complexity measures are based on a notion of $\mathcal{Z}$-valued trees.

Definition 2.5.1. *A $\mathcal{Z}$-valued tree v of depth n is a sequence $v_{1:n}$ of mappings $v_i : \{\pm 1\}^{i-1} \to \mathcal{Z}$. A sequence $\sigma_{1:n} \in \{\pm 1\}^n$ defines a path in a given tree so that $v_t(\sigma_{1:t-1})$ outputs a value of a leaf defined by $\sigma_{1:t-1}$.*

To shorten the notations, $v_t(\sigma_{1:t-1})$ is denoted as $v_t(\sigma)$ Now we can define covering numbers.

Definition 2.5.2. *A set $V = \{v^1, \dots, v^N\}$ of $\mathbb{R}$-valued trees of depth n is a **sequential ε-cover** (with respect to the ℓ_∞-norm) of $\mathcal{F} \subseteq \{f : \mathcal{Z} \to \mathbb{R}\}$ on a $\mathcal{Z}$-valued tree v of depth n if*

$$\forall f \in \mathcal{F}, \forall \sigma \in \{\pm 1\}^n, \exists j \in [N] : \max_{1 \le t \le n} \left| f(v_t(\sigma)) - v_t^j(\sigma) \right| \le \varepsilon. \tag{2.25}$$

*The **sequential ε-covering number** of a function class $\mathcal{F}$ on a given tree v is*

$$\mathcal{S}_\infty(\mathcal{F}, \theta, v) = \min\{|V| : V \text{ is an } \varepsilon\text{-cover w.r.t. } \ell_\infty\text{-norm of } \mathcal{F} \text{ on } v\}. \tag{2.26}$$

*The **maximal sequential ε-covering number** of a function class $\mathcal{F}$ over depth-n trees is*

$$\mathcal{S}_\infty(\mathcal{F}, \theta, n) = \sup_v \mathcal{S}_\infty(\mathcal{F}, \theta, v). \tag{2.27}$$

The sequential covering numbers can be controlled similarly to the standard covering numbers by a corresponding version of a fat-shattering dimension.

Definition 2.5.3. *A $\mathcal{Z}$-valued tree v of depth n is γ-shattered by a function class $\mathcal{F} \subseteq \{f : \mathcal{Z} \to \mathbb{R}\}$ if there exists an $\mathbb{R}$-valued tree s of depth n such that*

$$\forall \sigma \in \{\pm 1\}^n, \exists f \in \mathcal{F} : 1 \le t \le n, \sigma_t(f(z_t(\sigma)) - s_t(\sigma)) \ge \gamma/2. \tag{2.28}$$

At any time point $t = 1, 2, \ldots$:

- **output** hypothesis h_t
- **observe** the state z_t
- **suffer** the loss $\ell(h_t, z_t)$

Figure 2.1: Online learning protocol

The sequential fat-shattering dimension $\mathcal{S}\text{-}fat_\gamma(\mathcal{F})$ *at scale* γ *is the largest* n *such that* $\mathcal{F}$ γ*-shatters a* $\mathcal{Z}$*-valued tree of depth* n*.*

An important result of [Rakhlin *et al.*, 2014] is the following connection between the sequential covering numbers and the sequential fat-shattering dimension.

Lemma 2.5.4 (Corollary 1 of [Rakhlin *et al.*, 2014])**.** *Let* $\mathcal{F} \subseteq \{f : \mathcal{Z} \to [-1, 1]\}$*. For any* $\varepsilon > 0$ *and any* $n \geq 1$*, we have that*

$$\mathcal{S}_\infty(\mathcal{F}, \varepsilon, n) \leq \left(\frac{2en}{\varepsilon}\right)^{\mathcal{S}\text{-}fat_\varepsilon(\mathcal{F})}. \tag{2.29}$$

Having introduced the sequential covering numbers and their properties, we can show now how they can be used for uniform convergence of martingale differences.

Theorem 2.5.5 ([Kuznetsov and Mohri, 2015], Theorem 1)**.** *For a function class* $\mathcal{F} \subseteq \{f : \mathcal{Z} \to [0, 1]\}$ *and any vector* $w \in \mathbb{R}^n$*, we have for any* $\varepsilon > 0$

$$\mathbb{P}\left[\mathcal{V}_n(\mathcal{F}, w) > \varepsilon\right] \leq \mathcal{S}_\infty(\mathcal{F}, \varepsilon/4, n) e^{-\frac{\varepsilon^2}{16\|w\|_2^2}}. \tag{2.30}$$

2.6 Online learning

So far we presented the situations when the data generation process is of stochastic nature. In contrast, the online learning aims to design algorithms that can predict arbitrary sequences that can be chosen even by an adversary. In standard learning theory the learner is presented with the whole dataset at once and has to output a hypothesis for all future data. In online learning, the process goes in steps: at each point the learner outputs a hypothesis for this step and then receives the next data point (see Figure 2.1). As it is impossible to predict each step of an arbitrary adversarial sequence, the goal of learning in this case is to perform as well as

the best possible hypothesis for the sequence in hindsight. Formally, if we denote the sequence of the hypothesis produced by the learner by h_t, the regret of this learner is defined as follows:

$$W_n = \sum_{t=1}^{n} \ell(h_t, z_t) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{n} \ell(h, z_t). \tag{2.31}$$

A desired property of any learner is Hannan consistency.

Definition 2.6.1. *An algorithm is called* Hannan consistent *if the sequence of hypotheses it produces satisfies*

$$\limsup_{n \to \infty} \frac{1}{n} W_n \leq 0. \tag{2.32}$$

In case the learner produces a randomized hypotheses, the above statement should hold almost surely.

Now we consider two examples of the algorithms that we will refer to later in the text.

Follow The Leader. Perhaps the simplest algorithm for online learning is the Follow The Leader (FTL) algorithm that instructs us to choose a hypothesis with the lowest cumulative risk so far, i.e.

$$h_t = \operatorname*{argmin}_{h \in \mathcal{H}} L_t(h) \tag{2.33}$$

for

$$L_t(h) = \sum_{s=1}^{t} \ell(h, z_s). \tag{2.34}$$

In general, it is not a very good strategy, however, there are some settings where it can be quite efficient. An example is given in the following theorem.

Theorem 2.6.2 ([Cesa-Bianchi and Lugosi, 2006], Section 3.2)**.** *Let $\mathcal{H}$ be a Euclidean ball in $\mathbb{R}^d$ of radius* 1 *and let z_t also take values in this ball. If we run the FTL algorithm with $\ell(h, z) = \|h - z\|_2^2$, then its regret satisfies*

$$W_n \leq 8(1 + \log n). \tag{2.35}$$

This theorem ensures Hannan consistency of FTL for this particular setting.

Exponentially Weighted Average. The general form of the Exponentially Weighted Average (EWA) algorithm can be described as follows: let π_t be a distribution over $\mathcal{H}$ that we use to sample h_t, starting from some initial distribution π_1. Having observed z_t, the learner computes the update as

$$\pi_{t+1}(dh) = \frac{e^{-\eta \ell(h,z_t)} \pi_t(dh)}{\int_{\mathcal{H}} e^{-\eta \ell(h',z_t)} \pi_t(dh')} \tag{2.36}$$

with some $\eta > 0$. The following theorem gives a bound on the expected regret of EWA.

Theorem 2.6.3. *Let $h^\star$ be the minimizer of $L_n(h)$. Then for any loss function $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$, the EWA algorithm run with $\eta > 0$ satisfies*

$$\mathbb{E}[W_n] \leq \frac{\eta n}{8} + \frac{-\log \pi_1(h^\star)}{\eta}. \tag{2.37}$$

That gives for $\eta = \frac{1}{\sqrt{n}}$

$$\mathbb{E}[W_n] \leq \sqrt{n}\left(\frac{1}{8} - \log \pi_1(h^\star)\right). \tag{2.38}$$

The expected regret bound can be turned into high-probability statement for a bounded loss functions. Also, the bound of the theorem can be simplified in the case of finite hypotheses classes, see [Cesa-Bianchi and Lugosi, 2006] for more details.

3 Theory of Conditional Risk Minimization

In this chapter we present the problem of conditional risk minimization (CRM). Our goal is to study what properties of stochastic processes are sufficient to achieve conditional learnability, a notion of learnability tailored to stochastic processes. First, we characterize the class of stochastic processes for which the well established algorithms like ERM solve the task. For the rest we propose to make use of additional structural information (if available) and study the guarantees obtainable by two new algorithms: WERM and MACRO. These algorithms differ in the type of learnability they are able to achieve and in computational complexity. While MACRO is more computationally efficient, WERM is able to achieve a stronger notion of learnability. We finish the chapter by studying the terms appearing in the proven generalization bounds for different types of well-known classes of stochastic processes.

3.1 Learning theory for stochastic processes

In this chapter the dataset S is a sequence of observations $\{z_t\}_{t=1}^n$ from a stochastic process taking values in some space $\mathcal{Z}$. There is a long history of research of the extension of PAC learning theory to non-i.i.d. situations. When we move away from the i.i.d. assumption, there is no single distribution D to use in the definition of risk and we have many options with different notions having different properties and uses.

When one is interested in the long-term behaviour of the process, it is sensible to consider the marginal distribution M (for stationary and ergodic processes) and solve the problem of minimizing $R(h, M)$. This makes it possible to reduce the problem again to studying the uniform deviations of the empirical mean similar to $\mathcal{U}_n(\mathcal{H})$, but with an expectation with respect to M, like it done in [Yu, 1994; Meir, 2000]. Typically, this direction does not introduce new algorithms and rather focuses on conditions and situations when the existing algorithms

(like ERM) can be shown to achieve learnability.

In other settings, the short-term behaviour of the process can be of greater interest, as it was argued in [Pestov, 2010; Shalizi and Kontorovitch, 2013]. In such cases, it makes sense to use the conditional distributions of the process at every step, D_t, as they are naturally tuned for the concrete realization of the process at hand, and to minimize $R(h, D_t)$. For some class of processes, like exchangeable ones, it is possible to show learnability by ERM by studying the uniform deviations from the conditional means, [Berti and Rigo, 1997].

However, beyond that there is very few studies that look at the learnability with respect to conditional distributions and a lot of questions are still unanswered. What is the exact description of the class that can be learned by ERM? If we have a class that can not be learned by ERM, what can we do? What does learnability even means if we use the conditional distributions?

In this chapter we give answers to these questions and emphasize an important role of the discrepancies between conditionals of the process for the characterization of the learnability with respect to conditional distributions.

3.2 Conditional risk minimization problem

In this chapter we consider the conditional risk minimization problem where the goal at each step is to find a hypothesis with the minimal conditional risk, i.e. the minimizer of the expected loss on the next point conditioned on the observed data so far. Formally, the risk at step t is

$$R(h, D_t) = \mathbb{E}_{t-1}\left[\ell(h, z_t)\right] \tag{3.1}$$

and at the step n we want to perform the minimization

$$\min_{h \in \mathcal{H}} R(h, D_{n+1}). \tag{3.2}$$

Let consider an example of predicting the next step of discrete valued process with a state space $\mathcal{S}$. For this we define $\mathcal{X} = \emptyset$, $\mathcal{Y} = \mathcal{S}$ and the hypotheses class of constant functions $\mathcal{H} = \{h_s(x) = s, \forall s \in \mathcal{S}\}$. Let us assume that the process is a Markov chain with a state space S and fix a transition function $\pi : S \to \mathcal{Q}_S$, where $\mathcal{Q}_S$ is a space of distributions over

S. Then finding the most probable value on the next step can be stated as a conditional risk minimization problem with 0-1 loss:

$$\min_{h\in\mathcal{H}} \mathbb{E}\left[\ell(h, z_{n+1})|\, z_{1:n}\right] = \min_{s\in S} \mathbb{P}\left[s \neq z_{n+1}|\, z_n\right], \tag{3.3}$$

which is equivalent to $\max_{s\in S} \pi(s|z_n)$.

As in the standard learning theory, the distribution of the process is unknown, hence, we are looking for a method that can perform (3.2) based only on the observed data, i.e. that produces a sequence of hypotheses h_n, where each h_n can be computed from the data observed up to step n and h_n approximates the minimum of (3.2). As the target of learning changes with every step, the standard PAC-learning definition does not makes sense for conditional risk minimization. Therefore, we need to define a new notion of learnability that is suited for this setting.

Definition 3.2.1 (**Conditional Learnability**). *For a fixed loss function ℓ and a hypotheses class $\mathcal{H}$, we call a class of processes $\mathcal{C}$* conditionally learnable in the limit *if there exists an algorithm that, for every process P in $\mathcal{C}$, produces a sequence of hypotheses, h_n, each based on $z_{1:n}$, satisfying*

$$R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) \to 0 \tag{3.4}$$

in probability over the samples drawn from P. We call an algorithm that satisfies this condition a limit learner for the class $\mathcal{C}$.

It is also possible to consider almost sure convergence in the definition of learnability with a minor modifications of our statements.

The above definition concerns itself with the complete convergence. More in line with the PAC-learnability, we will also use the following notion of learnability.

Definition 3.2.2 (**ε-conditional Learnability**). *For a fixed loss function ℓ and a hypotheses class $\mathcal{H}$, we call a class of processes $\mathcal{C}$* ε-conditionally learnable *for $\varepsilon > 0$ if there exists an algorithm that, for every process P in $\mathcal{C}$, produces a sequence of hypotheses, h_n, each based on $z_{1:n}$, satisfying*

$$\mathbb{P}\left[R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) > \varepsilon\right] \to 0. \tag{3.5}$$

An algorithm that satisfies (3.5) *we call an* ε-learner *for the class $\mathcal{C}$.*

Note that this notion of learnability is weaker than conditional learnability. Moreover, a class of processes is conditionally learnable if and only if it is ε-conditionally learnable for all $\varepsilon > 0$.

3.3 Prior work on conditional learnability

A number of classes have been shown to be learnable prior to the work presented in this thesis: i.i.d. [Steinwart, 2005], exchangeable [Berti and Rigo, 1997; Pestov, 2010], conditionally i.i.d. [Berti and Rigo, 2017] and some special cases of stochastic processes [Mohri and Rostamizadeh, 2013]. All these results look at the processes for which the conditional risk can be estimated by a uniformly weighted average over the previous observations and are covered by our results from Section 3.7.

[Kuznetsov and Mohri, 2014] looked at the problem of minimizing $R(h, D_{n+s})$ for different values of s. Their approach relied on the estimation of the conditional risk by an empirical average with the convergence of their bound requiring $s \to \infty$ as n grows. The requirement on s to grow makes the resulting problem quite different from CRM that corresponds to a fixed value $s = 1$ for all n.

Conditional risk minimization was considered in [Kuznetsov and Mohri, 2015] and later extended in [Kuznetsov and Mohri, 2016]. Without taking the conditional learnability into account, they consider the behaviour of the empirical risk minimization algorithm at each fixed time step by taking a non-adaptive estimator of the risk with non-adaptive meaning the same irrespectively of the observed realization. Unfortunately, the proposed methods can not be used to show any form of conditional learnability, because the resulting generalization bounds have a constant term in the upper bound, which prevents it from converging.

3.4 Connection to time series prediction

The CRM framework can be connected to existing theoretical approaches to time series prediction. In particular, we consider two frameworks, which are close enough to conditional risk minimization. In both cases, we show that the conditional risk minimization solves harder problem in a sense that its solutions can be used to solve these particular problems, but it requires more assumptions to be valid.

Prediction by learning. We start with a framework of time series prediction by statistical learning, considered for example in [Alquier *et al.*, 2013; McDonald *et al.*, 2012]. Fixing some point n in time, we consider a hypotheses class $\tilde{\mathcal{H}} \subseteq \{h : \mathcal{Z}^n \to \mathcal{Z}\}$, where each hypotheses h gives us a prediction of the next step by evaluating the whole history. For any loss function $\ell : \mathcal{Z} \times \mathcal{Z} \to [0, 1]$, we consider the following risk minimization problem:

$$\min_{h \in \tilde{\mathcal{H}}} \mathbb{E}\left[\ell(h(z_{1:n}), z_{n+1})\right]. \tag{3.6}$$

To set up the conditional risk minimization, we define a class of constant functions $\mathcal{H} = \{h_{z'}(z) = z', \forall z' \in \mathcal{Z}\}$. Then if the process belongs to a class learnable with $\mathcal{H}$ and ℓ, we can guarantee that there is an algorithm to choose a point z'_n, such that with probability $1 - \delta$

$$\mathbb{E}\left[\ell(z'_n, z_{n+1}) | \, z_{1:n}\right] \leq \inf_{z'} \mathbb{E}\left[\ell(z', z_{n+1}) | \, z_{1:n}\right] + \varepsilon_n(\delta), \tag{3.7}$$

where $\varepsilon_n(\delta)$ is a sequence of errors guaranteed by the algorithm for a given confidence δ so that $\varepsilon_n(\delta) \to 0$. Converting this to the bound on the expectation, we get

$$\mathbb{E}\left[\ell(z'_n, z_{n+1})\right] \leq \mathbb{E}\left[\inf_{z'} \mathbb{E}\left[\ell(z', z_{n+1}) | \, z_{1:n}\right]\right] + \varepsilon_n(\delta) + \delta. \tag{3.8}$$

Notice that

$$\mathbb{E}\left[\inf_{z'} \mathbb{E}\left[\ell(z', z_{n+1}) | \, z_{1:n}\right]\right] \leq \mathbb{E}\left[\inf_{h \in \tilde{\mathcal{H}}} \mathbb{E}\left[\ell(h(z_{1:n}), z_{n+1}) | \, z_{1:n}\right]\right] \tag{3.9}$$

$$\leq \inf_{h \in \tilde{\mathcal{H}}} \mathbb{E}\left[\ell(h(z_{1:n}), z_{n+1})\right]. \tag{3.10}$$

Therefore, if the process is from a learnable class, there is an algorithm that always give good predictions according to this framework as well.

Online prediction. The second setting, which was considered by [Wintenberger, 2017], is very close to the online sequence prediction. In order to reduce the notations and simplify the presentation, we assume that the learner has an access to a (usually finite) hypothesis class $\mathcal{H}$ and at every step t he should choose a distribution π_t over $\mathcal{H}$ in a way that minimizes the following notion of the regret:

$$\sum_{t=1}^{n} \mathbb{E}_{t-1}\left[\ell(\mathbb{E}_{\pi_t} h, z_t)\right] - \min_{h \in \mathcal{H}} \sum_{t=1}^{n} \mathbb{E}_{t-1}\left[\ell(h, z_t)\right]. \tag{3.11}$$

Again, if the process belongs to a learnable class with $\mathcal{H}$ and ℓ, then there is an algorithm, which produce the sequence h_t that satisfies with probability $1 - \delta$

$$\mathbb{E}_{t-1}\left[\ell(h_t, z_t)\right] \leq \min_{h \in \mathcal{H}} \mathbb{E}_{t-1}\left[\ell(h, z_t)\right] + \varepsilon_t(\delta/n) \tag{3.12}$$

for all $1 \leq t \leq n$. Summing up over t, we get

$$\sum_{t=1}^{n} \mathbb{E}_{t-1}\left[\ell(h_t, z_t)\right] \leq \sum_{t=1}^{n} \min_{h \in \mathcal{H}} \mathbb{E}_{t-1}\left[\ell(h, z_t)\right] + \sum_{t=1}^{n} \varepsilon_t(\delta/n) \tag{3.13}$$

$$\leq \min_{h \in \mathcal{H}} \sum_{t=1}^{n} \mathbb{E}_{t-1}\left[\ell(h, z_t)\right] + \sum_{t=1}^{n} \varepsilon_t(\delta/n). \tag{3.14}$$

Thus giving us $\sum_{t=1}^{n} \varepsilon_t(\delta/n)$ bound on the regret with high probability. For nice sequences (like i.i.d.) $\varepsilon_t(\delta/n)$ is of order $\mathcal{O}\left(\sqrt{\frac{\log n}{t}}\right)$, which gives a regret bound of order $\mathcal{O}\left(\sqrt{n \log n}\right)$. On the downside, we can get guarantees only for a class of learnable processes, while the results of [Wintenberger, 2017] hold for any stochastic process. The reason for this is that conditional risk minimization is inherently more difficult problem, since it requires to optimize at every step and not in the cumulative sense.

3.5 Limits to learnability

As discussed in Section 3.3, a number of classes of stochastic processes was shown to be conditionally learnable. On the opposite side, the class of all stationary and ergodic binary processes is not learnable in the particular prediction setting, as we show based on the results of [Gyorfi *et al.*, 1998].

Theorem 3.5.1. *Let $\mathcal{X} = \emptyset$, $\mathcal{Y} = \{0, 1\}$, $\mathcal{H} = [0, 1]$ and $\ell(h, z) = (h - z)^2$. Also, let $\mathcal{C}$ be a class of all stationary ergodic processes taking values in $\mathcal{Z}$. Then for any learning algorithm that produces a sequence of hypotheses h_n, there is a process $P \in \mathcal{C}$ such that*

$$\mathbb{P}\left[\limsup_{n \to \infty} \left(R(h_n, D_{n+1}) - \inf_{h \in \mathcal{H}} R(h, D_{n+1})\right) > \frac{1}{16}\right] \geq \frac{1}{8}. \tag{3.15}$$

3.6 Discrepancies

Having discussed the limits to CRM and its relation to other problems, we start to work towards solving CRM by introducing a key notion of *pairwise discrepancies*, a measure of distance between conditional distributions at different time steps.

Definition 3.6.1 (Pairwise discrepancy). *For a sample $z_1, z_2, \ldots$ from a fixed stochastic process, the pairwise discrepancy between time points i and j is*

$$d_{i,j} = \sup_{h \in \mathcal{H}} \left|R(h, D_i) - R(h, D_j)\right|. \tag{3.16}$$

As a distance measure between two distributions, it is also known by the name of an integral probability metric and is studied, for example, in [Zolotarev, 1983; Müller, 1997]. For learning problems it is a very suitable measure of distance, because it is adapted to the underlying hypothesis set, making it a popular choice in the domain adaptation literature [Kifer *et al.*, 2004; Ben-David *et al.*, 2007; Mansour *et al.*, 2009; Ben-David *et al.*, 2010; Mohri and Medina, 2012].

From this point we distinguish between two situations: when the discrepancies exhibit a special form of convergence and when they do not. In the former case, we show that using the ERM algorithm is sufficient. For the latter, we present two approaches, one based on weighted ERM and another based on a new MACRO algorithm, which we introduce later, allowing to control the discrepancies.

3.7 Convergent case

We start describing the situation when the existing algorithms are sufficient to achieve conditional learnability. The intuition behind this case is that if a sequence (of numbers, random variables, etc.) is convergent then the average of elements in the sequence also converges to the same limit. However, in our situation we do not have a single sequence, but rather a double array of discrepancies, $d_{t,n}$. We use the following definition of convergence, which is a modification of standard convergence in probability to our special case of double array.

Definition 3.7.1. *A double array of random variables $d_{t,n}$ with $n \in \mathbb{N}$ and $0 \leq t < n$ is called* convergent *if*

$$\begin{gathered} \forall \varepsilon > 0, \forall \delta > 0, \exists n_0, t_0 : 0 \leq t_0 < n_0, \forall n \geq n_0, \\ \forall t_0 \leq t < n : \mathbb{P}\left[d_{t,n} > \varepsilon\right] \leq \delta. \end{gathered} \tag{3.17}$$

With this definition in hand, we can state the following theorem.

Theorem 3.7.2. *For any hypotheses class $\mathcal{H}$ such that $\mathcal{L}(\mathcal{H})$ has a finite sequential fat-shattering dimension, if every process in the class $\mathcal{C}$ has convergent discrepancies, then the ERM algorithm is a limit learner.*

A trivial example of the convergent situation is an i.i.d. process, because all the discrepancies are zero. For a more general example, we consider a class of *$\mathcal{F}$-uniformly convergent*

martingales. This class consists of processes that form martingales for every function $f \in \mathcal{F}$ applied to its values, that is $\mathbb{E}_s\left[\mathbb{E}_t\left[f(z_{t+1})\right]\right] = \mathbb{E}_s\left[f(z_{s+1})\right]$ for $s < t$. By standard results in the theory of martingales, e.g. [Williams, 1991], for every f there is a limit random variable r_f, such that $\mathbb{E}_t f \to r_f$ in probability and $\mathbb{E}_t f = \mathbb{E}_t r_f$. This motivates the following definition.

Definition 3.7.3. *For a functions class $\mathcal{F} \subseteq \{f : \mathcal{Z} \to [0,1]\}$, a martingale process is called a $\mathcal{F}$-*uniformly convergent martingale *if*

$$\sup_{f \in \mathcal{F}} |\mathbb{E}_t f - r_f| \to 0 \tag{3.18}$$

in probability.

For such classes we have the following results.

Lemma 3.7.4. *A $\mathcal{L}(\mathcal{H})$-uniformly convergent martingale has convergent discrepancies.*

Corollary 3.7.5. *If a class $\mathcal{C}$ consists of $\mathcal{L}(\mathcal{H})$-uniformly convergent martingales, then it is conditionally learnable by the ERM algorithm.*

As shown in [Berti *et al.*, 2002], a prominent example of uniformly convergent martingales is a class of exchangeable sequences that is widely used in the statistical literature.

Definition 3.7.6. *A process is called exchangeable if $(z_{j_1}, \ldots, z_{j_n})$ and $(z_1, \ldots, z_n)$ have the same distribution for every n and every n-tuple of distinct indices $j_1, \ldots, j_n$.*

Exchangeability means that the sampled data has the same distribution irrespectively of the order of variables. In addition to i.i.d., this assumption also covers an important case of complete dependence, when one observes copies of the same random variable. From this perspective, Corollary 3.7.5 can be seen as a generalization of the results proven by [Berti *et al.*, 2002] for exchangeable sequences and, even further, by [Berti and Rigo, 2017] for conditionally identically distributed sequences.

Another example of a class with convergent discrepancies is a class of processes used in [Mohri and Rostamizadeh, 2013]. Their assumption (equation 6) states for any hypothesis h that depends only on $z_{1:t}$ and any $n \geq s > t$ we have

$$\mathbb{E}\left[\ell(h, z_{n+1}) \middle| z_{1:s}\right] = \mathbb{E}\left[\ell(h, z_{n+1}) \middle| z_{1:t}\right]. \tag{3.19}$$

In particular, that means that for any fixed h (not depending on the sample)

$$\mathbb{E}\left[\ell(h, z_{n+1}) \middle| z_{1:n}\right] = \mathbb{E}\left[\ell(h, z_{n+1})\right]. \tag{3.20}$$

If the marginal distributions at each step are the same, as assumed in [Mohri and Rostamizadeh, 2013], then (3.20) yields that all $d_{t,n}$ are zero, so they are convergent.

3.8 Non-convergent case

In the case the discrepancies do not converge in the sense of Definition 3.7.1, we need an additional information about their behaviour. In particular, we show that all that is needed is a computable upper bound on the discrepancies. This is summarized in the following definitions of a D-bound and an M-bound. Later we will describe the WERM algorithm that requires an M-bound to achieve conditional learnability and MACRO algorithm that requires only a D-bound, but achieves a weaker ε-conditional learnability.

Definition 3.8.1 (D-bound). *A double array of random variables $M_{i,j}$, with $i, j \in \mathbb{N}$ is called a* D-bound *if*

1. *$M_{i,j}$ is measurable with respect to $\Sigma_{\max(i,j)-1}$*
2. *$d_{i,j} \leq M_{i,j}$ for all $t, j \in \mathbb{N}$.*

Trivially, discrepancies themselves form a non-computable D-bound, however, a general D-bound is a necessary abstraction and differs from the discrepancies by fact that we should be able to compute it from the sample. For example, in case of discrete-state Markov chains, a D-bound $M_{i,j} = \mathbb{I}[z_{i-1} \neq z_{j-1}]$ fulfills the necessary conditions. Another, computable but nonetheless trivial example is a constant sequence $M_{i,j} = 1$. However, this D-bound is not very useful as we will later see from the additional conditions on the behavior of D-bounds that allow for conditional learnability.

To prove conditional learnability for the weighted empirical risk minimization the mere existence of a D-bound is not enough. In addition, we require it to satisfy a special measurability condition that summarized in the following definition.

Definition 3.8.2 (M-bound). *A* D-bound *$M_{i,j}$ with $i, j \in \mathbb{N}$ is called a* M-bound *if there exists a sequence of functions $\Psi_i(r)$ for $i, r \in \mathbb{N}$ and a sequence of random variables J_r taking values in $\mathbb{N}$ such that*

1. *$M_{i,j} = \Psi_i(J_j)$ for all j and $1 \leq i \leq j$,*

Input: M-bound $M_{i,j}$, smoothing functions g_t

Initialization: $S = \emptyset$

At any time point $t = 1, 2, \ldots$:

- **compute** the weights over current training set S: $w_s = \frac{g_t(M_{s,t})}{\sum_{j=1}^{t-1} g_t(M_{j,t})}$
- **output** the current hypothesis: $h_t \leftarrow \operatorname{argmin}_h \sum_{s=1}^{t-1} w_s \ell(h, z_s)$
- **observe** the next value of the process, z_t
- **update** the current training set: $S \leftarrow S \cup \{z_t\}$

Figure 3.1: Weighted ERM algorithm

2. *$\Psi_i(r)$ is measurable with respect to Σ_{i-1} for all fixed $r \in \mathbb{N}$.*

In the example of Markov chain prediction, let $s_1, s_2, \ldots, s_{|S|}$ be any enumeration of the state space S. Then we can define $\Psi_i(r) = \mathbb{I}[z_{i-1} \neq s_r]$ for $1 \leq r \leq |S|$ and set $J_j = k$ for k that satisfies $s_k = z_{j-1}$.

3.8.1 Weighted ERM

As in standard learning theory, we first focus on the empirical risk minimization principle, which governs us to construct an estimator of the risk based on the data and use the minimizer of the estimator as an output hypothesis. The main question is how to construct this estimator. In Section 2.1, we used $\frac{1}{n}\sum_{t=1}^{n} \ell(h, z_t)$ as an estimator in case of i.i.d. data, however, for general processes, this quantity is not a good choice as it does not converge to the conditional risk, except for some special cases, which are covered by Theorem 3.7.2. For other situations we consider linear estimators of the form $\sum_{t=1}^{n} w_t \ell(h, z_t)$ with $w_t \geq 0$ and $\sum_{t=1}^{n} w_t = 1$ (we omit the dependence of w_t's on n, but emphasize that at each step the weights can be different, because we estimate different quantities) and output the minimizer: $h_n = \operatorname{argmin}_h \sum_{t=1}^{n} w_t \ell(h, z_t)$. Thus we change to the problem to finding "good" weights w based on the observed sample that make empirical risk minimization a limit learner. This makes the weights a function of the observed data, so they must be treated as random variables.

In the case of Markov chain, the estimator for state s has the form $\sum_{t=1}^{n} w_t \mathbb{I}[s \neq z_t]$. Clearly, there is no fixed choice of weights that would approximate the conditional risk well for every realization of the process. The empirical average with uniform weights, $w_t \equiv \frac{1}{n}$, for example, converges to the risk with respect to the stationary distribution of the chain, not the

conditional one. Instead, we should choose w_t to be large, if the (conditional) distribution of z_t is similar to the distribution of z_{n+1}, i.e. $\pi(\cdot|z_{t-1}) \approx \pi(\cdot|z_n)$. Otherwise, w_t should be small. The same intuition holds for general processes, as we will show later in Section 3.8.

To study the properties of the weighted ERM, we use the fact similar to Lemma 2.1.2 that

$$R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) \leq 2 \sup_{h\in\mathcal{H}} \left| \sum_{t=1}^{n} w_t \ell(h, z_t) - R(h, D_{n+1}) \right| \tag{3.21}$$

and, henceforward, focus on the right hand side, i.e. uniform deviations of the estimator. The starting point to understanding its behaviour is the following decomposition, proposed in [Kuznetsov and Mohri, 2015]:

$$\begin{aligned} \sup_{h\in\mathcal{H}} \left| \sum_{t=1}^{n} w_t \ell(h, z_t) - R(h, D_{n+1}) \right| \leq & \sup_{h\in\mathcal{H}} \left| \sum_{t=1}^{n} w_t \left(\ell(h, z_t) - R(h, D_t) \right) \right| \\ & + \sup_{h\in\mathcal{H}} \left| \sum_{t=1}^{n} w_t R(h, D_t) - R(h, D_{n+1}) \right|. \end{aligned} \tag{3.22}$$

The first term is $\mathcal{V}_n(\mathcal{L}(\mathcal{H}), w)$ introduced in Section 2.4. For the second term, we go further and observe that for weights that satisfy $w_t \geq 0$ and $\sum_{t=1}^{n} w_t = 1$, we can further upper bound it as

$$\sup_{h\in\mathcal{H}} \left| \sum_{t=1}^{n} w_t R(h, D_{t-1}) - R(h, D_{n+1}) \right| \leq \sum_{t=1}^{n} w_t d_{t,n+1}. \tag{3.23}$$

That gives us

$$\sup_{h\in\mathcal{H}} \left| \sum_{t=1}^{n} w_t \ell(h, z_t) - R(h, D_{n+1}) \right| \leq \mathcal{V}_n(\mathcal{L}(\mathcal{H}), w) + \sum_{t=1}^{n} w_t d_{t,n+1}. \tag{3.24}$$

For fixed data-independent weights, $\mathcal{V}_n(\mathcal{L}(\mathcal{H}), w)$ represents the stochastic part of the problem and can be shown to converge under very general conditions using the machinery of Section 2.5. An important fact is that the rate of convergence is determined by $\|w\|_2$. For example, uniform weights $w_t = \frac{1}{n}$ yield an optimal, $\mathcal{O}(\frac{1}{\sqrt{n}})$, rate. On the contrary, if only one sample is present, i.e. $w_t = 1$ for some t and $w_t = 0$ for all others, then there is no convergence as $\|w\|_2$ does not decrease as a function of n.

The second term in 3.22 measures how well D_{n+1} is approximated by the previous distributions $D_1, \ldots, D_n$. Opposite to $\mathcal{V}_n(\mathcal{L}(\mathcal{H}), w)$, it is minimized by setting $w_t = 1$ for index t that has the smallest discrepancy. This creates a natural trade-off between two desirable properties of the weights: they should offer good statistical power (have small $\|w\|_2$), while achieving a high approximation quality (small average discrepancy).

As it can be clearly observed in the example of a Markov chain, in the non-convergent situation there is no single choice of weights that can be applied irrespectively of the process

and the sample. Rather one should adjust the weights to the data at hand. This is where the notion of D-bounds comes into play. Observe that for a given D-bound, we can further upper bound (3.23):

$$\sum_{t=1}^{n} w_t d_{t,n+1} \le \sum_{t=1}^{n} w_t M_{t,n+1}. \tag{3.25}$$

This expression would be minimized by setting $w_t = 1$ for the index t with minimal value of $M_{t,n+1}$, while keeping the other weights at 0. However, as discussed above, such a choice is disastrous for the stochastic part of (3.24). Therefore, we suggest to use a version of soft-min with some smoothing function $g_n : \mathbb{R} \to [0, 1]$ that also could be different at each step.

$$w_t = \frac{g_n(M_{t,n+1})}{\sum_{j=1}^{n} g_n(M_{j,n+1})} \tag{3.26}$$

for $1 \le t \le n$. A popular smoothing function is $g_n(x) = e^{-\gamma_n x}$ for some $\gamma_n > 0$. In the example of a Markov chain, the simpler $g_n(x) = \mathbb{I}[x = 0] = \lim_{\gamma \to 0} e^{-\gamma x}$ is sufficient as can be seen in the next example. The final description of the proposed WERM algorithm is given in Figure 3.1.

We observe that for the Markov chain the pairwise discrepancies can be written using the transition function, $d_{i,j} = \max_{s \in S} |\pi(s|z_{i-1}) - \pi(s|z_{j-1})|$. Then it is immediate that $d_{t,n+1} \le \mathbb{I}[z_{t-1} \ne z_n]$, hence, setting $w_t = \frac{\mathbb{I}[z_{t-1}=z_n]}{\sum_{j=2}^{n} \mathbb{I}[z_{j-1}=z_n]}$ seems like a good choice: it uses only samples from the distribution we are trying to predict for, and it distributes the mass evenly among those. Consequently, the discrepancy term in (3.24) is zero and $\|w\|_2$ is minimal.

Due to the stochastic nature of the process, it may not be possible to have a good bound for each possible realization. Imagine a situation in a Markov chain when at step n we observe the state z_n for the first time. Then we have no information in the sample about the distribution of the next step. Nevertheless, if such realizations are rare, the process can still be learnable. We formalize this idea in an exceptional set of realizations, which we are going to ignore and require that they appear with small probability.

Definition 3.8.3 (Exceptional set). *Let $M_{i,j}$ be an M-bound with representation $M_{i,j} = \Psi_i(J_j)$. For a fixed n, for any $k \ge 1$ and $1 \le m \le n$, set*

$$\mathcal{E}_{k,m} = \Big\{ J_{n+1} \le k \ \wedge \ \sum_{t=1}^{n} g_n(M_{t,n+1}) \ge m \Big\}. \tag{3.27}$$

We define $\mathcal{E}^c_{k,m}$, the complement of $\mathcal{E}_{k,m}$, as an exceptional set *of the realizations.*

Note that this set is also different at each step, but we omit the index n to avoid cluttering of the notations. The condition in (3.27) mainly requires to have a lower bound on the

denominator of w_t's, thereby avoiding the situation observed in a Markov chain example. We will discuss the behaviour of $\mathbb{P}\left[\mathcal{E}_{k,m}^c\right]$ for discrete state Markov chains (and other processes) in Section 3.8.4.

The following theorem provides guarantees on the performance of empirical risk minimization with our proposed choice of weights.

Theorem 3.8.4. *For any M-bound $M_{t,j}$, let h_n be the sequence of hypotheses produced by the corresponding weighted ERM algorithm. Then for any fixed n, for any $k, m \geq 1$, $\alpha \in [0,1]$ and $\beta \in [0, \alpha/4]$ the following inequality holds*

$$\mathbb{P}\left[R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) - 2\Lambda_n \geq 2\alpha\right] \leq \frac{2k\mathcal{S}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{(\alpha - 4\beta)^2} e^{-\frac{1}{2}m(\alpha-4\beta)^2} + \mathbb{P}\left[\mathcal{E}_{k,m}^c\right], \tag{3.28}$$

where $\Lambda_n = \sum_{t=1}^n w_t M_{t,n+1}$.

It may be instructive to look at the different form of Theorem 3.8.4. For any $k, m \geq 1$, any $\delta > \mathbb{P}[\mathcal{E}_{k,m}^c]$ and $\beta > 0$, with probability $1 - \delta$ the following inequality holds:

$$R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) \leq 2\Lambda_n + 8\beta + 2\sqrt{\frac{2\log\frac{4m}{\delta} + 2\log 2k\mathcal{S}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{m}}. \tag{3.29}$$

From this form we can read off conditions under which the ERM algorithm becomes a limit learner. As this means that the right hand side converges to 0, in particular we need that Λ_n vanishes in the limit.

Corollary 3.8.5. *Assume that for every process P in the class $\mathcal{C}$ there exists a sequence of M-bounds and smoothing functions satisfying $\Lambda_n \to 0$. Let $\mathcal{L}(\mathcal{H})$ have a finite sequential fat-shattering dimension. If there exist k_n, m_n satisfying $\frac{m_n}{\log n} \to \infty$ and $\mathbb{P}[E_{k_n,m_n}^c] \to 0$, then $\mathcal{C}$ is conditionally learnable in the limit by the WERM algorithm based on the given M-bounds and smoothing functions.*

Note that the algorithm does not require knowledge of the parameters k and m, merely the existence of good values.

The above results show that the quality of M-bounds is of crucial importance for establishing conditional learnability. In Section 3.8.3 we highlight constructions of M-bounds based on prior knowledge (or assumptions) on the processes. Moreover, each process of a class should not produce unfavorable sequences very often. We will look into this property of processes in more details in Section 3.8.4.

Input: D-bound $M_{i,j}$

Initialization: $T \leftarrow \emptyset$, $N \leftarrow 0$

At any time point $t = 1, 2, \ldots$:

- **choose** the active hypothesis from the closest ε-close subroutine or a newly started one
  - identify all ε-close subroutines: $J = \{1 \leq j \leq N : M_{t,\tau_j} \leq \varepsilon\}$.
  - if $J = \emptyset$: create a new subroutine, S_{N+1}, and set $J = \{N+1\}; \tau_{N+1} \leftarrow t; N \leftarrow N+1$.
  - set the active hypothesis: $h_a \leftarrow \text{output}(S_j)$ for $j = \text{argmin}_{j \in J} M_{t,\tau_j}$.
- **output** the currently active hypothesis, $h_t \leftarrow h_a$
- **observe** the next value of the process, z_t
- **update** all ε-close subroutines: $S_j \leftarrow \text{update}(S_j, z_t)$ for all $j \in J$

Figure 3.2: MACRO algorithm

3.8.2 MACRO

The weighted ERM algorithm of the previous section is able to achieve learnability for a wide class of processes with non-convergent discrepancies. Now, if one wants to apply the algorithm in practice, then we are faced with a problem: a running time of the algorithm at each step is at least quadratic in the number of samples. This makes it infeasible to apply it to datasets even of moderate size. The main goal of the this section is to present an algorithm that has better computational complexity, while still retaining learnability guarantees in the form of ε-conditional learnability.

We present a Meta Learning Algorithm for Conditional Risk Minimization (MACRO) that is build on the similar idea to WERM: if two conditional distributions are very similar, we can use the same hypothesis for both of them. To find these hypotheses, the meta-algorithm maintains a list of learning subroutines, where each of them is run independently and updated using a selected subset of the observed data points. Over the course of the algorithms, the meta-algorithm always maintains an *active hypothesis* that can immediately be applied when a new observation arrives. After each observation, one or more of the existing subroutines are updated, and a new subroutine can be added to the list, if necessary. The meta-algorithm then constructs a new active hypothesis from the ones produced by the currently running subroutines, to be prepared for the next step of the process. The schema of the algorithm is given in Figure 3.2.

Before we proceed to the theoretical properties of the meta-algorithm, we fix further no-

tations for its components. At any time step n, we denote by N_n the number of started subroutines (i.e. the current value of N). The time steps in which the j-th subroutine is updated up to step n form a set $C_{j,n} = \{t_{j,1}, \ldots, t_{j,s_{j,n}}\}$ of size $s_{j,n}$. By $h_{j,i}$ we denote the output of the j-th subroutine after having been updated i-times. By $I_n \in [N_n]$ we denote the index of the subroutine that MACRO outputs in step n, i.e. $h_n = h_{I_n, s_{I_n,n}}$.

Computational considerations. The overall computational complexity of the weighted ERM principle is proportional to n^2 for a dataset of size n, while MACRO is able to reduce it to nN_n with a potential for further reduction. The following lemma considers the quantitative behavior of N_n.

Lemma 3.8.6. *For any D-bound $M_{i,j}$, let $\mathcal{N}(M, n, \varepsilon)$ be an ε-covering number of $\{D_1, \ldots, D_n\}$ with respect to $M_{i,j}$'s. Then for any $n = 1, 2, \ldots$, it holds that*

$$\mathcal{N}(M, n, \varepsilon) \leq N_n \leq \mathcal{N}(M, n, \varepsilon/2). \tag{3.30}$$

Observe that $\mathcal{N}(M, n, \varepsilon)$ is always lower-bounded by $\mathcal{N}(d, n, \varepsilon)$, making it a natural limit on how many separate subroutines are required to learn a particular sequence.

Exceptional sets. As discussed in Section 3.8 (and resembling the "probably" aspect of PAC learning), learnability guarantees for stochastic processes may not hold for every possible realization of the process. Henceforth, we follow the same strategy for MACRO and introduce a set of exceptional realizations. However, the definition differs from (3.8.3), as it is adapted to the working mechanisms of the meta-algorithm.

Definition 3.8.7 (Exceptional set for MACRO)**.** *For a fixed n, for any $k \geq 1$ and $1 \leq m \leq n$, set*

$$\mathcal{X}_{k,m} = \{|\text{supp}(I_n)| \leq k \wedge \min_{j \in \text{supp}(I_n)} s_{j,n} \geq m\}, \tag{3.31}$$

where $\text{supp}(I_n)$ *denotes the support of I_n. Then $\mathcal{X}_{k,m}^c$, the complement of $\mathcal{X}_{k,m}$, is an exceptional set of realizations.*

In words, the favorable realizations are the ones that do not force the algorithm to use too many subroutines (at most k) and, at same time, all used subroutines are updated often enough (at least m times). The intuition behind this is that a subroutine will be slow in converging to an optimal predictor if it is updated very rarely. However, the overall performance of the

meta-algorithm can suffer only if rarely updated subroutines are nevertheless used from time to time.

Subroutines. MACRO, as a meta algorithm, relies on the subroutines to perform the actual learning of hypotheses. In the following, we will go through several option for subroutines and discuss the resulting theoretical guarantees.

Empirical risk minimization. We start with the simplest choice of a subroutine: an ERM algorithm. Writing it using the introduced notations, the j-th ERM subroutine outputs

$$h_{j,n} = \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{s_{j,n}} \sum_{t \in C_{j,n}} \ell(h, z_t). \tag{3.32}$$

Consequently, MACRO's output is $h_n = h_{I_n,n}$ for which we can prove the following theorem.

Theorem 3.8.8. *For any D-bound $M_{i,j}$, if MACRO is run with ERM as a subroutine, then we have for any $k, m \geq 1, \alpha \in [0,1]$ and $\beta \in [0, \alpha/4]$*

$$\mathbb{P}\left[R(h_n, D_{n+1}) - \inf_{h \in \mathcal{H}} R(h, D_{n+1}) > 2\alpha + 4\varepsilon\right] \leq \frac{2k\mathcal{S}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{(\alpha - 4\beta)^2} e^{-\frac{1}{2}m(\alpha-4\beta)^2} + \mathbb{P}\left[\mathcal{X}^c_{k,m}\right].$$

From this theorem we can read off the conditions for learnability of the meta-algorithm. First, we consider hypotheses class $\mathcal{H}$ so that $\mathcal{L}(\mathcal{H})$ has a finite sequential fat-shattering dimension. If there exist sequences k_n, m_n, satisfying $\frac{m_n}{\log n} \to \infty$ and $\mathbb{P}[\mathcal{X}^c_{k_n,m_n}] \to 0$, then the meta-algorithm with ERM as a subroutine is an ε-learner (up to a constant). The condition on the rate of growth of m_n comes from the fact that it needs to compensate for the growth of covering numbers, which is a polynomial of n (see Lemma 2.5.4). The existence of such sequences k_n and m_n depends purely on the properties of the process (or class of processes) that the data is sampled from. Importantly, neither k_n nor m_n are needed to be known by MACRO as it automatically adapts to unfavorable conditions and exploits the favorable ones. Note that the computation of h_n can be seen as a minimization of non-uniformly weighted average over the observed data as done in WERM.

Online learning. ERM as a subroutine is interesting from a theoretical perspective, but it defeats the main purpose of the meta-algorithm, namely that not all data of the process has to be stored. Instead, one would prefer to rely on a subroutine that can be trained incrementally, i.e. one sample at a time, as it is typical in *online learning*.

In the following, by an *online subroutine* we understand any algorithm that is designed to control the *regret* over each particular realization, like we presented in Section 2.6. The regret of the j-th subroutine at the step n is defined as

$$W_{j,n} = \sum_{i=1}^{s_{j,n}} \ell(h_{j,i-1}, z_{t_{j,i}}) - \inf_{h\in\mathcal{H}} \sum_{i=1}^{s_{j,n}} \ell(h, z_{t_{j,i}}). \tag{3.33}$$

The choice of a particular subroutine depends on the loss function and the hypotheses class. To abstract from concrete bounds and subroutines, we prove a theorem that bounds the performance of the meta-algorithm in terms of the regrets of the subroutines. Thereby, we obtain that any regret minimizing algorithm will be efficient as a subroutine for MACRO as well.

As our goal is not to minimize regret, but the conditional risk, we perform an *online-to-batch conversion* to choose the output hypothesis of each subroutine. In this work we consider two of the many existing online-to-batch conversion methods, one specifically for the convex losses and the other one for the general case.

Convex losses. For a convex loss function, we set the output of a subroutine to the average over the hypotheses it produced so far. In this case, MACRO's output is $h_n = \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} h_{I_n,i}$. and we can prove the following theorem.

Theorem 3.8.9. *For a convex loss ℓ, if the subroutines of MACRO use an averaging for online-to-batch conversion, we have for any $\alpha \in [0,1]$ and $\beta \in [0, \alpha/8]$*

$$\begin{aligned} &\mathbb{P}\left[R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) > \alpha + W_{I_n,n}/s_{I_n,n} + 4\varepsilon\right] \\ &\leq \frac{4k\mathcal{S}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{(\alpha/2 - 4\beta)^2} e^{-\frac{1}{2}m(\alpha/2-4\beta)^2} + \mathbb{P}\left[\mathcal{X}_{k,m}^c\right]. \end{aligned} \tag{3.34}$$

For Hannan-consistent online algorithms, $W_{I_n,n}/s_{I_n,n}$ vanishes as $s_{I_n,n}$ grows. Hence, the same conditions as the ones given after Theorem 3.8.8 ensures that MACRO is an ε-learner in this case.

Non-convex losses. For non-convex losses, a simple averaging for online-to-batch conversion does not work, so we need to perform a more elaborate procedure. We use a modification of the method introduced in [Cesa-Bianchi *et al.*, 2004]. As the original method was designed to work for i.i.d. data, we need to extend it to stochastic processes. The general idea is to

assign a score to each of $h_{j,i}$ and choose the hypothesis with the lowest one. For a given confidence $\delta > 0$, the score of $h_{j,i}$ is computed as

$$u_n(j,i) = \frac{1}{s_{j,n} - i} \sum_{k=i+1}^{s_{j,n}} \ell(h_{j,i}, z_{t_{j,k}}) + c_{j,\delta}(s_{j,n} - i), \tag{3.35}$$

where

$$c_{j,\delta}(t) = \sqrt{\frac{1}{2(t+1)} \log \frac{s_{j,n}^3(s_{j,n}+1)}{\delta}} \tag{3.36}$$

reflects the uncertainty of this value that is caused by different subroutines having been trained on different amounts of training data. Setting $J_n = \operatorname{argmin}_{1 \le i \le s_{I_n,n}} u_n(I_n, i)$, MACRO's output is $h_n = h_{I_n,J_n}$. and we are able to prove the following theorem.

Theorem 3.8.10. *For any $\delta \in [0,1]$ and $\beta > 0$, denote*

$$U_\delta(j,\beta) = 2\sqrt{\frac{1}{s_{j,n}} \log \frac{s_{j,n}^3(s_{j,n}+1)}{\delta}} + \sqrt{\frac{1}{s_{j,n}} \log \frac{s_{j,n}^2}{\delta}} + \sqrt{\frac{1}{s_{j,n}} \log \frac{s_{j,n}^2 \mathcal{S}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{\delta}} + 4\beta. \tag{3.37}$$

If the subroutines of MACRO use the score-based online-to-batch conversion with confidence δ, it holds that

$$\mathbb{P}\left[R(h_n, D_{n+1}) - \inf_{h \in \mathcal{H}} R(h, D_{n+1}) > 2\varepsilon + W_{I_n,n}/s_{I_n,n} + U_\delta(I_n, \beta)\right] \le k\delta/m + \mathbb{P}\left[\mathcal{X}_{k,m}^c\right]. \tag{3.38}$$

The same conditions as before will ensure ε-learnability. Note that to perform this form of online-to-batch conversion neither k nor m need to be known.

3.8.3 Controlling pairwise discrepancies

In this section we give two examples of how D-bounds and M-bounds can be constructed. We already discussed an example for the problem of predicting the next state of a discrete state Markov processes in Section 3.8. The construction can be extended to more general situations, when the discrepancy between two time steps can be related to the similarity of their histories, namely for all $h \in \mathcal{H}$,

$$\left|\mathbb{E}\left[\ell(h, z_{i+1}) \middle| z_{1:i}\right] - \mathbb{E}\left[\ell(h, z_{j+1}) \middle| z_{1:j}\right]\right| \le \lambda(z_{1:i}, z_{1:j}), \tag{3.39}$$

where $\lambda : \mathcal{Z}^i \times \mathcal{Z}^j \to \mathbb{R}_+$ is a form of distance measure. This property can be derived, e.g., from the continuity of the conditional distribution with respect to the history, which is a

common assumption in the literature on nonparametric estimation, e.g. [Györfi *et al.*, 1989; Hansen, 2008; Linton and Sancetta, 2009]. This makes $M_{i,j} = \lambda(z_{1:i-1}, z_{1:j-1})$ a natural D-bound.

For an example of M-bound, let us assume that λ takes only the q most recent values into account and is a metric, so that we can rewrite inequality (3.39) as

$$\sup_{h\in\mathcal{H}} |R(h, D_i) - R(h, D_j)| \leq \lambda(z_{i-q:i-1}, z_{j-q:j-1}). \tag{3.40}$$

Now, for fixed $\varepsilon > 0$ let $\mathcal{M}$ be a ε-cover of $\mathcal{Z}^q$ with respect to λ. For any $\bar{z}$, let $c(\bar{z})$ denote the closest element of $\mathcal{M}$. Then we have

$$\sup_{h\in\mathcal{H}} \left|R(h, D_i) - R_{D_j}(h)\right| \leq \varepsilon + \lambda(z_{i-q:i-1}, c(z_{j-q:j-1})). \tag{3.41}$$

Now, let $\bar{m}_1, \bar{m}_2, \ldots$ be an enumeration of $\mathcal{M}$, and define $\Psi_i(r) = \varepsilon + \lambda(z_{i-q:i-1}, \bar{m}_r)$. Then we obtain an M-bound by setting $J_j = k$ for k that satisfies $m_k = c(z_{j-q:j})$. In a similar way, we can obtain D-bounds and M-bounds for related statistical settings, as the assumption of continuity is a fundamental ingredient for many theoretical results in nonparametric statistics.

Another example of an M-bound can be given in the scenario of *rarely changing distributions*. In this case we observe independent samples, however, the distribution from which these are sampled may occasionally change [Tartakovsky *et al.*, 2014]. Formally, for a fixed sequence of distribution $Q_1, \ldots, Q_k$ and change points, $1 = c_1 < \cdots < c_{k+1} = n + 1$, the samples $z_{c_i:c_{i+1}-1}$ are drawn independently from the distribution Q_i, for $i = 1, \ldots, k$, i.e. $D_{c_i}, \ldots, D_{c_{i+1}-1}$ are equal to Q_i. A simple strategy for this task is to perform change point detection, for example [Tartakovsky *et al.*, 2014; Khaleghi and Ryabko, 2016; Kifer *et al.*, 2004], and then distribute the weight uniformly over the samples since the last change, i.e. $w_t = 0$ for $t = 1, \ldots, c_k - 1$ and $w_t = \frac{1}{n-c_k+1}$ for $t = c_k, \ldots, n$. A more elaborate scheme in this situation would be to estimate the discrepancies between segments and use the estimates in the place of real discrepancies. A similar approach was studied in the active learning scenario by [Pentina and Lampert, 2017].

3.8.4 Controlling the exceptional set for WERM

For classes of processes with non-convergent discrepancies, learnability of WERM requires not just the existence of an M-bound, but also control of the exceptional set (Corollary 3.8.5). In this section, we connect this property to some well-known properties of stochastic processes.

To isolate the properties of the process from the assumptions required to get an M-bound, we will analyze the ERM algorithm with a universal (though unfortunately incomputable) M-bound: we assume that we have access to the individual discrepancies $d_{i,j}$ and define the bound in the following way. Fix a time step n, some $\varepsilon_n > 0$ and let

$$J_{n+1} = \inf \{t \geq 1 : d_{t,n+1} \leq \varepsilon_n\}. \tag{3.42}$$

Then, for $t \geq J_{n+1}$

$$d_{t,n+1} \leq d_{J_{n+1},n} + d_{t,J_{n+1}} \leq \varepsilon_n + d_{t,J_{n+1}}, \tag{3.43}$$

by a triangle inequality. This gives us a M-bound as we can set $\Psi_t(r) = \varepsilon + d_{t,r}$ for $t \geq r$ and $\Psi_t(r) = 1$ for $t < r$.

In addition, we choose $g_n(x) = \mathbb{I}[x \leq \varepsilon_n]$ as smoothing function so that we can guarantee that $\Lambda_n \leq \varepsilon_n$ if $J_{n+1} < n$ and for $\varepsilon_n \to 0$, we only need to show the existence of k_n and $m_n \to \infty$ such that $\frac{m_n}{\log n} \to \infty$ and $\mathbb{P}[\mathcal{E}^c_{k_n,m_n}] \to 0$. Now we consider a few different classes of processes and analyze the behaviour of $\mathbb{P}[\mathcal{E}^c_{k_n,m_n}]$ for the defined M-bound. We repeatedly use that $\mathbb{P}[\mathcal{E}^c_{k,m}] = \mathbb{P}[A_k] + \mathbb{P}[B_{k,m}]$ for $A_k = \{J_{n+1} > k\}$ and $B_{k,m} = \left\{J_{n+1} \leq k \wedge \sum_{t=J_{n+1}}^{n} \mathbb{I}[d_{t,J_{n+1}} \leq \varepsilon_n] < m\right\}$. Hence, we can consider the two events separately if needed. The first two examples were already covered by the convergent case, but we still mention them for illustrative purposes.

I.i.d. As noticed above, in this case $d_{i,j} = 0$ for all i, j. This means that J_{n+1} always equals to 1 and we can guarantee that $\mathbb{P}[\mathcal{E}^c_{k,m}] = 0$ for $k = 1$ and $m = n - 1$.

Complete dependence. Let z_1 be a random variable and $z_t = z_1$ for $t > 1$. Then after the first step, the conditional distributions are just delta measures concentrated on the previous point and we always get $J_{n+1} = 2$, so that we obtain $\mathbb{P}[\mathcal{E}^c_{k,m}] = 0$ for $k = 2$ and $m = n - 2$.

Periodic sequences. Consider a periodic deterministic sequence with a fixed period $T \in \mathbb{N}$, like the one obtained by observing the trajectory of a pendulum. Because of periodicity, we know that every conditional distribution occurs at least once within each cycle, therefore, $J_{n+1} \leq T$ is guaranteed and, hence, $\mathbb{P}[\mathcal{E}^c_{k,m}] = 0$ for $k = T$ and $m = \lfloor \frac{n}{T} \rfloor - 1$.

Discrete state Markov chains. For discrete state Markov chains the bounds on the probability of $\mathcal{E}^c_{k,m}$ are deeply connected to the notion of recurrence times. For a state $s \in S$ let T_s be the recurrence time to this state: $T_s = \inf\{t > 1 : z_t = s | z_1 = s\} - 1$. Then the following connection holds:

$$\mathbb{P}[B_{k,m}] \leq |S| m \max_s \mathbb{P}\Big[T_s > \lfloor \frac{n-k}{m} \rfloor\Big]. \tag{3.44}$$

Therefore, the bound can be devised from the concentration properties of the recurrence times. For the other part of $\mathcal{E}^c_{k,m}$ we can show that

$$\mathbb{P}[A_k] \leq |S| \max_s \mathbb{P}[F_s > k], \tag{3.45}$$

where $F_s = \inf\{t \geq 1 : z_t = s\}$ are the first passage times, which also play an important role in the theory of Markov chains, because they reflect how fast the chain explores its state space. In combination,

$$\mathbb{P}\left[E^c_{k,m}\right] \leq |S| \max_s \mathbb{P}[F_s > k] + |S| m \max_s \mathbb{P}\Big[T_s > \lfloor \frac{n-k}{m} \rfloor\Big]. \tag{3.46}$$

For an example of an obtainable rate, we apply Markov's inequality to (3.44),

$$\mathbb{P}[B_{k,m}] \leq \frac{|S| m^2}{n-k} \max_s \mathbb{E}[T_s]. \tag{3.47}$$

One of the basic results from the theory of finite-state Markov chains [Norris, 1998] tells us that some particular state s can be either recurrent or transient, depending on whenever $\mathbb{E}[T_s]$ is finite or not. If all $\mathbb{E}[T_s]$ are finite, then all we need is m growing slower than $\sqrt{n-k}$. This offers a nice connection of the recurrence properties of Markov chains to their learnability.

Dynamical systems. Let $(\mathcal{Z}, \Sigma, \mu, F)$ be a dynamical system, where Σ is a σ-algebra on $\mathcal{Z}$, μ is some measure on $(\mathcal{Z}, \Sigma)$ and $F : \mathcal{Z} \to \mathcal{Z}$ is a measure-preserving transformation, meaning that for any set $A \in \Sigma$ we have $\mu(F^{-1}(A)) = \mu(A)$. The evolution of a system is as follows: first $z_1 \sim \mu$ is sampled and then any subsequent point is obtained through the iteration $z_{t+1} = F(z_t) = F^t(z_1)$. Consequently, $d_{i,j} = \sup_{h \in \mathcal{H}} |\ell(h, F(z_i)) - \ell(h, F(z_j))|$. We assume $d_{i,j} \leq \lambda(z_i, z_j)$ for some metric λ on $\mathcal{Z}$. Let $C_j = \{z \in \mathcal{Z} : \lambda(z, z_j) \leq \varepsilon_n\}$ be a ball around z_j with radius ε_n, then $\mathbb{P}\Big[\mathcal{E}^c_{k,m}\Big]$ is controlled by the first passage times and the recurrence times to the sets C_j (analogously to the discrete Markov chain case). Formally, the recurrence time from a point $z \in \mathcal{Z}$ to a set C is defined as $T(z, C) = \inf\{t \geq 1 : F^t(z) \in C\}$. Then, the first passage time to the set is defined as $F(C) = T(z_1, C)$ and the recurrence time

to a set from itself is $T(C) = \operatorname{ess\,sup}_{z \in C} T(z, C)$. Similarly to the Markov chain case, the following bound holds

$$\mathbb{P}\left[\mathcal{E}_{k,m}^c\right] \leq \mathbb{P}\left[F(C_n) > k\right] + k \max_{1 \leq j \leq k} \mathbb{P}\Big[T(C_j) > \lfloor \frac{n-j}{m} \rfloor\Big]. \tag{3.48}$$

Poincaré's theorem, e.g. [Katok and Hasselblatt, 1997], tells us that any of the sets C_j will be visited infinitely often. A quantitative characterization of the behaviour of the recurrence times for dynamical systems can be found, for example, in [Barreira *et al.*, 2008].

General stationary processes. To relate the setting to the existing work in the nonparametric prediction, assume that the process is stationary and ergodic and $d_{i,j} \leq \lambda(z_{i-q:i-1}, z_{j-q:j-1})$ for some integer q and metric λ on $\mathcal{Z}^q$. For $\bar{z} \in \mathcal{Z}^q$ let $C(\bar{z}) = \{\bar{y} \in \mathcal{Z}^q : \lambda(\bar{y}, \bar{z}) \leq \varepsilon_n\}$. Along the lines of the previous examples, define $F(C) = \inf\{t > q : z_{t-q:t-1} \in C\}$ as a first passage time to a set C. Then we have for $k > q$

$$\mathbb{P}\left[\mathcal{E}_{k,m}^c\right] \leq \mathbb{P}\left[F(C(z_{n-q+1,n})) > k\right] + k \max_{q+1 \leq j \leq q+1+k} \mathbb{P}\left[\sum_{t=q+k+1}^{n} \mathbb{I}\left[z_{t-q:t-1} \in C(z_{j-q,j-1})\right] < m\right]. \tag{3.49}$$

In case of mixing processes, it is possible to determine the rate of recurrence for the second term. More concretely, it can be shown that

$$\sum_{t=k+1}^{n} \mathbb{I}\left[z_{t-q:t-1} \in C(z_{j-q,j-1})\right] \approx \sum_{t=k+1}^{n} \mathbb{P}\left[z_{t-q:t-1} \in C(z_{j-q,j-1})\right] \tag{3.50}$$

$$\geq \inf_{\bar{z}} (n-k) \mathbb{P}\left[z_{t-q:t-1} \in C(\bar{z})\right], \tag{3.51}$$

see for example [Caires and Ferreira, 2005]. Therefore, for mixing processes m can be chosen proportionally to n.

Distribution drift. [Bartlett, 1992] introduced the setting of distributional drift: there is a deterministic sequence of distributions $D_1, \ldots, D_{n+1}$ and samples are drawn independently from the corresponding distribution: $z_i \sim D_i$. Therefore, any conditional expectations is the expectation with respect to the marginal distribution of a point. Since in the distribution drift scenario the samples are independent, the values of J_{n+1} and $\sum_{t=J_{n+1}}^{n} \mathbb{I}\left[d_{t,J_{n+1}} \leq \varepsilon_n\right]$ in the definitions of $\mathcal{E}_{k,m}$ are deterministic. Hence, we can ensure that $\mathbb{P}\left[\mathcal{E}_{k,m}^c\right] = 0$ by trivially setting $k = J_{n+1}$ and $m = \sum_{t=J_{n+1}}^{n} \mathbb{I}\left[d_{t,J_{n+1}} \leq \varepsilon_n\right]$.

3.8.5 Controlling the exceptional set for MACRO

Similarly to the previous section, we now study the behavior of the exceptional set for MACRO. Since MACRO requires only a D-bound, we can just use $M_{i,j} = d_{i,j}$ to isolate the properties of the process from the D-bound assumption. Introduce $A_k = \{|\text{supp}(I_n)| > k\}$ and $B_{k,m} = \{|\text{supp}(I_n)| \leq k \wedge \min_{j \in \text{supp}(I_n)} s_{j,n} > m\}$ so that $\mathbb{P}\left[\mathcal{X}^c_{k,m}\right] = \mathbb{P}[A_k] + \mathbb{P}[B_{k,m}]$. In general, the behavior of the $\mathbb{P}\left[\mathcal{X}^c_{k,m}\right]$ is very similar to that of $\mathbb{P}\left[\mathcal{E}^c_{k,m}\right]$ with slight differences that we highlight on some examples of the previous section.

I.i.d. As all $d_{i,j} = 0$, MACRO uses only a single subroutine, hence for $k = 1$ and $m = n$ we get $\mathbb{P}\left[\mathcal{X}^c_{k,m}\right] = 0$.

Complete dependence. In this case, for sufficiently small ε MACRO will initialize two subroutines: one for the initial distribution and one for all conditional ones. However, the former will not be used after the first step anymore, hence for $k = 1$ and $m = n - 1$ we get $\mathbb{P}\left[\mathcal{X}^c_{k,m}\right] = 0$.

Periodic sequences. For a periodic sequence, the number of the subroutines that MACRO creates is bounded by the length of period T. Hence, we get $\mathbb{P}\left[\mathcal{X}^c_{k,m}\right] = 0$ for $k = T$ and $m = \lfloor \frac{n}{T} \rfloor - 1$.

Discrete state Markov chains. For a Markov chain, MACRO creates at most $|S|$ subroutines, therefore $\mathbb{P}[A_k] = 0$ for $k = |S|$. The bound for $B_{k,m}$ can be obtained by the same argument as for WERM, so we get

$$\mathbb{P}[B_{k,m}] \leq |S|\, m \mathbb{P}\left[T_s > \lfloor \frac{n}{m} \rfloor\right], \tag{3.52}$$

that has the same behavior as in the case of WERM. Together, these bounds give us

$$\mathbb{P}\left[\mathcal{X}^c_{|S|,m}\right] \leq |S|\, m \mathbb{P}\left[T_s > \lfloor \frac{n}{m} \rfloor\right]. \tag{3.53}$$

Dynamical systems. Let us use the notation and the assumption from the corresponding discussion for WERM. Now set $\mathcal{M}(\gamma)$ to be γ-cover of $\mathcal{Z}$ with respect to λ. Then from Lemma 3.8.6 we know that the number of subroutines that MACRO creates can be bounded

by $\mathcal{M}(\varepsilon/2)$, thus for $k = \mathcal{M}(\varepsilon/2)$ we are guaranteed that $\mathbb{P}[A_k] = 0$. If we denote by G_γ an arbitrary γ-ball in $\mathcal{Z}$, we can bound

$$\mathbb{P}[B_{k,m}] \leq \mathcal{M}(\varepsilon/2) \sup_{G_\varepsilon \subseteq \mathcal{Z}} \mathbb{P}\left[T(G_\varepsilon) > \lfloor \frac{n}{m} \rfloor\right]. \tag{3.54}$$

That together gives us

$$\mathbb{P}\left[\mathcal{X}^c_{\mathcal{M}(\varepsilon/2),m}\right] \leq \mathcal{M}(\varepsilon/2) \sup_{G_\varepsilon \subseteq \mathcal{Z}} \mathbb{P}\left[T(G_\varepsilon) > \lfloor \frac{n}{m} \rfloor\right]. \tag{3.55}$$

Distribution drift. Since the sequence of distributions is fixed, the Lemma 3.8.6 gives us the best characterization of the number of subroutines initialized by MACRO. Hence, if we consider such a $\varepsilon/2$-cover of the sequence of $D_1, \ldots, D_n$, k can be set to its size and we can approximate m by the smallest number of distributions that a single element of the cover covers.

3.9 Conclusion

In this chapter we studied the problem of conditional risk minimization. We showed a dichotomy of all stochastic processes into two groups. For one the existing algorithm provably perform well. For another we presented two new algorithms with different characteristics and provable performance guarantees. Additionally, we discussed how the presented bounds behave for a range of well-known classes of stochastic processes.

4 Conditional Risk Minimization in Practice

In the previous chapter we studied the theoretical properties of the algorithms for conditional risk minimization. The goal of the present chapter is to show the practical applicability of these algorithms. In particular, our focus is on the MACRO algorithm as it tackles the computational challenges that arise with weighted empirical risk minimization. We present the two datasets of different characteristics: a binary classification of the delays of the flights and the action recognition on the video streams. In all experiments the goal is to compare MACRO with more traditional approaches, like online learning and standard i.i.d. learning. That being said, we do not focus on achieving the best accuracy for the tasks in favor of fair comparison of the approaches.

We adopt a classification setting: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y}$ is a set of discrete labels, and ℓ is the 0/1-loss. Following the discussions of Section 3.8.3, we use a distance between histories for the D-bound and for each experiment we describe the choice in detail. The version of the algorithm used for theoretical analysis is oblivious to the fact how we initialize the subroutines. In the implementation, however, whenever we start a new subroutine, we give it a warm start by initializing it with the parameters of the closest subroutine in terms of discrepancies.

4.1 DataExpo Airline dataset

The first set of experiments we present uses the *DataExpo Airline* dataset [1], which contains entries about all commercial flights in the United States between 1987 and 2008. Out of these, we select the most recent year with complete data, 2007, and a number of the most active airports at that time, which gives, for example, more than 300000 flights for the Atalanta airport (ATL). The task is a binary classification: predict if a flight is delayed ($y = 1$) or not

[1] http://stat-computing.org/dataexpo/2009/

($y = 0$), where flights count as delayed if they arrive more than 15 minutes later than their scheduled arrival time. Clearly, the temporal order creates dependencies between flight delays that a CRM approach can try to exploit for higher classification accuracy. Observations are defined by grouping the flights into 10 minute chunks, so that at each time step, the task is to output a predictor that is applied to all flights in the next chunk. Formally, the original sequence of flights can be represented as a double sequence of pairs (x_i, t_i) with x_i being a feature vector representing the flight and t_i the real time when the flight departs. We define chunks by grouping the features from the double sequence together if their departure times t_i's fall into the same 10 minute period (in a histogram manner) and then the final stochastic process is a sequence of such grouped feature vectors.

We perform experiments for both types of subroutines that we introduced in Section 3.8.2 and which reflect the go-to choices for online classification problems in practice.

ERM As tractable approximations for ERM we use logistic regression classifiers that are trained incrementally using stochastic gradient descent, i.e.

$$\omega_{t+1} \leftarrow \omega_t + \alpha_t \nabla_\omega \log \mathbb{P}\left[y_t \middle| x_t, \omega\right], \tag{4.1}$$

where ω_t are the parameters of the model at step t and α_t is a learning rate. Since we do not optimize the actual performance, but aim to fairly compare MACRO with a baseline of pure SGD, we set a fixed $\alpha_t = 0.05$ in all runs for both. The conditional probability in (4.1) is a usual logistic regression model:

$$\mathbb{P}\left[y = 1 \middle| x, \omega\right] = \frac{1}{1 + e^{-\langle w, x \rangle}}. \tag{4.2}$$

VW As an online learning subroutine, we use *Vowpal Wabbit*[2], a popular software package for large-scale online learning tasks. We set VW to use logistic loss as well with the default choice of hyper parameters.

As both algorithms are also typical representatives of the classes of algorithms we compare MACRO to, we also use both algorithms when they are run using the whole dataset as baselines. This ensures the fair comparison between the approaches and allows to highlight the effect of using the CRM approach.

[2] `https://github.com/JohnLangford/vowpal_wabbit`

As announced above, we use distances between histories of the process as D-bounds for the experiments. We try out two options: feature- and label-based distances. For the feature-based distance we choose a bottleneck distance, as the number of flights in each chunk changes with time, and approximate it for efficiency in the following way: to compare two sets of vectors, S and T, we compute all pairwise ℓ_2-distances between the elements of S and T, take the smallest $\max\{|S|, |T|\}$ ones and compute their average. Denoting this approximate bottleneck distance as $\bar{\Delta}_1$, the final distance between two chunks is computed as

$$\Delta_1(S,T) = \frac{1}{2}\bar{\Delta}_1(S^0,T^0) + \frac{1}{2}\bar{\Delta}_1(S^1,T^1), \tag{4.3}$$

where S^y and T^y are the subsets of S and T with label y.

For the second distance, Δ_2, we only make use of the labels of the points in the histories. For any history S, define $p(S) = (p_1, p_2)^T$ with p_i being the fraction of the class i in S. Then we set $\Delta_2(S,T) = \|p(S) - p(T)\|^2$.

Figures 4.1 and 4.2 show the results of the evaluating the MACRO with ERM and VW as subroutines comparing to a single ERM and VW algorithms run on the whole data. We see that in all of the presented airports MACRO achieves a better accuracy than the marginal versions of the corresponding algorithms for a wide range of thresholds ε. The effect is most profound with VW subroutine, where MACRO is able to achieve the performance on the level of MACRO with ERM subroutine, even though the VW subroutine itself seems to perform sub-optimally.

In addition to evaluating MACRO for a range of fixed thresholds, we show results for two methods that do not require to fix this parameter. Both methods run a number of MACRO instances with different thresholds in parallel. Formally, we denote by H_i the corresponding instances, i.e. H_i is MACRO run with ε_i, for a range of thresholds $\varepsilon_1, \ldots, \varepsilon_K$, and by $L_t(H_i)$ the cumulative loss of the H_i at step t. We then use two standard online learning strategies: Follow The Leader (FTL) and Exponentially Weighted Average (EWA). FTL strategy chooses the hypothesis produced by the algorithm with the smallest loss, i.e. at step t it chooses $\operatorname{argmin}_{i=1,\ldots,K} L_t(H_i)$. EWA strategy samples the instance proportionally to $e^{-L_t(H_i)/t}$ at each step. The results for these algorithms are presented togehter with the rest in Figures 4.1 and 4.2. Both strategies generally achieve good results, in particular better than marginal training, with the online FTL strategy usually outperforming the EWA strategy and in all cases achieving an error-rate close to the best fixed threshold. Even though both strategies use

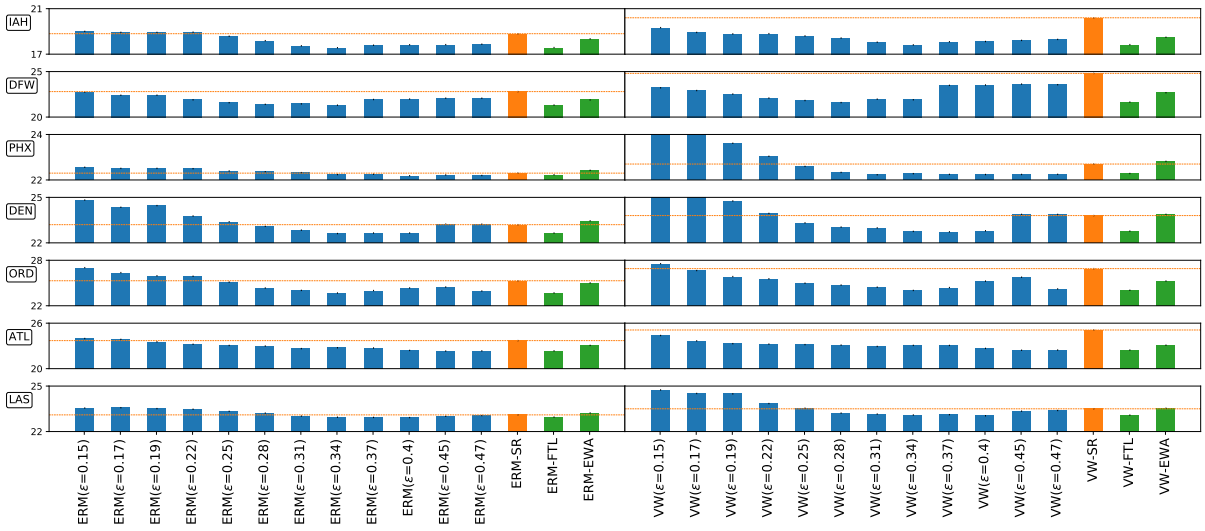

Figure 4.1: Performance of MACRO with different subroutines on the DataExpo Airline dataset with the feature-based distance function. Each row corresponds to a different airport labeled by its IATA code. The y-axes shows error-rates; the x-axes is labeled by the short name of a subroutine and a threshold used in MACRO. ERM-FTL, ERM-EWA, VW-FTL and VW-EWA are the online strategies to choose the threshold. Marginal versions of the subroutines, ERM-SR and VW-SR, act as baselines.

much more resources than a single instance of MACRO, they have the advantage of making the learning process completely parameter-free, and are therefore attractive if sufficient resources are available.

4.2 Breakfast Actions dataset

In this set of experiments we present MACRO in a quite different setting. We use the Breakfast Actions Dataset[3], which consists of videos of 52 people performing 10 actions related to breakfast preparation. The task is a multi-class classification. The dataset comes with two types of annotations: coarse- and fine-grained. The number of labels differs depending on the action and the level of labelling (coarse/fine), For example, the *cereals* task has 4 coarse labels, while *scrambled egg* task has 10. Each combination of a person and an action is treated as a separate learning task and the performance is measured by per frame error rate. Following the usage of a Gaussian assumption by previous approaches [Kuehne *et al.*, 2014; Kuehne *et al.*, 2016], we use Gaussian Naive Bayes classifiers trained online as subroutines.

[3] `http://serre-lab.clps.brown.edu/resource/breakfast-actions-dataset/`

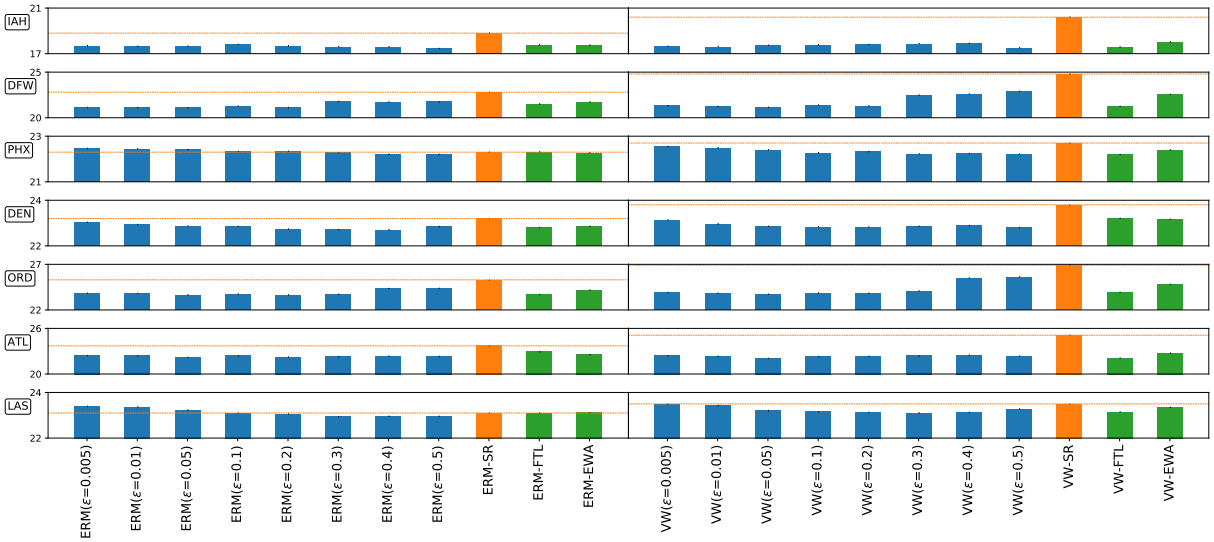

Figure 4.2: Performance of MACRO with different subroutines on the DataExpo Airline dataset with the label-based distance function. Each row corresponds to a different airport labeled by its IATA code. The y-axes shows error-rates; the x-axes is labeled by the short name of a subroutine and a threshold used in MACRO. ERM-FTL, ERM-EWA, VW-FTL and VW-EWA are the online strategies to choose the threshold. Marginal versions of the subroutines, ERM-SR and VW-SR, act as baselines.

G-NB The algorithm tracks the running average in the feature space for each class separately and predicts the class with the closest mean. After receiving a new point, the algorithm incrementally updates the mean of the corresponding class.

The version of G-NB that is run on the whole data is used as a baseline. As for the airports dataset, we present the results for both feature- and label-based distances. For the feature-based one we fix a finite length history, compute the ℓ_2-distance between the vectors on the same position and take the average. Formally, let x_t be the sequence of features and q the history length, then we use $\frac{1}{q}\sum_{t=1}^{q}\|x_{i-t} - x_{j-t}\|_2$ as the D-bound between steps i and j. In the experiments we used the histories of length 5, however, we tried other values and found that the results are not very sensitive to the actual length. The label-based distance defined in the same way as Δ_2, but using the modified embedding $p(S) = (p_1, \ldots, p_K)$, where p_i is a fraction of the class i in the history S and K is the number of classes.

The results are presented in Figures 4.3, 4.4, 4.5 and 4.6. The Figures 4.3 and 4.4 show the performance of MACRO with feature-based distance for coarse and fine annotations. The fine annotations present a bigger challenge, hence explaining the higher overall error rates.

However, the trend is the same in both cases: there is always a region of thresholds where MACRO clearly outperforms the baseline, sometimes by more than 70%. We also evaluate the FTL and EWA strategies for threshold selection that are described in the previous section. Both threshold-selection strategies show excellent performance achieving the error rates close to MACRO with the best fixed threshold. At the same time, FTL consistently outperforms EWA strategy.

The Figures 4.5 and 4.6 show the performance of MACRO with label-based distance. The common feature is more uniform performance of MACRO over the range of thresholds that can be explained by the discrete nature of the distance. Nevertheless, there is always a threshold where MACRO improves over the baseline with FTL and EWA strategies being able to recover the best error rate.

4.3 Conclusion

In this chapter, devoted to the practical aspects of conditional risk minimization, we presented two different practical learning problems and showed that MACRO consistently outperforms the traditional online algorithms for both datasets. This illustrates two facts: CRM is indeed a promising approach to sequential prediction problems, and MACRO's favorable computational complexity allows applying CRM principles to large real-world datasets that WERM is unable to handle.

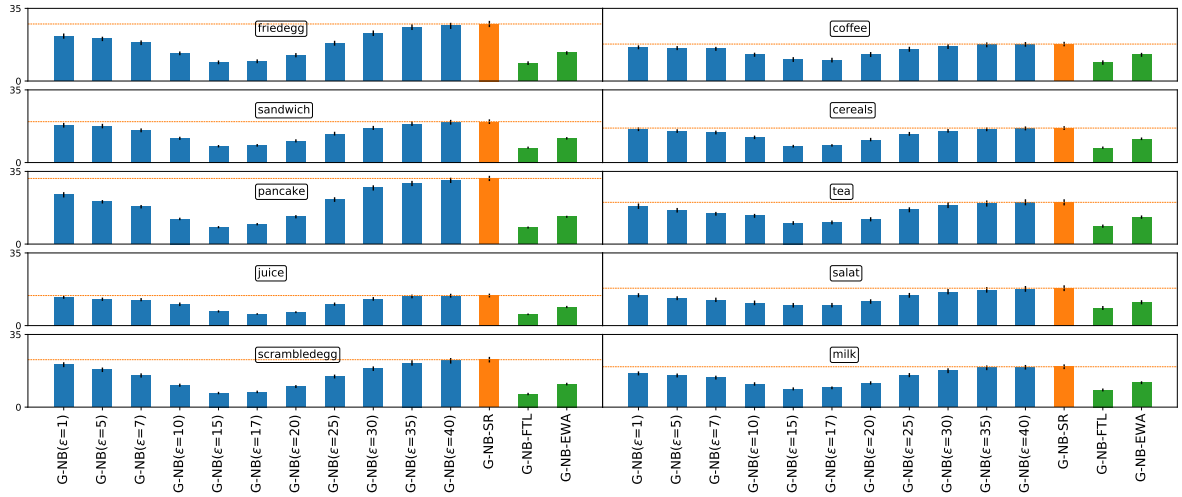

Figure 4.3: Performance of MACRO with different subroutines on the Breakfast Actions dataset with feature-based distance function for coarse annotations. Each plot corresponds to different action. The y-axes shows error-rates averaged over the persons performing each action. The x-axes is labeled by the short name of a subroutine and a threshold used in MACRO. G-NB-FTL and G-NB-EWA represent the online strategies to choose the threshold. The baseline G-NB-SR is the marginal version of G-NB algorithm.

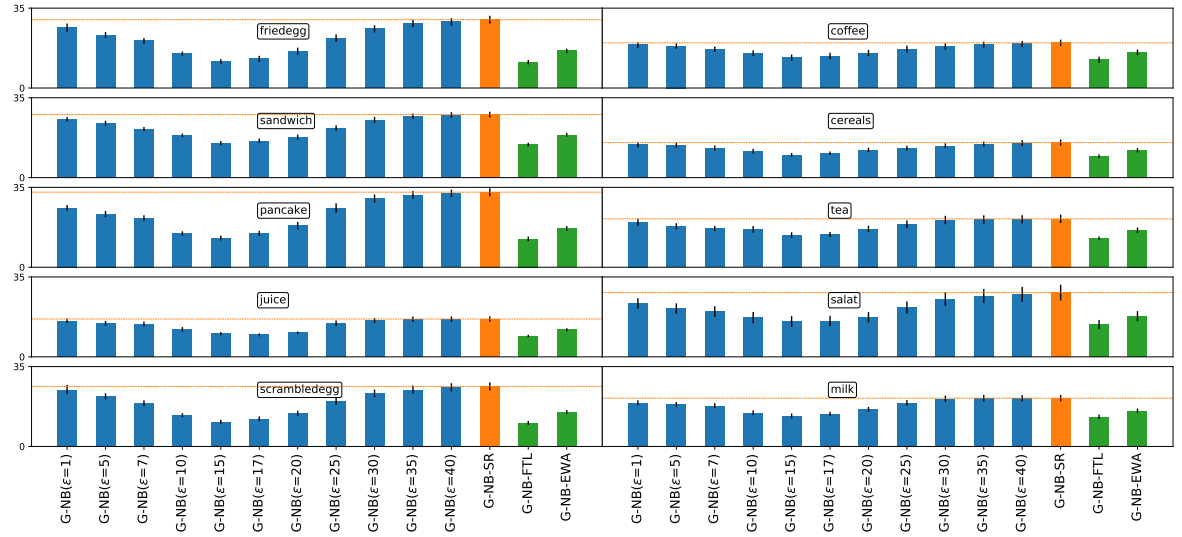

Figure 4.4: Performance of MACRO with different subroutines on the Breakfast Actions dataset with feature-based distance function for fine annotations. Each plot corresponds to different action. The y-axes shows error-rates averaged over the persons performing each action. The x-axes is labeled by the short name of a subroutine and a threshold used in MACRO. G-NB-FTL and G-NB-EWA represent the online strategies to choose the threshold. The baseline G-NB-SR is the marginal version of G-NB algorithm.

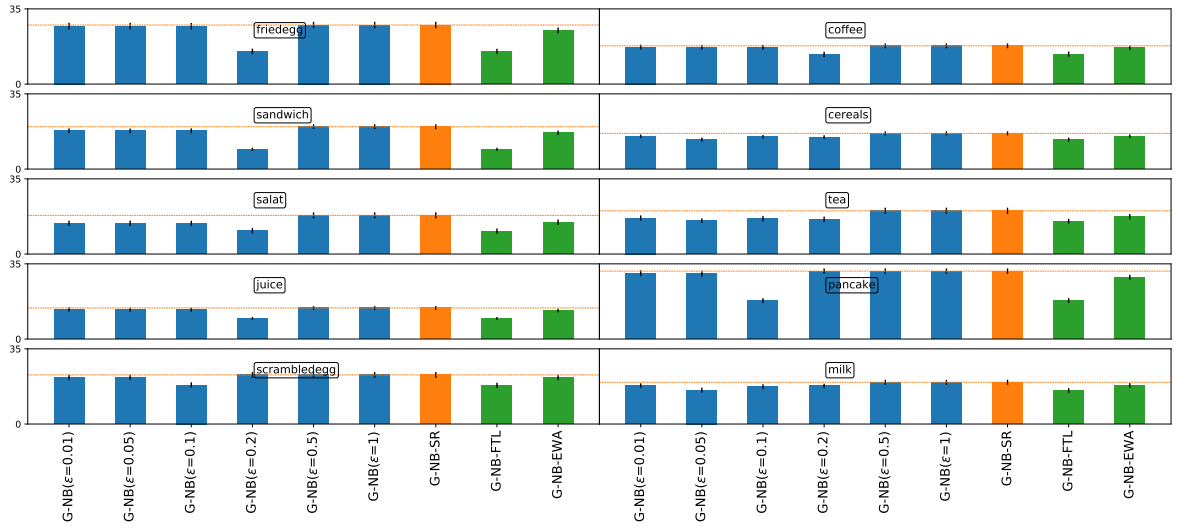

Figure 4.5: Performance of MACRO with different subroutines on the Breakfast Actions dataset with label-based distance function for coarse annotations. Each plot corresponds to different action. The y-axes shows error-rates averaged over the persons performing each action. The x-axes is labeled by the short name of a subroutine and a threshold used in MACRO. G-NB-FTL and G-NB-EWA represent the online strategies to choose the threshold. The baseline G-NB-SR is the marginal version of G-NB algorithm.

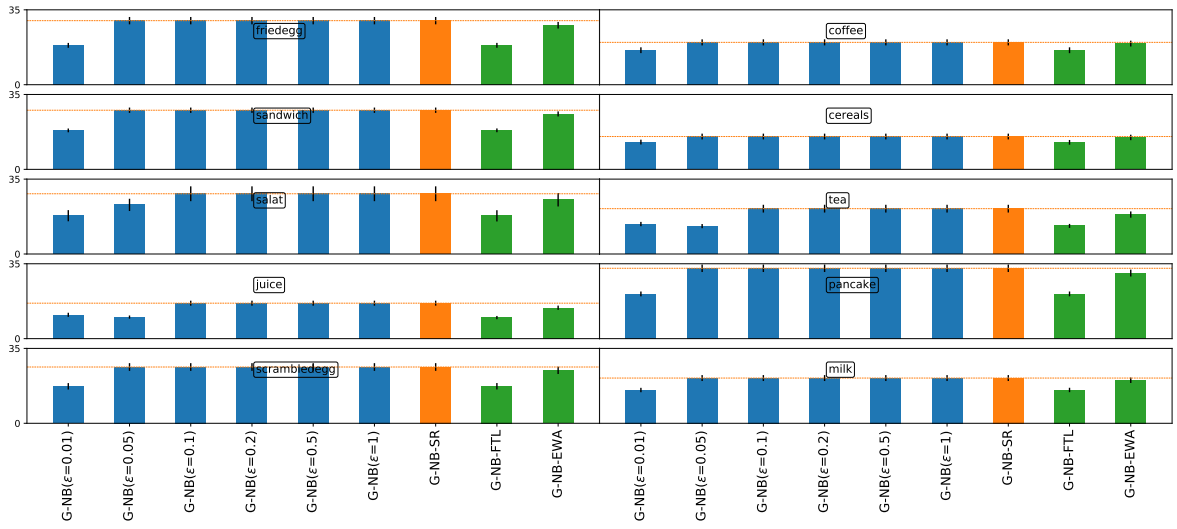

Figure 4.6: Performance of MACRO with different subroutines on the Breakfast Actions dataset with label-based distance function for fine annotations. Each plot corresponds to different action. The y-axes shows error-rates averaged over the persons performing each action. The x-axes is labeled by the short name of a subroutine and a threshold used in MACRO. G-NB-FTL and G-NB-EWA represent the online strategies to choose the threshold. The baseline G-NB-SR is the marginal version of G-NB algorithm.

5 Online Multi-task learning

In this chapter we turn our attention to a multi-task setting where the learner faces a number of possibly related task in online manner. We present a new algorithm, MTLAB, and prove a true risk regret bound in the PAC-Bayes setting. After considering a few exemplar instances of the algorithm, we discuss how to achieve an individual task bounds. Similarly to CRM, the discrepancies between tasks play an important role and we prove that whenever they can be estimated from data it is possible to use the estimates inside MACRO-style algorithm that achieves a favorable per-task performance.

5.1 Multi-task learning of sequential tasks

We face a sequence of tasks $k_1, \ldots, k_n, \ldots$, where each k_t is from a task environment $\mathbb{K}$, and the sequence is a random realization of a stochastic process over $\mathbb{K}$. Note that this general formulation includes the situations most commonly studied in the literature: the case of finitely many fixed tasks (in which case the distribution over the tasks sequence is a delta peak) and the lifelong learning setting with i.i.d. [Baxter, 2000; Pentina and Lampert, 2014] or non-i.i.d. tasks [Pentina and Lampert, 2017].

All tasks share the same *input set* $\mathcal{X}$, *output set* $\mathcal{Y}$, and *hypothesis set* $\mathcal{H}$. Each task k_t, however, has its own associated *joint probability distribution*, D_t, over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, conditioned on k_t. Whenever we *observe* a task k_t, we receive a set $S_t = \{z_{t,i}\}_{i=1}^{m_t}$ sampled i.i.d. from the task distribution D_t, and we are given a loss function, $\ell_t : \mathcal{H} \times \mathcal{Z} \to [0, 1]$ that measures the quality of predictions. Alternatively, one can assume that all tasks share the same, a priori known, loss function.

Learning a task k_t means to identify a hypothesis $h \in \mathcal{H}$ with as small as possible *per-task risk* $R(h, D_t)$. We follow the PAC-Bayes framework described in Section 2.3 that stud-

At any time point $t = 1, 2, \ldots$:

- **receive** dataset S_t
- **output** predictor $\hat{Q}_t$
- **suffer** the loss $R(\hat{Q}_t, D_t)$

Figure 5.1: Online multi-task learning protocol

ies the performance of stochastic (Gibbs) predictors. A stochastic predictor is defined by a probability distribution Q over the hypotheses set with the corresponding risk $R(Q, D_t) = \mathbb{E}_{h\sim Q}[R(h, D_t)]$.

We do not require that data for all tasks is available at the same time. Instead, we adopt an online learning protocol for tasks: at step t we observe the dataset S_t for task k_t, and we output the distribution $\hat{Q}_t$.

5.2 Learning across task boundaries

Our first goal at any step n is to bound the *regret* of a learned sequence of predictors $\hat{Q}_1, \ldots, \hat{Q}_n$ with respect to any fixed reference distribution Q from some set, Δ, of distributions, i.e.

$$\mathcal{R}_n(Q) = \sum_{t=1}^{n} R(\hat{Q}_t, D_t) - \sum_{t=1}^{n} R(Q, D_t). \tag{5.1}$$

This setting resembles online learning discussed in Section 2.6 and is summarized in Figure 5.1. However, there are two striking differences: the learner observes the data for the task *before* it outputs the predictor for the current step and the regret is defined using *true risks*, that we do not observe, in contrast to empirical ones.

The main idea of the algorithm is to run an online learning algorithm on the samples from all tasks, essentially ignoring the task structure of the problem, and then use a properly defined online-to-batch conversion to obtain predictors for the individual tasks. We work with a *Proximal Point Algorithm* [Martinet, 1970] run on the level of samples. Let P be some prior distribution over $\mathcal{H}$. We set $Q_{1,0} = P$ and, once we receive a dataset $S_t = \{z_{t,1}, \ldots, z_{t,m_t}\}$ on step t, we compute

$$Q_{t,i} = \operatorname*{argmin}_{\tilde{Q}\in\Delta} \{ \frac{\eta}{m_t} \mathbb{E}_{h\sim\tilde{Q}}[\ell_t(h, z_{t,i})] + \mathrm{KL}(\tilde{Q}|Q_{t,i-1}) \}, \tag{5.2}$$

Input: decision set Δ, initial distribution P, learning rate η

Initialization: $Q_{1,0} = P$

At any time point $t = 1, 2, \ldots$:

- **receive** dataset S_t of size m_t
- **compute** $Q_{t,i} = \operatorname{argmin}_{\tilde{Q}\in\Delta}\{\frac{\eta}{m_t}\mathbb{E}_{h\sim\tilde{Q}}[\ell_t(h, z_{t,i})] + \mathrm{KL}(\tilde{Q}|Q_{t,i-1})\}$ for $i = 1, \ldots, m_t$
- **output** the batch solution: $\hat{Q}_t \leftarrow \frac{1}{m_t}\sum_{i=1}^{m_t} Q_{t,i}$
- **set** prior of the next task: $Q_{t+1,0} \leftarrow Q_{t,m_t}$

Figure 5.2: MTLAB algorithm

for all $i = 1, \ldots, m_t$ with $\eta > 0$. Afterwards, the algorithm outputs a predictor $\hat{Q}_t = \frac{1}{m_t}\sum_{i=1}^{m_t} Q_{t,i}$ for task t, and sets $Q_{t+1,0} = Q_{t,m_t}$, to be used as a starting distribution for the next task.

As an example, when Δ is the set of all probability distributions, the minimizer of (5.2) has the explicit form

$$dQ_{t,i}(h) = \frac{e^{-\frac{\eta}{m_t}\ell_t(h,z_{t,i})}dQ_{t,i-1}(h)}{\int e^{-\frac{\eta}{m_t}\ell_t(h',z_{t,i})}dQ_{t,i-1}(h')}. \tag{5.3}$$

We call the above procedure MTLAB (multi-task learning across task boundaries) and summarize it in Figure 5.2. Our first result is a regret bound for the true risks of the sequence of distributions that it produces.

Theorem 5.2.1. *Let $\bar{m} = n/(\sum_{t=1}^{n} 1/m_t)$ be the harmonic mean of $m_1, \ldots, m_n$ and let P be a fixed prior distribution that is chosen independently of the data. The predictors produced by MTLAB satisfy with probability $1-\delta$ (over the random training sets) uniformly over $Q \in \Delta$*

$$\mathcal{R}_n(Q) \leq \frac{\eta n}{4\bar{m}} + \frac{2KL(Q|P) + \log\frac{2}{\delta}}{\eta}. \tag{5.4}$$

Corollary 5.2.2. *Set $\eta = \sqrt{\frac{\bar{m}}{n}}$. Then, with probability $1-\delta$, it holds uniformly over $Q \in \Delta$*

$$\frac{1}{n}\mathcal{R}_n(Q) \leq \frac{1}{\sqrt{n\bar{m}}}(\frac{1}{4} + 2KL(Q|P) + \log\frac{2}{\delta}). \tag{5.5}$$

To put this result into perspective, we compare it to the average regret bounds given in [Alquier *et al.*, 2017], where the goal is find the best possible data representation for tasks. Even though the settings are a bit different, it gives a good idea of the qualitative nature of our result. [Alquier *et al.*, 2017] provides $\mathcal{O}(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}})$ bound (if all tasks are of the same size m) that can be sometimes improved to $\mathcal{O}(\frac{1}{\sqrt{n}} + \frac{1}{m})$. In either case, convergence happen

only in the regime that the number of tasks *and* the amount of data for each task both tend to infinity. In contrast to this, the right hand side of inequality (5.5) converges to zero even if only one of the two quantities grows, so in particular for the most common case that the number of tasks grows to infinity, but the amount of data per task remains bounded.

5.3 Connection to traditional PAC-Bayes bounds

We obtain further insight into the behavior of MTLAB by comparing it to the situation in which each task is learned independently. A more traditional PAC-Bayes bound, like the one in Theorem 2.3.1, gives us the following bound with probability $1 - \delta$

$$R(Q, D_t) \leq \frac{1}{m_t} \sum_{i=1}^{m_t} \mathbb{E}_Q \left[\ell_t(h, z_{t,i})\right] + \frac{\mathrm{KL}(Q|P) + \log \frac{1}{\delta}}{\sqrt{m_t}}. \tag{5.6}$$

This inequality suggests a learning algorithm, namely to minimize the upper bound with respect to Q. In principle, MTLAB is based on a similar objective, but it acts on the sample level and it automatically provides relevant prior distributions for each task. Thereby it is able to achieve better guarantee than one could get by combining separate bounds of the form (5.3) for multiple tasks.

5.4 MTLAB for lifelong learning

The bound of Theorem 5.2.1 holds for any stochastic process over the tasks. In particular, it holds in special case where tasks are sampled independently from a hyper distribution over the task environment, which is usually called *lifelong learning* [Baxter, 2000; Pentina and Lampert, 2014]. In this setting, we have a fixed distribution $\mathcal{T}$ over $\mathbb{K}$, and the sequence $k_1, \ldots, k_n$ is an i.i.d. sample from this distribution. One can then define the *lifelong risk* as

$$\mathcal{E}(h) = \mathbb{E}_{k \sim \mathcal{T}} \left[\mathbb{E}_{z \sim D_k} \left[\ell_k(h, z)\right]\right], \tag{5.7}$$

where D_k and ℓ_k are the distribution and loss function for a task k, respectively. The risk of the Gibbs predictor is then $\mathcal{E}(Q) = \mathbb{E}_{h \sim Q} \left[\mathcal{E}(h)\right]$. Let $\hat{Q}_1, \ldots, \hat{Q}_n$ be the output of MTLAB, then we define the corresponding batch solution as $\bar{Q}_n = \frac{1}{n} \sum_{t=1}^{n} \hat{Q}_t$ and observe

$$\mathcal{E}(\bar{Q}_n) = \frac{1}{n} \sum_{t=1}^{n} \mathcal{E}(\hat{Q}_t) = \mathbb{E}\left[\frac{1}{n} \sum_{t=1}^{n} R(\hat{Q}_t, D_t)\right]. \tag{5.8}$$

Using Theorem 5.2.1 we obtain the following guarantee.

Theorem 5.4.1. *In the lifelong learning setting, if we run MTLAB with* $\eta = \frac{\sqrt{\bar{m}}}{\sqrt{n}}$*, for any fixed prior distribution* P *that is chosen independently from the data, with probability* $1-\delta$ *uniformly over* $Q \in \Delta$

$$\mathcal{E}(\bar{Q}_n) - \mathcal{E}(Q) \leq \frac{1}{\sqrt{n\bar{m}}}\left(\frac{1}{4} + 2KL(Q|P) + \log\frac{2}{\delta}\right). \tag{5.9}$$

Typical results for this setting, such as shown in [Pentina and Lampert, 2014; Maurer *et al.*, 2016; Alquier *et al.*, 2017], show the convergence rate $\mathcal{O}(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{\bar{m}}})$, which goes to zero only in the case of infinite data *and* infinite tasks. In contrast, the generalization error for MTLAB converges in the most realistic scenario of finite data per task and increasing number of tasks.

5.5 Examples

Before continuing our theoretical analysis by providing performance bounds for individual tasks, we would like to provide two illustrative examples on how real-world implementations of MTLAB could look like.

Stochastic Neural Networks We first illustrate an implementation of MTLAB in a deep learning context. Following the presentation in [Amit and Meir, 2017], we make use of stochastic neural networks. Let $\mathcal{H} = \{h_\omega, \omega \in \mathbb{R}^d\}$, and assume that the loss function is differentiable with respect to ω. We think of h_ω as a neural network with d weights and we take Δ as the set of distributions over ω of the form

$$Q(\omega) = \prod_{j=1}^{d} \mathcal{N}(\omega_j|\mu_j, \sigma_j^2), \tag{5.10}$$

where $\mathcal{N}(\omega|\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. The main question is how to perform the optimization step (5.2). First, we note that the KL-divergence has a closed form expression in this case: for two distributions $Q_1(\omega) = \prod_{j=1}^{d} \mathcal{N}(\omega_j|\mu_{1,j}, \sigma_{1,j}^2)$ and $Q_2(\omega) = \prod_{j=1}^{d} \mathcal{N}(\omega_j|\mu_{2,j}, \sigma_{2,j}^2)$, it holds

$$\mathrm{KL}(Q_1|Q_2) = \frac{1}{2}\sum_{j=1}^{d}\left(\log\frac{\sigma_{2,j}^2}{\sigma_{1,j}^2} + \frac{\sigma_{1,j}^2 + (\mu_{1,j}-\mu_{2,j})^2}{\sigma_{2,j}^2} - 1\right). \tag{5.11}$$

Second, to be able to differentiate $\mathbb{E}_{\omega\sim Q}\left[\ell_t(h_\omega, z_{t,i})\right]$ with respect to the parameters of Q, we employ the re-parameterization trick [Kingma *et al.*, 2015], that converts a function with stochastic behavior into a deterministic function of an additional stochastic input. MTLAB

tells us to perform the optimization for all data points within one task and to then average the resulting distributions. While the resulting mixture distribution does not have a simple parametric form, it is easy use as a Gibbs predictor: to sample from it one first samples a random index from 1 to m_t and then samples network parameters from the corresponding Gaussian distribution.

Linear predictors A second practical setting for MTLAB is classification with linear predictors. Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$ and $\mathcal{H} = \{h(x) = \text{sign}\langle\omega, x\rangle, \omega \in \mathbb{R}^d\}$ with $\ell_t(a, b) = \mathbb{I}[a \neq b]$. Following [Germain *et al.*, 2009], we restrict Δ to Gaussian distributions with unit variance and we use $P = \mathcal{N}(0, I_d)$ and $Q_{t,i} = \mathcal{N}(\mu_{t,i}, I_d)$. Then $\text{KL}(Q_{t,i}|Q_{t,i-1}) = \frac{\|\mu_{t,i} - \mu_{t,i-1}\|_2^2}{2}$ and $\mathbb{E}_Q[\ell_t(h, (x_{t,i}, y_{t,i}))] = \Phi(\frac{y_{t,i}\langle\mu, x_{t,i}\rangle}{\|x_{t,i}\|_2})$ with $\Phi(a) = \frac{1}{\sqrt{2\pi}}\int_a^{+\infty}\exp(-\frac{1}{2}x^2)dx$. Therefore, MTLAB computes as each step

$$\min_{\mu\in\mathbb{R}^d} \quad \frac{\eta}{m_t}\Phi\left(\frac{y_{t,i}\langle\mu, x_{t,i}\rangle}{\|x_{t,i}\|_2}\right) + \frac{\|\mu - \mu_{t,i-1}\|_2^2}{2}. \tag{5.12}$$

Note that, as commonly done in algorithms inspired by PAC-Bayes theory, after learning one can convert the randomized predictors into deterministic ones, by using the distribution's mean parameters, μ, as weight vectors. This preserves the guarantees up to a constant factor, see [Langford and Shawe-Taylor, 2003] for details.

5.6 Per-task bounds

The results of Section 5.2 provide guarantees on MTLAB's multi-task regret. In this section we compliment those results by presenting a modification that provides guarantees for individual risks of each task.

As a start, let us consider a bound that can be obtained immediately from Theorem 5.2.1. We make use of the following notion of relatedness between tasks that is similar to Definition 3.6.1.

Definition 5.6.1. *For a fixed hypothesis class $\mathcal{H}$, the* discrepancy *between tasks k_i and k_j is defined as*

$$disc(k_i, k_j) = \sup_{h\in\mathcal{H}} |R(h, D_i) - R(h, D_j)|. \tag{5.13}$$

The following theorem is an immediate corollary of Theorem 5.2.1.

Theorem 5.6.2. *Let P be a fixed prior distribution that is chosen independently of the data. Let $\hat{Q}_t$ be a sequence of predictors produced by MTLAB run with $\eta = \sqrt{\frac{\bar{m}}{n}}$ and let $\bar{Q}_n = \frac{1}{n}\sum_{t=1}^{n} \hat{Q}_t$. Then the following inequality holds with probability $1-\delta$, uniformly over $Q \in \Delta$*

$$R(\bar{Q}_n, D_n) \leq R(Q, D_n) + \frac{2}{n}\sum_{i=1}^{n} disc(k_i, k_n) + \frac{1}{\sqrt{n\bar{m}}}(\frac{1}{4} + 2KL(Q|P) + \log\frac{2}{\delta}). \qquad (5.14)$$

This bound resembles the guarantees typical in the setting of *learning from drifting distributions* [Mohri and Medina, 2012]. It converges if $\frac{1}{n}\sum_{i=1}^{n} \mathrm{disc}(k_i, k_n) \to 0$ with n, so if either tasks are identical to each other, or if tasks get suitably more similar on average with growing n. This resembles the convergent case of CRM (Section 3.7).

The main question of this section is if we can improve upon the bound of Theorem 5.6.2 in the case when $\frac{1}{n}\sum_{i=1}^{n} \mathrm{disc}(k_i, k_n)$ does not vanish over time. Consider, for example, a simple case of two alternating tasks, i.e. $\frac{1}{n}\sum_{i=1}^{n} \mathrm{disc}(k_i, k_n) \to \frac{1}{2}$ for $n \to \infty$. If we split the sequence of tasks into two subsequences, one for tasks with even and one for tasks with odd indices, and then run MTLAB separately for each sequence, we could nevertheless guarantee the convergence of the error rate for the resulting procedure.

Unfortunately, it is rather easy to construct examples in which convergence to zero is not achievable, even with the best possible split of the sequence of tasks into subsequences: for example, if simply all tasks differs by at least ε from each other in the discrepancy sense, then no matter what split we use, it will not be possible to achieve an upper bound below ε. Consequently, we redefine our goal to prove error rates that converge below a given threshold ε.

We present an online algorithm, MTLAB.MS (for *MTLAB with Multiple Sequences*), that splits the tasks into subsequences on the fly given some distance $\mathrm{dist}(k, k')$ between tasks. In Sections 5.6.1 and 5.6.2 we will discuss two ways to construct such distances and we will prove the risk bounds for MTLAB.MS run with the corresponding distance. Following the MACRO methodology, Section 3.8.2, we keep a representative task for each subsequence, and we use the distances to the representatives to decide which subsequence to extend with the new task, or if a new subsequence needs to be initialized.

Pseudo-code for MTLAB.MS is provided in Figure 5.3. The notation $\tilde{Q}, P' = \mathrm{MTLAB}(S, P)$ denotes a single run of MTLAB that takes a dataset S, runs its learning procedure starting from distribution P and outputs two distributions: the final distribution P' to be used in the

Input: task distance dist, prior distribution P, threshold ε

Initialization: set of representative tasks $R = \emptyset$, set of priors $\mathcal{P} = \emptyset$

At any time point $t = 1, 2, \ldots$:

- **receive** dataset S_t.
- **set** $\mathcal{I} = \{r \in R : \text{dist}(k_r, k_t) \leq \varepsilon\}$
- **if** $\mathcal{I} = \emptyset$ **then**
  - add t to the set of representatives R
  - set $\mathcal{P}(t) = P$
- **choose** the closest representatives $r^\star = \text{argmin}_{r \in \mathcal{I}} \text{dist}(k_r, k_t)$
- **run** the transfer algorithm: $\bar{Q}_t, P' = \text{MTLAB}(S_t, \mathcal{P}(r^\star))$
- **set** $\mathcal{P}(r^\star) = P'$
- **output** $\bar{Q}_t$

Figure 5.3: MTLAB.MS algorithm

subsequent runs and the aggregate distribution $\tilde{Q}$ that is a final predictor for the task. Further notation used are: $\mathcal{I}_n$ are the indices of the tasks in the subsequence chosen at step n, $s_n = |\mathcal{I}_n|$ is the size of this subsequence, $\bar{m}_n$ is the harmonic average of the sizes of tasks in the chosen subsequence and η_n is the learning rate of MTLAB associated with the chosen subsequence.

The following theorem shows that if MTLAB.MS is (could be) run with the task discrepancies as distances, it would, for any given threshold ε, yield subsequences with generalization error below ε.

Theorem 5.6.3. *Let P be a fixed prior distribution that is chosen independently of the data. If we run MTLAB.MS with dist$(k_i, k_j) =$ disc(k_i, k_j), we get with probability $1 - \delta$, uniformly over $Q \in \Delta$*

$$R(\bar{Q}_n, D_n) \leq R(Q, D_n) + 2\varepsilon + \frac{2\eta_n}{\bar{m}_n} + \frac{2KL(Q|P) + \log \frac{n}{\delta}}{\eta_n s_n}. \tag{5.15}$$

In practice, however, the true discrepancy values are unknown. Therefore, we present two approaches for estimating them: based on labelled and based on unlabelled data.

5.6.1 Estimation from labelled data

The most direct method to determine the right subsequence for each task is to estimate the discrepancies from the data and use the estimates in the MTLAB.MS algorithm. However, choosing the subsequence is equivalent to choosing a prior distribution for the next task. For the guarantees of the theory to hold, that step needs to be done independently of the labeled data used for learning that task. We overcome this problem by splitting the labeled data into two subsets: one for estimating the discrepancy (denoted by $\hat{S}_t$) and one for learning (denoted by S_t). With $\hat{m}_t = |\hat{S}_t|$ and $m_t = |S_t|$ we define the discrepancy estimates as

$$\widehat{\text{disc}}_{i,j} = \sup_{h \in \mathcal{H}} \left| \frac{1}{\hat{m}_i} \sum_{z \in \hat{S}_i} \ell_i(h, z) - \frac{1}{\hat{m}_j} \sum_{z \in \hat{S}_j} \ell_j(h, z) \right|. \tag{5.16}$$

The standard uniform convergence bounds can be leveraged to guarantee the quality of this estimation. For example, using Theorem 8 of [Bartlett and Mendelson, 2002] we can show that with probability $1 - \delta$

$$\text{disc}(k_i, k_j) \leq \widehat{\text{disc}}_{i,j} + B_L(\hat{S}_i, \hat{S}_j, \delta), \tag{5.17}$$

where the estimation error B_L is defined as

$$B_L(\hat{S}_i, \hat{S}_j, \delta) = \hat{\mathfrak{R}}_{\hat{S}_i}(\mathcal{L}_i(\mathcal{H})) + \hat{\mathfrak{R}}_{\hat{S}_j}(\mathcal{L}_j(\mathcal{H})) + 3\sqrt{\frac{8 \log \frac{2}{\delta}}{|\hat{S}_i|}} + 3\sqrt{\frac{8 \log \frac{2}{\delta}}{|\hat{S}_j|}} \tag{5.18}$$

with $\mathcal{L}_t(\mathcal{H}) = \{(x, y) \to \ell_t(h, x, y), \forall h \in \mathcal{H}\}$. Now we can prove the following theorem for MTLAB.MS used with $\widehat{\text{disc}}_{i,j}$ as task distances.

Theorem 5.6.4. *Let P be a fixed prior distribution that is chosen independently of the data. If we run MTLAB.MS with $\text{dist}_{k_i,k_j} = \widehat{disc}_{i,j}$, we get with probability $1 - 2\delta$ uniformly over $Q \in \Delta$*

$$R(\bar{Q}_n, D_n) \leq R(Q, D_n) + 2\varepsilon + \frac{2\eta_n}{\bar{m}_n} + \frac{2KL(Q|P) + \log \frac{n}{\delta}}{\eta_n s_n} + \frac{1}{s_n} \sum_{t \in \mathcal{I}_n} B_L(\hat{S}_t, \hat{S}_n, \frac{\delta}{n}). \tag{5.19}$$

Remark. Theorem 5.6.4 works when the transfer algorithm uses a fixed learning rates η for each subsequence. It is possible to prove a similar statement for the case when the parameters are optimized for the length of each subsequence using the machinery developed in Section 3.8. However, the final statement gets more complicated and adds little to the discussions in the current chapter. Therefore, we leave this extension for future work.

5.6.2 Estimation from unlabelled data

In many situations labeled data is scarce and setting aside a part of the training set for discrepancy estimation would leave us with too little data for training. In such situations, however, it might be rather easy to obtain at least unlabelled data. In the case that learning tasks are deterministic realizable, this additional data can be used to estimate the discrepancy between tasks before we observe the labeled data for the new task. The deterministic case assumes D_t to be a distribution over $\mathcal{X}$ only, while the target is determined by a fixed labelling function $f_t : \mathcal{X} \to \mathcal{Y}$. The loss of a hypothesis then can be written as $\ell_t(h,(x,y)) = \ell_t(h,(x,f_t(x))) = \ell_t(h,f_t,x)$ and we will stick to that notation for this section. The realizable scenario assumes $f_t \in \mathcal{H}$. For this setting, we use the following definition of discrepancies [Ben-David *et al.*, 2007; Mohri and Medina, 2012].

Definition 5.6.5. *The symmetric discrepancy between tasks k_i and k_j is defined as*

$$e(k_i,k_j) = \sup_{h_1,h_2\in\mathcal{H}} \left| \mathbb{E}_{x\sim D_i}\left[\ell_i(h_1,h_2,x)\right] - \mathbb{E}_{x\sim D_j}\left[\ell_j(h_1,h_2,x)\right] \right|. \tag{5.20}$$

Let U_i and U_j be i.i.d. unlabelled sample sets from the task distributions D_i and D_j, respectively. Then we can estimate the symmetric discrepancies by [Kifer *et al.*, 2004]

$$\hat{e}(k_i,k_j) = \sup_{h_1,h_2\in\mathcal{H}} \left| \frac{1}{|U_i|}\sum_{x\in U_i}\ell_i(h_1,h_2,x) - \frac{1}{|U_j|}\sum_{x\in U_j}\ell_j(h_1,h_2,x) \right|. \tag{5.21}$$

We then pass these estimates as values of dist_{k_i,k_j} to MTLAB.MS. Similarly to the setting with labelled data, the proposed estimates can be proven to estimate $e(k_i,k_j)$ quite closely. Proposition 2 from [Mansour *et al.*, 2009] gives us that with probability $1-\delta$:

$$e(k_i,k_j) \le \hat{e}(k_i,k_j) + B_U(U_i,U_j,\delta), \tag{5.22}$$

with

$$B_U(U_i,U_j,\delta) = \hat{\mathfrak{R}}_{U_i}(\mathcal{L}_i^s(\mathcal{H})) + \hat{\mathfrak{R}}_{U_j}(\mathcal{L}_j^s(\mathcal{H})) + \frac{3\log\frac{4}{\delta}}{\sqrt{2\,|U_i|}} + \frac{3\log\frac{4}{\delta}}{\sqrt{2\,|U_j|}}, \tag{5.23}$$

where $\mathcal{L}_t^s(\mathcal{H}) = \{x \to \ell(h,h',x), h,h' \in \mathcal{H}\}$. To compensate for the lack of label information, we additionally need the notion of an ideal joint hypothesis, i.e. the one that minimizes joint error for two tasks:

$$\lambda_{i,j} = \min_{h\in\mathcal{H}}\{R(h,D_i) + R(h,D_j)\}. \tag{5.24}$$

Overall, we obtain the following performance guarantee for MTLAB.MS when run in this setting.

Theorem 5.6.6. *Let P be a fixed prior distribution that is chosen independently of the data. In the deterministic realizable case, if we run MTLAB.MS with $dist_{k_i,k_j} = \hat{e}_{i,j}$, we get with probability $1-2\delta$, uniformly over $Q \in \Delta$:*

$$R(\bar{Q}_n, D_n) \leq R(Q, D_n) + 2\varepsilon + \frac{2\eta_n}{\bar{m}_n} + \frac{2KL(Q|P) + \log\frac{n}{\delta}}{\eta_n s_n} + \frac{1}{s_n}\sum_{t\in\mathcal{I}_n}\lambda_{t,n} + \frac{1}{s_n}\sum_{t\in\mathcal{I}_n} B_U(U_t, U_n, \frac{\delta}{n}). \tag{5.25}$$

5.7 Conclusion

In this chapter we presented a new algorithm for online mutli-task learning and proved a true risk regret bound for it. The main feature of the bound is that it is sublinear simultaneously in the number of the tasks and their sizes. In addition, we presented a version of the algorithm that achieves favorable per-task bounds utilizing the estimates of the discrepancies between the tasks.

6 Conclusion and Future Work

In this thesis we made a further step towards bridging the gap between the theory and practice in machine learning. We challenged the most fundamental assumption made in statistical learning theory that is the independence of the data. We studied different learning settings that involve dependence, giving a particular focus to stochastic processes, and introduced new algorithms in single- and multi-task frameworks. Our main contributions include the theoretical guarantees on the performance of new algorithms as well as their empirical evaluation.

There is a number of directions for extension of the present thesis. Stochastic processes is only a single model for the data generation process. There are other models that are better suited for particular applications: spatial models (e.g. [Banerjee *et al.*, 2008]), dependency graphs (e.g. [Ralaivola *et al.*, 2010]), graphical models (e.g. [London *et al.*, 2013]), etc.

In all the learning settings in this work the prediction made by the learning algorithm do not affect the future values of the data. However, this is not always the case, with reinforcement learning being a good example. It is an interesting direction to consider an intermediate setting that allows to consider general stochastic processes (generalizing RL) with the predictions affecting the future (generalizing CRM).

The conditional risk minimization problem itself is far from being solved. For example, an interesting question is to come up with an algorithm that is practical and is able to achieve conditional learnability. The key assumption we explored for CRM is the existence of a D-bound, however, it is an open question of how to choose a D-bound in a particular application. A data driven approach for this would be great step towards the wide applicability of CRM principle. On a more general note, an interesting study is to determine the algorithm-oblivious properties of a process that characterize conditional learnability. For example, so far, WERM and MACRO use different notions of exceptional sets that are tailored for each algorithm specifically. We believe that there is a fundamental characterization of these sets that is

independent of the algorithm and made a first step towards such characterization in Lemma 3.8.6 by connecting MACRO computational complexity to process properties.

Bibliography

[Alon *et al.*, 1997] Noga Alon, Shai Ben-David, Nicolo Cesa-Bianchi, and David Haussler, "Scale-sensitive dimensions, uniform convergence, and learnability," *Journal of the ACM (JACM)*, 44(4):615–631, 1997.

[Alquier *et al.*, 2013] Pierre Alquier, Xiaoyin Li, and Olivier Wintenberger, "Prediction of time series by statistical learning: general losses and fast rates," *Dependence Modeling*, 1:65–93, 2013.

[Alquier *et al.*, 2017] Pierre Alquier, The Tien Mai, and Massimiliano Pontil, "Regret Bounds for Lifelong Learning," In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[Amit and Meir, 2017] Ron Amit and Ron Meir, "Meta-Learning by Adjusting Priors Based on Extended PAC-Bayes Theory," *arXiv preprint arXiv:1711.01244*, 2017.

[Banerjee *et al.*, 2008] Sudipto Banerjee, Alan E Gelfand, Andrew O Finley, and Huiyan Sang, "Gaussian predictive process models for large spatial data sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):825–848, 2008.

[Barreira *et al.*, 2008] Luis Barreira, A Weinstein, H Bass, and J Oesterl, *Dimension and recurrence in hyperbolic dynamics*, Springer, 2008.

[Bartlett, 1992] Peter L Bartlett, "Learning with a slowly changing distribution," In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 243–252. ACM, 1992.

[Bartlett *et al.*, 1996] Peter L Bartlett, Philip M Long, and Robert C Williamson, "Fat-shattering and the learnability of real-valued functions," *Journal of Computer and System Sciences*, 52(3):434–452, 1996.

[Bartlett and Mendelson, 2002] Peter L Bartlett and Shahar Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, 3:463–482, 2002.

[Baxter, 2000] J. Baxter, "A model of inductive bias learning," *Journal of Artificial Intelligence Research (JAIR)*, 12:149–198, 2000.

[Ben-David *et al.*, 2010] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan, "A theory of learning from different domains," *Machine Learning*, 79(1-2):151–175, 2010.

[Ben-David *et al.*, 2007] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira, "Analysis of representations for domain adaptation," In *Conference on Neural Information Processing Systems (NIPS)*, 2007.

[Berti *et al.*, 2002] Patrizia Berti, A Mattei, and Pietro Rigo, "Uniform convergence of empirical and predictive measures," *Atti del Seminario Matematico e Fisico dell'Universita' di Modena*, 50(2):465–478, 2002.

[Berti and Rigo, 1997] Patrizia Berti and Pietro Rigo, "A Glivenko-Cantelli theorem for exchangeable random variables," *Statistics & Probability Letters*, 32(4):385–391, 1997.

[Berti and Rigo, 2017] Patrizia Berti and Pietro Rigo, "Asymptotic predictive inference with exchangeable data," *Brazilian Journal of Probability and Statistics*, 2017.

[Caires and Ferreira, 2005] S Caires and JA Ferreira, "On the non-parametric prediction of conditionally stationary sequences," *Statistical inference for stochastic processes*, 8(2):151–184, 2005.

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

[Cesa-Bianchi *et al.*, 2004] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile, "On the generalization ability of on-line learning algorithms," *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

[Cesa-Bianchi and Lugosi, 2006] Nicolo Cesa-Bianchi and Gabor Lugosi, *Prediction, learning, and games*, Cambridge University Press, 2006.

[Csiszár, 1975] Imre Csiszár, "I-divergence geometry of probability distributions and minimization problems," *Annals of Probability*, pages 146–158, 1975.

[Germain *et al.*, 2009] Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand, "PAC-Bayesian learning of linear classifiers," In *International Conference on Machine Learing (ICML)*, 2009.

[Gyorfi *et al.*, 1998] L Gyorfi, Gusztáv Morvai, and Sidney J Yakowitz, "Limits to consistent on-line forecasting for ergodic time series," *IEEE Transactions on Information Theory*, 44(2):886–892, 1998.

[Györfi *et al.*, 1989] László Györfi, Wolfgang Härdle, Pascal Sarda, and Philippe Vieu, *Nonparametric curve estimation from time series*, volume 60, Springer-Verlag Berlin, 1989.

[Györfi *et al.*, 2002] László Györfi, Adam Krzyzak, Michael Kohler, and Harro Walk, *A distribution-free theory of nonparametric regression*, Springer, 2002.

[Hansen, 2008] Bruce E Hansen, "Uniform convergence rates for kernel estimation with dependent data," *Econometric Theory*, 24(03):726–748, 2008.

[Katok and Hasselblatt, 1997] Anatole Katok and Boris Hasselblatt, *Introduction to the modern theory of dynamical systems*, volume 54, Cambridge University Press, 1997.

[Khaleghi and Ryabko, 2016] Azadeh Khaleghi and Daniil Ryabko, "Nonparametric multiple change point estimation in highly dependent time series," *Theoretical Computer Science*, 620:119–133, 2016.

[Kifer *et al.*, 2004] Daniel Kifer, Shai Ben-David, and Johannes Gehrke, "Detecting change in data streams," In *International Conference on Very Large Data Bases (VLDB)*, volume 30, pages 180–191, 2004.

[Kingma *et al.*, 2015] Diederik P Kingma, Tim Salimans, and Max Welling, "Variational dropout and the local reparameterization trick," In *Conference on Neural Information Processing Systems (NIPS)*, 2015.

[Klenke, 2013] Achim Klenke, *Probability theory: a comprehensive course*, Springer Science & Business Media, 2013.

[Koltchinskii, 2001] Vladimir Koltchinskii, "Rademacher penalties and structural risk minimization," *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[Kuehne *et al.*, 2014] Hilde Kuehne, Ali Arslan, and Thomas Serre, "The language of actions: Recovering the syntax and semantics of goal-directed human activities," In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 780–787, 2014.

[Kuehne *et al.*, 2016] Hilde Kuehne, Juergen Gall, and Thomas Serre, "An end-to-end generative framework for video segmentation and recognition," In *IEEE Winter Conference on Applications of Computer Vision (WACV), 2016*, pages 1–8. IEEE, 2016.

[Kuznetsov and Mohri, 2014] Vitaly Kuznetsov and Mehryar Mohri, "Generalization Bounds for Time Series Prediction with Non-stationary Processes," In *Algorithmic Learning Theory (ALT)*, pages 260–274. Springer, 2014.

[Kuznetsov and Mohri, 2015] Vitaly Kuznetsov and Mehryar Mohri, "Learning Theory and Algorithms for Forecasting Non-Stationary Time Series," In *Conference on Neural Information Processing Systems (NIPS)*, pages 541–549, 2015.

[Kuznetsov and Mohri, 2016] Vitaly Kuznetsov and Mehryar Mohri, "Time Series Prediction and Online Learning," In *Workshop on Computational Learning Theory (COLT)*, pages 1190–1213, 2016.

[Langford and Shawe-Taylor, 2003] John Langford and John Shawe-Taylor, "PAC-Bayes & margins," In *Conference on Neural Information Processing Systems (NIPS)*, 2003.

[Linton and Sancetta, 2009] Oliver Linton and Alessio Sancetta, "Consistent estimation of a general nonparametric regression function in time series," *Journal of Econometrics*, 152(1):70–78, 2009.

[London *et al.*, 2013] Ben London, Bert Huang, Ben Taskar, and Lise Getoor, "Collective stability in structured prediction: Generalization from one example," In *International Conference on Machine Learning (ICML)*, pages 828–836, 2013.

[Mansour *et al.*, 2009] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," In *Workshop on Computational Learning Theory (COLT)*, 2009.

[Martinet, 1970] B. Martinet, "Régularisation d'inéquations variationnelles par approximations successives," *Rev. FranÃğaise Informat. Recherche Opérationnelle*, pages 154–158, 1970.

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

[McAllester, 1999] D.A. McAllester, "PAC-Bayesian model averaging," In *Proceedings of the twelfth annual conference on Computational Learning Theory (COLT)*, pages 164–170. ACM, 1999.

[McDonald *et al.*, 2012] Daniel J. McDonald, Cosma Rohilla Shalizi, and Mark Schervish, "Time series forecasting: model evaluation and selection using nonparametric risk bounds," *arXiv preprint arXiv:1212.0463*, 2012.

[Meir, 2000] Ron Meir, "Nonparametric time series prediction through adaptive model selection," *Machine Learning*, 39(1):5–34, 2000.

[Mohri and Medina, 2012] Mehryar Mohri and Andres Munoz Medina, "New analysis and algorithm for learning with drifting distributions," In *Algorithmic Learning Theory (ALT)*, pages 124–138. Springer, 2012.

[Mohri and Rostamizadeh, 2013] Mehryar Mohri and Afshin Rostamizadeh, "Stability Bounds for Stationary φ-mixing and β-mixing Processes," http://www.cs.nyu.edu/~mohri/pub/niidj.pdf, 2013, (Oct 10, corrected version of [JMLR (11), 2010]).

[Müller, 1997] Alfred Müller, "Integral probability metrics and their generating classes of functions," *Advances in Applied Probability*, pages 429–443, 1997.

[Norris, 1998] James R Norris, *Markov chains*, Cambridge University Press, 1998.

[Pentina and Lampert, 2014] A. Pentina and C.H. Lampert, "A PAC-Bayesian bound for Lifelong Learning," *International Conference on Machine Learning (ICML)*, 2014.

[Pentina and Lampert, 2017] Anastasia Pentina and Christoph H. Lampert, "Multi-task Learning with Labeled and Unlabeled Tasks," In *International Conference on Machine Learning (ICML)*, 2017.

[Pestov, 2010] Vladimir Pestov, "Predictive PAC learnability: A paradigm for learning from exchangeable input data.," In *IEEE International Conference on Granular Computing (GrC)*, pages 387–391, 2010.

[Rakhlin *et al.*, 2011] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari, "Online Learning: Stochastic, Constrained, and Smoothed Adversaries," In *Conference on Neural Information Processing Systems (NIPS)*, pages 1764–1772, 2011.

[Rakhlin *et al.*, 2014] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari, "Sequential complexities and uniform martingale laws of large numbers," *Probability Theory and Related Fields*, pages 1–43, 2014.

[Ralaivola *et al.*, 2010] Liva Ralaivola, Marie Szafranski, and Guillaume Stempfel, "Chromatic PAC-Bayes bounds for non-iid data: Applications to ranking and stationary β-mixing processes," *Journal of Machine Learning Research*, 11:1927–1956, 2010.

[Sauer, 1972] Norbert Sauer, "On the density of families of sets," *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

[Shalizi and Kontorovitch, 2013] Cosma Rohilla Shalizi and Aryeh Kontorovitch, "Predictive PAC Learning and Process Decompositions," In *Conference on Neural Information Processing Systems (NIPS)*, pages 1619–1627, 2013.

[Shelah, 1972] Saharon Shelah, "A combinatorial problem; stability and order for models and theories in infinitary languages," *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

[Steinwart, 2005] Ingo Steinwart, "Consistency of support vector machines and other regularized kernel classifiers," *IEEE Transactions on Information Theory*, 51(1):128–142, 2005.

[Tartakovsky *et al.*, 2014] Alexander Tartakovsky, Igor Nikiforov, and Michèle Basseville, *Sequential analysis: Hypothesis testing and changepoint detection*, CRC Press, 2014.

[Valiant, 1984] Leslie G Valiant, "A theory of the learnable," *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vapnik and Chervonenkis, 1971] Vladimir Vapnik and Alexey Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[Williams, 1991] David Williams, *Probability with martingales*, Cambridge University Press, 1991.

[Wintenberger, 2017] Olivier Wintenberger, "Optimal learning with Bernstein online aggregation," *Machine Learning*, 106(1):119–141, 2017.

[Yu, 1994] Bin Yu, "Rates of convergence for empirical processes of stationary mixing sequences," *Annals of Probability*, pages 94–116, 1994.

[Zolotarev, 1983] Vladimir Mikhailovich Zolotarev, "Probability metrics," *Teoriya Veroyatnostei i ee Primeneniya*, 28(2):264–287, 1983.

A Proofs from Chapter 3

A.1 Technical results regarding the convergence of martingales

In this section presenting a few technical results regarding uniform convergence of martingales. We start by introducing some additional notations. For a double sequence $u_{1:n}, u'_{1:n}$ of points in $\mathcal{Z}$, we define $\chi_t(\sigma)$ as u_t if $\sigma = 1$ and u'_t if $\sigma = -1$. Also define distributions $p_t(\sigma_{1:t-1}, u_{1:t-1}, u'_{1:t-1})$ over $\mathcal{Z}$ as a conditional distribution of z_t conditioned on history $\{z_1 = \chi_1(\sigma_1), \ldots, z_{t-1} = \chi_{t-1}(\sigma_{t-1})\}$. Then we can define a distribution ρ over two $\mathcal{Z}$-valued trees v and v' as follows: v_1 and v'_1 are sampled independently from the initial distribution of the process and for any path $\sigma_{1:n}$ for $2 \leq t \leq n$, $v_t(\sigma)$ and $v'_t(\sigma)$ are sampled independently from $p_t(\sigma_{1:t-1}, v_{1:t-1}(\sigma), v'_{1:t-1}(\sigma))$. Now we are ready to introduce a notion of *symmetrization*.

Definition A.1.1. *Let $z_{1:n}$ be a sample from a process and $z'_{1:n}$ its decoupled sequence. Let ξ be a random variable that is measurable with respect to Σ_n (a σ-algebra generated by $z_{1:n}$). Due to measurability, we know that there exists a measurable function ψ such that $\xi = \psi(z_1, \ldots, z_n)$. Then we define its symmetrized counterpart $\tilde{\xi} = \psi(\chi_1(\sigma_1), \ldots, \chi_n(\sigma_n))$, where $\sigma_{1:n}$ are i.i.d. Rademacher random variables and $\chi_{1:n}$ uses a double sequence $z_{1:n}, z'_{1:n}$.*

Lemma A.1.2. *Let $z_{1:n}$ be a sample from a process, $z'_{1:n}$ its decoupled sequence and $\mathcal{F} \subseteq \{f : \mathcal{Z} \to \mathbb{R}\}$. Let v and v' be two $\mathcal{Z}$-valued trees with the distribution ρ over them as described above. Then for any process $y_{1:n}$ such that each $y_t \sim \Sigma_{t-1}$, for any measurable*

functions $\varphi : \mathbb{R} \to \mathbb{R}$ *and* $\psi : \mathcal{Z}^n \to \mathbb{R}$*, we have*

$$\mathbb{E}\left[\varphi\left(\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n} y_t\left(f(z_t) - f(z'_t)\right)\right|\right)\psi(z_{1:n})\right] \tag{A.1}$$

$$= \mathbb{E}_{(v,v')\sim\rho}\mathbb{E}_{\sigma}\left[\varphi\left(\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n} \tilde{y}_t\sigma_t\left(f(v_t(\sigma)) - f(v'_t(\sigma))\right)\right|\right)\tilde{\psi}\right], \tag{A.2}$$

where $\tilde{\psi}$ *is a symmetrized version of* $\psi(z_{1:n})$ *and* $\sigma_{1:n}$ *are i.i.d. Rademacher random variables.*

Proof of Lemma A.1.2. The proof is direct extension of Theorem 3 from [Rakhlin *et al.*, 2011]. Let us denote $F = \varphi\left(\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n} y_t\left(f(z_t) - f(z'_t)\right)\right|\right)\psi(z_{1:n})$. We start by using the tower property of conditional expectations and the definition of a decoupled sequence.

$$\mathbb{E}[F] = \mathbb{E}_{z_1,z'_1\sim D_1}\mathbb{E}_{z_2,z'_2\sim D_2(z_1)}\ldots\mathbb{E}_{z_n,z'_n\sim D_n(z_{1:n-1})}[F], \tag{A.3}$$

where we denoted be $D_t(z_{1:t})$ a conditional distribution of the process at step t given $z_{1:t}$. Observe that we can we can rename z_1 and z'_1 in (A.3) to get

$$\mathbb{E}[F] = \mathbb{E}_{z_1,z'_1\sim D_1}\mathbb{E}_{z_2,z'_2\sim D_2(z'_1)}\ldots\mathbb{E}_{z_n,z'_n\sim D_n(z'_1,z_{2:n-1})}[F'], \tag{A.4}$$

where

$$F' = \varphi\left(\sup_{f\in\mathcal{F}}\left|-y_1(f(z_1) - f(z'_1)) + \sum_{t=2}^{n} y'_t\left(f(z_t) - f(z'_t)\right)\right|\right)\psi(z'_1, z_{2:n}) \tag{A.5}$$

with y'_t being the value of y_t as if the process has taken values $z'_1, z_2, \ldots, z_{t-1}$. Let us fix $\sigma \in \{\pm 1\}^n$ that will indicate whenever we swap z_t with z'_t or not. Using this notation and recalling the definition of χ_t, we get

$$\mathbb{E}[F] = \mathbb{E}_{z_1,z'_1\sim D_1}\mathbb{E}_{z_2,z'_2\sim D_2(\chi_1(1))}\ldots\mathbb{E}_{z_n,z'_n\sim D_n(\chi_1(1),\ldots,\chi_{n-1}(1))}[F] \tag{A.6}$$

$$= \mathbb{E}_{z_1,z'_1\sim D_1}\mathbb{E}_{z_2,z'_2\sim D_2(\chi_1(\sigma_1))}\ldots\mathbb{E}_{z_n,z'_n\sim D_n(\chi_1(\sigma_1),\ldots,\chi_{n-1}(\sigma_{n-1}))}[F^{\sigma}], \tag{A.7}$$

where in the second line we introduced

$$F^{\sigma} = \varphi\left(\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n} \sigma_t\tilde{y}_t\left(f(z_t) - f(z'_t)\right)\right|\right)\tilde{\psi}. \tag{A.8}$$

Since (A.7) holds for any fixed value σ, we can now consider σ to be an i.i.d. Rademacher random variables and get

$$\mathbb{E}[F] = \mathbb{E}_{\sigma}\mathbb{E}_{z_1,z'_1\sim D_1}\mathbb{E}_{z_2,z'_2\sim D_2(\chi_1(\sigma_1))}\ldots\mathbb{E}_{z_n,z'_n\sim D_n(\chi_1(\sigma_1),\ldots,\chi_{n-1}(\sigma_{n-1}))}[F^{\sigma}]. \tag{A.9}$$

The proof finishes by recalling the definition of a $\mathcal{Z}$-valued tree and the distribution ρ over the trees introduced at the beginning of the section. □

Lemma A.1.3. *Let $z_{1:n}$ be a sample, $\mathcal{F} \subseteq \{f : \mathcal{Z} \to [0,1]\}$ and $y_{1:n}$ be any process such that each $y_t \sim \Sigma_{t-1}$. Denote $E = \sum_{t=1}^n |y_t|$ and $V = \sum_{t=1}^n y_t^2$. Then for a fixed $\lambda, \beta > 0$ and $c = \ln 2\mathcal{S}_\infty(\mathcal{F}, \beta, n)$*

$$\mathbb{E}\left[e^{\lambda \sup_{f\in\mathcal{F}}\left|\sum_{t=1}^n y_t(f(z_t)-\mathbb{E}_{t-1}[f])\right| - \lambda^2 V - 2\lambda\beta E - c}\right] \leq 1. \tag{A.10}$$

Proof of Lemma A.1.3. Let $z'_{1:n}$ be a decoupled tangent sequence to $z_{1:n}$, i.e. a sequence that satisfies

$$\mathbb{E}_{t-1}\left[f(z_t)\right] = \mathbb{E}_{t-1}\left[f(z'_t)\right] = \mathbb{E}\left[f(z'_t)|\, z_{1:n}\right]. \tag{A.11}$$

Then

$$\mathbb{E}\left[e^{\lambda \sup_{f\in\mathcal{F}}\left|\sum_{t=1}^n y_t(f(z_t)-\mathbb{E}_{i-1}[f])\right| - \lambda^2 V - 2\lambda\beta E - c}\right] \leq \mathbb{E}\left[e^{\lambda \sup_{f\in\mathcal{F}}\left|\sum_{t=1}^n y_t(f(z_t)-f(z'_t))\right| - \lambda^2 V - 2\lambda\beta E - c}\right]. \tag{A.12}$$

The Lemma A.1.2 gives us that (A.12) equals to

$$\mathbb{E}_{(v,v')\sim\rho}\mathbb{E}_\sigma\left[e^{\lambda \sup_{f\in\mathcal{F}}\left|\sum_{t=1}^n \tilde{y}_t\sigma_t(f(v_t(\sigma))-f(v'_t(\sigma)))\right| - \lambda^2\tilde{V} - 2\lambda\beta\tilde{E} - c}\right] \tag{A.13}$$

$$\leq \mathbb{E}_{v\sim\rho}\mathbb{E}_\sigma\left[e^{2\lambda \sup_{f\in\mathcal{F}}\left|\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma))\right| - \lambda^2\tilde{V} - 2\lambda\beta\tilde{E} - c}\right], \tag{A.14}$$

where each $\tilde{y}_t$ is a symmetrized version of y_t, $\tilde{E} = \sum_{t=1}^n |\tilde{y}_t|$, $\tilde{V} = \sum_{t=1}^n \tilde{y}_t^2$ and we used Jensen inequality to get the second line. Now we take a sequential β-cover of $\mathcal{F}$ with respect to ℓ_∞-norm to get the following bound on (A.14)

$$\mathbb{E}_{v\sim\rho}\mathcal{S}_\infty(\mathcal{F},\beta,n)\mathbb{E}_\sigma\left[e^{2\lambda\left|\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma))\right| - \lambda^2\tilde{V} - c}\right] = \frac{1}{2}\mathbb{E}_{v\sim\rho}\mathbb{E}_\sigma\left[e^{2\lambda\left|\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma))\right| - \lambda^2\tilde{V}}\right]. \tag{A.15}$$

Introduce events $Y_+ = \{\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma)) \geq 0\}$ and $Y_- = \{\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma)) < 0\}$. Then the last line is equal to

$$\frac{1}{2}\mathbb{E}_{v\sim\rho}\mathbb{E}_\sigma\left[e^{2\lambda\left|\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma))\right| - \lambda^2\tilde{V}}\mathbb{I}\left[Y_+\right]\right] + \frac{1}{2}\mathbb{E}_{v\sim\rho}\mathbb{E}_\sigma\left[e^{2\lambda\left|\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma))\right| - \lambda^2\tilde{V}}\mathbb{I}\left[Y_-\right]\right] \tag{A.16}$$

$$\leq \frac{1}{2}\mathbb{E}_{v\sim\rho}\mathbb{E}_\sigma\left[e^{2\lambda\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma)) - \lambda^2\tilde{V}}\right] + \frac{1}{2}\mathbb{E}_{v\sim\rho}\mathbb{E}_\sigma\left[e^{-2\lambda\sum_{t=1}^n \tilde{y}_t\sigma_t f(v_t(\sigma)) - \lambda^2\tilde{V}}\right] \tag{A.17}$$

$$\leq 1, \tag{A.18}$$

where the last line follows by the standard martingale argument, since $\tilde{y}_t\sigma_t f(v_t(\sigma))$ is a martingale difference sequence (for a fixed tree v). □

In all of the proofs for MACRO, we use the following technical lemma about the meta-algorithm. The notations are introduced in Section 3.8.

Lemma A.1.4. *For any subroutine algorithm used by MACRO, for any $\alpha \in [0,1]$ and $\beta \in [0, \alpha/4]$, we have*

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}} |\frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} (\ell(h, z_t) - R(h, D_t))| > \alpha \wedge \mathcal{X}_{k,m}\right] \leq \frac{2k\mathcal{S}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{(\alpha - 4\beta)^2} e^{-\frac{1}{2}m(\alpha-4\beta)^2}. \tag{A.19}$$

Moreover, for any $\alpha \in [0,1]$ with $g_{j,i} = \ell(h_{j,i-1}, z_{t_{j,i}}) - R(h_{j,i-1}, D_{t_{j,i}})$

$$\mathbb{P}\left[|\frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} g_{I_n,i}| > \alpha \wedge \mathcal{X}_{k,m}\right] \leq \frac{2k}{\alpha^2} e^{-\frac{1}{2}m\alpha^2}.$$

Proof of Lemma A.1.4. Introduce events $A_i = \{I_n = i\}$ for $i = 1, \ldots, k$ and $B_{i,j} = \{s_{j,n} = i\}$ (we suppress the dependence on n to increase readability). Observe that $\mathcal{X}_{k,m} = \{\cup_{i\geq 1} A_i\} \wedge \{\cup_{i\geq m} B_{i,I_n}\}$. Denoting $\Lambda(j) = \sup_{h\in\mathcal{H}} |\frac{1}{s_{j,n}} \sum_{t\in C_{j,n}} (\ell(h, z_t) - R(h, D_t))|$, we have

$$\mathbb{P}\left[\Lambda(I_n) > \alpha \wedge \mathcal{X}_{k,m}\right] \leq \sum_{j\in\mathrm{supp}(I_n)} \mathbb{P}\left[\Lambda(j) > \alpha \wedge \{\cup_{i\geq m} B_{i,j}\}\right]. \tag{A.20}$$

Each of the last probabilities can be bounded using a union bound:

$$\mathbb{P}\left[\Lambda(j) > \alpha \wedge \{\cup_{i\geq m} B_{i,j}\}\right] \leq \sum_{i\geq m} \mathbb{P}\left[\Lambda(j) > \alpha \wedge B_{i,j}\right]. \tag{A.21}$$

Now observe that a sequence $w_t = \frac{1}{i}\mathbb{I}\left[t \in C_{j,n}\right]$ is adapted is Σ_{t-1} and we can also re-write $\Lambda(j)$ on the event $B_{i,j}$:

$$\sup_{h\in\mathcal{H}} |\frac{1}{s_{j,n}} \sum_{t\in C_{j,n}} (\ell(h, z_t) - R(h, D_t))| = \sup_{h\in\mathcal{H}} |\sum_{t=1}^{n} w_t(\ell(h, z_t) - R(h, D_t))| \tag{A.22}$$

Let us introduce $E = \sum_{t=1}^n w_t$ and $V = \sum_{t=1}^n w_t^2$, then Lemma A.1.3, tells us for $\lambda, \beta > 0$

$$\mathbb{E}\left[e^{\lambda\Lambda(j) - \lambda^2 V - 2\lambda\beta E - \ln 2\mathcal{S}_\infty(\mathcal{L}(\mathcal{H}),\beta,n)}\right] \leq 1. \tag{A.23}$$

That translates into

$$\mathbb{P}\left[\Lambda(j) > \alpha \wedge B_{i,j}\right] \leq 2\mathcal{S}_\infty(\mathcal{L}(\mathcal{H}), \beta, n) e^{-\frac{1}{2}i(\alpha-4\beta)^2}. \tag{A.24}$$

Summing the probabilities, we obtain the first statement of the lemma.

For the second part of the lemma, denote $\Lambda(j) = |\frac{1}{s_{j,n}} \sum_{i=1}^{s_{j,n}} g_{j,i}|$ and, after applying the same decomposition as above, we end up bounding $\mathbb{P}\left[\Lambda(j) > \alpha \wedge B_{i,j}\right]$. Observe that each $h_{j,i}$ is adapted to the filtration generated by $\{z_{t_{j,i}}\}_{i=1}^{\infty}$, hence, $g_{j,i}$ behaves like a martingale difference sequence. However, there is a technical difficulty in the fact that the indices $t_{j,i}$ are, in fact, stopping times. To get around it, observe that we can write $\Lambda(j)$ as a sum over all the

data with the adapted weights. Set $w_{j,t}$ to 1 if we updated the algorithm j at step t and to 0 otherwise. Correspondingly, define $\bar{h}_{j,t}$ as the last chosen hypothesis by the j-th algorithm up to step t. This way, both $w_{j,t}$ and $\bar{h}_{j,t}$ depend only on $z_{1:t-1}$. Then

$$\sum_{i=1}^{s_{j,n}} g_{j,i} = \sum_{i=1}^{s_{j,n}} (\ell(h_{j,i-1}, z_{t_{j,i}}) - R(h_{j,i-1}, D_{t_{j,i}})) = \sum_{t=1}^{n} w_{j,t} (\ell(\bar{h}_{j,t}, z_t) - R(\bar{h}_{j,t}, D_t)). \tag{A.25}$$

At this point we can again use Lemma A.1.3 and get the second statement of the lemma. □

The next result is an analogue of Toeplitz lemma for the definition of convergent double array from Section 3.7.

Lemma A.1.5. *If double array $d_{t,n}$ is convergent in a sense of Definition 3.7.1, then $\frac{1}{n}\sum_{t=1}^{n} d_{t,n+1}$ converges to 0 in probability.*

Proof of Lemma A.1.5. The proof is similar to that of the Toeplitz lemma, but adapted to our notion of convergence. Fix $\varepsilon > 0$ and $\delta > 0$. Then, by the definition of a convergent array, for $\varepsilon' = \delta' = \frac{\delta\varepsilon}{4}$

$$\exists n_0, \exists t_0 : 0 \leq t_0 < n_0, \forall n \geq n_0, \forall t_0 \leq t < n : \mathbb{P}[d_{t,n} > \varepsilon'] \leq \delta'. \tag{A.26}$$

In particular, this means that for any $n \geq n_0$ and $\forall t_0 \leq t < n$ we have $\mathbb{E}[d_{t,n+1}] \leq \varepsilon' + \delta' = \frac{\delta\varepsilon}{2}$, because of the boundedness of $d_{t,n}$.

Now, choose any $n_1 \geq n_0$ that satisfies $\frac{n_0}{n_1} \leq \frac{\varepsilon}{2}$. Then for any $n \geq n_1$ we get

$$\mathbb{P}\left[\frac{1}{n}\sum_{t=1}^{n} d_{t,n+1} > \varepsilon\right] \leq \mathbb{P}\left[\frac{1}{n}\sum_{t=n_0+1}^{n} d_{t,n+1} > \frac{\varepsilon}{2}\right] \tag{A.27}$$

$$\leq 2\frac{\sum_{t=n_0+1}^{n} \mathbb{E}[d_{t,n+1}]}{n\varepsilon} \tag{A.28}$$

$$\leq \delta, \tag{A.29}$$

where the last line follows from the bound on the expectations. □

The MACRO with online subroutine in the non-convex case relies on the analogue of the online-to-batch conversion from [Cesa-Bianchi *et al.*, 2004]. The following lemma is a version of Lemma 3 from [Cesa-Bianchi *et al.*, 2004] proved for the case of dependent data and the conditional risk.

Lemma A.1.6. *For the setting of Theorem 3.8.10, let*

$$v(j,i) = R(h_{j,i}, D_{n+1}) + 2c_{I_n,\delta}(s_{I_n,n} - i). \tag{A.30}$$

Then we have

$$\mathbb{P}\left[R(h_n, D_{n+1}) > \min_{1\le i\le s_{I_n,n}} v(I_n, i) + 2\varepsilon \wedge \mathcal{X}_{k,m}\right] \le \frac{k\delta}{m}. \tag{A.31}$$

Proof. Introduce events $A_r = \{|\mathrm{supp}(I_n)| \le k \wedge s_{I_n,n} = r\}$. Using a union bound, we have

$$\mathbb{P}\left[R(h_n, D_{n+1}) > \min_{1\le i\le s_{I_n,n}} v(I_n, i) + 2\varepsilon \wedge \mathcal{X}_{k,m}\right] \tag{A.32}$$

$$\le \sum_{r\ge m} \mathbb{P}\left[R(h_n, D_{n+1}) > \min_{1\le i\le s_{I_n,n}} v(I_n, i) + 2\varepsilon \wedge A_r\right]. \tag{A.33}$$

Now we will focus on the last probabilities for each r. Let $J_n^\star = \mathrm{argmin}_{1\le i\le s_{I_n,n}} v(I_n, i)$ and also introduce events $B_i = \{u_n(I_n, i) \le u_n(I_n, J_n^\star)\}$. Then, since $u_n(I_n, J_n) \le u_n(I_n, J_n^\star)$ is always true by the definition of J_n, we get

$$\mathbb{P}\left[R(h_n, D_{n+1}) > \min_{1\le i\le s_{I_n,n}} v(I_n, i) + 2\varepsilon \wedge A_r\right] \tag{A.34}$$

$$\le \sum_{i=1}^{r} \mathbb{P}\left[R(h_{I_n,i}, D_{n+1}) > v(I_n, J_n^\star) + 2\varepsilon \wedge B_i \wedge A_r\right]. \tag{A.35}$$

Observe that if B_i is true, then at least one of the following events is also true.

$$\mathcal{T}_{1,i} = \{u_n(I_n, i) \le R(h_{I_n,i}, D_{n+1}) - 2\varepsilon\}, \tag{A.36}$$

$$\mathcal{T}_{2,i} = \{R(h_{I_n,i}, D_{n+1}) < v(I_n, J_n^\star) + 2\varepsilon\}, \tag{A.37}$$

$$\mathcal{T}_3 = \{u_n(I_n, J_n^\star) > v(I_n, J_n^\star)\}. \tag{A.38}$$

From this we get

$$\mathbb{P}\left[R(h_{I_n,i}, D_{n+1}) > v(I_n, J_n^\star) + 2\varepsilon \wedge B_i \wedge A_r\right] \tag{A.39}$$

$$\le \mathbb{P}\left[R(h_{I_n,i}, D_{n+1}) > v(I_n, J_n^\star) + 2\varepsilon \wedge \mathcal{T}_{1,i} \wedge A_r\right] \tag{A.40}$$

$$+ \mathbb{P}\left[R(h_{I_n,i}, D_{n+1}) > v(I_n, J_n^\star) + 2\varepsilon \wedge \mathcal{T}_{2,i} \wedge A_r\right] \tag{A.41}$$

$$+ \mathbb{P}\left[R(h_{I_n,i}, D_{n+1}) > v(I_n, J_n^\star) + 2\varepsilon \wedge \mathcal{T}_3 \wedge A_r\right]. \tag{A.42}$$

First, notice that

$$\mathbb{P}\left[R(h_{I_n,i}, D_{n+1}) > v(I_n, J_n^\star) + 2\varepsilon \wedge \mathcal{T}_{2,i} \wedge A_r\right] = 0. \tag{A.43}$$

Moreover, since

$$\left| R(h_{I_n,i}, D_{n+1}) - \frac{1}{s_{I_n,n} - i} \sum_{s=i+1}^{s_{I_n,n}} R(h_{I_n,i}, D_{t_{I_n,s}}) \right| \leq 2\varepsilon, \tag{A.44}$$

we have

$$\begin{aligned}
\mathbb{P}\left[\mathcal{T}_{1,i} \wedge A_r\right] &= \mathbb{P}\left[u_n(I_n, i) \leq R(h_{I_n,i}, D_{n+1}) - 2\varepsilon \wedge A_r\right] && \text{(A.45)}\\
&\leq \mathbb{P}\left[u_n(I_n, i) \leq \frac{1}{s_{I_n,n} - i} \sum_{s=i+1}^{s_{I_n,n}} R(h_{I_n,i}, D_{t_{I_n,s}}) \wedge A_r\right] && \text{(A.46)}\\
&\leq \sum_{j \in \mathrm{supp}(I_n)} \mathbb{P}\left[u_n(j, i) \leq \frac{1}{s_{j,n} - i} \sum_{s=i+1}^{s_{j,n}} R(h_{j,i}, D_{t_{j,s}}) \wedge A_r\right]. && \text{(A.47)}
\end{aligned}$$

From Lemma A.1.4 we get that

$$\mathbb{P}\left[u_n(j, i) \leq \frac{1}{s_{j,n} - i} \sum_{s=i+1}^{s_{j,n}} R(h_{j,i}, D_{t_{j,s}}) \wedge A_r\right] \leq \frac{\delta}{r^3(r+1)}. \tag{A.48}$$

And, hence,

$$\mathbb{P}\left[\mathcal{T}_{1,i} \wedge A_r\right] \leq \frac{k\delta}{r^3(r+1)}. \tag{A.49}$$

Similarly, from Lemma A.1.4,

$$\mathbb{P}\left[\mathcal{T}_3 \wedge A_r\right] \leq \frac{k\delta}{r^2(r+1)}. \tag{A.50}$$

Combining these two together, we get

$$\mathbb{P}\left[R(h_n, D_{n+1}) > \min_{1 \leq i \leq s_{I_n,n}} v(I_n, i) + 2\varepsilon \wedge A_r\right] \leq \frac{k\delta}{r^2}, \tag{A.51}$$

which gives us the statement on the lemma. □

A.2 Proof of Theorem 3.5.1

Using the fact that the minimizer of $\mathbb{E}_n\left[(h - \mathbf{z}_{n+1})^2\right]$ is $\mathbb{E}_n\mathbf{z}_{n+1}$, we can rewrite for any h_n measurable from $z_{1:n}$:

$$\begin{aligned}
R(h_n, D_{n+1}) - \inf_{h \in \mathcal{H}} R(h, D_{n+1}) &= \mathbb{E}_n\left[(h_n - \mathbf{z}_{n+1})^2\right] - \inf_{h \in \mathcal{H}} \mathbb{E}_n\left[(h - \mathbf{z}_{n+1})^2\right] && \text{(A.52)}\\
&= \mathbb{E}_n\left[(h_n - \mathbf{z}_{n+1})^2\right] - \mathbb{E}_n\left[(\mathbb{E}_n\mathbf{z}_{n+1} - \mathbf{z}_{n+1})^2\right] && \text{(A.53)}\\
&= (h_n - \mathbb{E}_n\mathbf{z}_{n+1})^2. && \text{(A.54)}
\end{aligned}$$

A minor modification of the proof of Theorem 1 of [Gyorfi *et al.*, 1998] gives that for every algorithm that produces a sequence h_n of hypotheses, there is a stationary and ergodic process such that

$$\mathbb{P}\left[\limsup_{n\to\infty} (h_n - \mathbb{E}_n \mathbf{z}_{n+1})^2 > \frac{1}{16}\right] \geq \frac{1}{8}, \tag{A.55}$$

which shows that no algorithm can be a limit learner for the class of all stationary and ergodic binary processes.

A.3 Proof of Theorem 3.7.2

Recall the upper bounds discussed in Section 3.8.1.

$$R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) \leq 2 \sup_{h\in\mathcal{H}} \left| \frac{1}{n}\sum_{t=1}^{n} \ell(h, z_t) - R(h, D_{n+1}) \right| \tag{A.56}$$

$$\leq 2\mathcal{V}_n(\mathcal{L}(\mathcal{H}), u) + \frac{2}{n}\sum_{t=1}^{n} d_{t,n+1}, \tag{A.57}$$

where $u \in \mathbb{R}^n$ is a vector of uniform weights, i.e. $(\frac{1}{n}, \ldots, \frac{1}{n})$. The convergence in probability of $\mathcal{V}_n(\mathcal{L}(\mathcal{H}), u)$ is guaranteed by Theorem 2.5.5 for any stochastic process. The convergence of $\frac{1}{n}\sum_{t=1}^{n} d_{t,n+1}$ follows from the definition of the convergent discrepancies and is a content of Lemma A.1.5.

A.4 Proof of Lemma 3.7.4

The proof follows from the following bound

$$d_{t,n} = \sup_{f\in\mathcal{L}(\mathcal{H})} |\mathbb{E}_t f - \mathbb{E}_n[\mathbf{x}_f]| \leq \mathbb{E}_n\left[\sup_{f\in\mathcal{L}(\mathcal{H})} |\mathbb{E}_t f - \mathbf{x}_f|\right]. \tag{A.58}$$

Then the convergence of the discrepancies follows from the Definition 3.7.3 of the uniformly convergent martingale.

A.5 Proof of Theorem 3.8.4

First, recall the definition of an M-bound and introduce the following explicit notation for the weights:

$$w_t(r) = \frac{g_n(\Psi_t(r))}{\sum_{j=1}^{n} g_n(\Psi_j(r))}. \tag{A.59}$$

Using this notation, the weighted ERM uses the weights $w_t(J_{n+1})$, see equation (3.26). The starting point of the proof is the decomposition discussed in Section 3.8.1.

$$R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) \leq 2\sup_{h\in\mathcal{H}} \left|\sum_{t=1}^{n} w_t(J_{n+1})\ell(h, z_t) - R(h, D_{n+1})\right| \tag{A.60}$$

$$\leq 2\sup_{h\in\mathcal{H}} \left|\sum_{t=1}^{n} w_t(J_{n+1})(\ell(h, z_t) - R(h, D_t))\right| \tag{A.61}$$

$$+ 2\sum_{t=1}^{n} w_t(J_{n+1}) d_{t,n+1} \tag{A.62}$$

$$\leq 2\Theta(\mathcal{L}(\mathcal{H}), J_{n+1}) + 2\Lambda_n, \tag{A.63}$$

where at the last line we introduced

$$\Theta(\mathcal{F}, r) = \sup_{f\in\mathcal{F}} \left|\sum_{t=1}^{n} w_t(r)\left(f(z_t) - \mathbb{E}_{t-1}[f]\right)\right|. \tag{A.64}$$

for some function class $\mathcal{F} \subseteq \{f : \mathcal{Z} \to [0,1]\}$ and an integer r. We are left to prove the high probability bound for $\Theta(\mathcal{F}, J_{n+1})$. To this end, let us define events $A_r = \{J_{n+1} = r\}$ and $B_i(r) = \{i \leq \sum_{t=1}^{n} g_n(\Psi_t(r)) \leq i+1\}$, so that the exceptional set decomposes as $\mathcal{E}_{k,m} = \{\cup_{r\leq k} A_r\} \cap \{\cup_{i\geq m} B_i(J_{n+1})\}$. Then we have

$$\mathbb{P}\left[\Theta(\mathcal{F}, J_{n+1}) \geq \alpha\right] \leq \mathbb{P}\left[\Theta(\mathcal{F}, J_{n+1}) \geq \alpha \wedge \mathcal{E}_{k,m}\right] + \mathbb{P}\left[\mathcal{E}_{k,m}^c\right]. \tag{A.65}$$

Now we can take a union bound for the first summand over A_r's and get

$$\mathbb{P}\left[\Theta(\mathcal{F}, J_{n+1}) \geq \alpha \wedge \mathcal{E}_{k,m}\right] \leq \sum_{r=1}^{k} \mathbb{P}\left[\Theta(\mathcal{F}, r) \geq \alpha \wedge \{\cup_{i\geq m} B_i(r)\}\right]. \tag{A.66}$$

Taking another union bound for each r, we end up with

$$\mathbb{P}\left[\Theta(\mathcal{F}, r) \geq \alpha \wedge \{\cup_{i\geq m} B_i(r)\}\right] \leq \sum_{i\geq m} \mathbb{P}\left[\Theta(\mathcal{F}, r) \geq \alpha \wedge B_i(r)\right]. \tag{A.67}$$

Now we study the last probability for a fixed r and i. On $B_i(r)$ we can lower bound the denominator of the weights: $\sum_{t=1}^{n} g_n(\Psi_t(r)) \geq i$, leading to

$$\Theta(\mathcal{F}, r) \leq \Theta_i(\mathcal{F}, r) = \frac{1}{i}\sup_{f\in\mathcal{F}} \left|\sum_{t=1}^{n} g_n(\Psi_t(r))\left(f(z_t) - \mathbb{E}_{t-1}[f]\right)\right|. \tag{A.68}$$

Let $\lambda > 0$ and denote $E = \frac{1}{i}\sum_{t=1}^{n} g_n(\Psi_t(r))$ and $V = \frac{1}{i^2}\sum_{t=1}^{n} g_n^2(\Psi_t(r))$. Then, since $\frac{1}{i} g_n(\Psi_t(r)) \sim \Sigma_{t-1}$ by the definition of an M-bound, Lemma A.1.3 gives us

$$\mathbb{E}\left[e^{\lambda\Theta_i(\mathcal{F},r) - \lambda^2 V - 2\lambda\beta E - \ln 2\mathcal{S}_\infty(\mathcal{F},\beta,n)}\right] \leq 1. \tag{A.69}$$

Let $C = \{\Theta_i(\mathcal{F}, r) \geq \alpha \wedge B_i(r)\}$ and note that $E \leq \frac{i+1}{i} \leq 2$ and $V \leq \frac{i+1}{i^2} \leq \frac{2}{i}$ on $B_i(r)$ by the boundedness of g_n. Then we have the following chain of inequalities

$$1 \geq \mathbb{E}\left[e^{\lambda\Theta_i(\mathcal{F},r)-\lambda^2 V-2\lambda\beta E-\ln 2\mathcal{S}_\infty(\mathcal{F},\beta,n)}\right] \tag{A.70}$$

$$\geq \mathbb{E}\left[e^{\lambda\Theta_i(\mathcal{F},r)-\lambda^2 V-2\lambda\beta E-\ln 2\mathcal{S}_\infty(\mathcal{F},\beta,n)}\mathbb{I}\left[C\right]\right] \tag{A.71}$$

$$\geq e^{\lambda\alpha-\lambda^2\frac{2}{i}-4\lambda\beta-\ln 2\mathcal{S}_\infty(\mathcal{F},\beta,n)}\mathbb{P}\left[C\right]. \tag{A.72}$$

Hence, by optimizing over λ, we get

$$\mathbb{P}\left[\Theta(\mathcal{F}, r) \geq \alpha \wedge B_i(r)\right] \leq 2\mathcal{S}_\infty(\mathcal{F}, \beta, n)e^{-\frac{1}{2}i(\alpha-4\beta)^2}. \tag{A.73}$$

Now, coming back to (A.67), we can evaluate it by computing the sum to obtain

$$\mathbb{P}\left[\Theta(\mathcal{F}, J_{n+1}) \geq \alpha \wedge \mathcal{E}_{k,m}\right] \leq \frac{2k\mathcal{S}_\infty(\mathcal{F}, \beta, n)}{(\alpha-4\beta)^2}e^{-\frac{1}{2}m(\alpha-4\beta)^2}. \tag{A.74}$$

And this finishes the proof.

A.6 Proof of Lemma 3.8.6

The lower bound comes from the fact that MACRO constructs an ε-covering.

For the upper bound, observe that a new subroutine is started if and only if its associated conditional distribution differs by more than ε from the ones of all previously created subroutines. Therefore, the set of conditional distribution associated with subroutines form an *ε-separated set* with respect to $M_{i,j}$'s (no two elements are closer than ε to each other). The maximal size of such a set is at most the *covering number* of half the distance.

A.7 Proof of Theorem 3.8.8

We start by the usual argument for the empirical risk minimization that allows us to focus on the uniform deviations.

$$R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) \leq 2\sup_{h\in\mathcal{H}} |R(h, D_{n+1}) - \frac{1}{s_{I_n,n}}\sum_{t\in C_{I_n,n}} \ell(h, z_t)|. \tag{A.75}$$

Now we can upper bound the last term.

$$\sup_{h\in\mathcal{H}} |R_n(h) - \frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} \ell(h, z_t)| \tag{A.76}$$

$$\leq \sup_{h\in\mathcal{H}} |R_n(h) - \frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} R(h, D_t)| \tag{A.77}$$

$$+ \sup_{h\in\mathcal{H}} |\frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} R(h, D_t) - \frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} \ell(h, z_t)| \tag{A.78}$$

$$\leq \frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} d_{t,n+1} + \sup_{h\in\mathcal{H}} |\frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} (\ell(h, z_t) - R(h, D_t))| \tag{A.79}$$

$$\leq \frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} M_{t,n+1} + \sup_{h\in\mathcal{H}} |\frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} (\ell(h, z_t) - R(h, D_t))| \tag{A.80}$$

$$\leq 2\varepsilon + \sup_{h\in\mathcal{H}} |\frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} (\ell(h, z_t) - R(h, D_t))|, \tag{A.81}$$

where the last bound follows from the way the meta-algorithm chooses I_n. Hence, we get

$$\mathbb{P}\left[R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) > \alpha + 2\varepsilon\right] \leq \mathbb{P}\left[\sup_{h\in\mathcal{H}} |\frac{1}{s_{I_n,n}} \sum_{t\in C_{I_n,n}} (\ell(h, z_t) - R(h, D_t))| > \alpha\right]. \tag{A.82}$$

The last probability can be bounded using Lemma A.1.4 giving us the statement of the theorem.

A.8 Proof of Theorem 3.8.9

Note that by the way I_n is chosen, we get for any $h \in \mathcal{H}$ that

$$R(h, D_{n+1}) - R(h, D_{t_{I_n,i}}) \leq 2\varepsilon. \tag{A.83}$$

Therefore, by using the convexity of the loss,

$$R(h_n, D_{n+1}) \leq \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h_{I_n,i}, D_{n+1}) \leq \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h_{I_n,i}, D_{t_{I_n,i}}) + 2\varepsilon. \tag{A.84}$$

Similarly, for any fixed h

$$R(h, D_{n+1}) \geq \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h, D_{t_{I_n,i}}) - 2\varepsilon. \tag{A.85}$$

Therefore,

$$R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) \leq 4\varepsilon + \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h_{I_n,i}, D_{t_{I_n,i}}) - \inf_{h\in\mathcal{H}} \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h, D_{t_{I_n,i}}). \tag{A.86}$$

We split the last difference into the following three terms and deal with them separately.

$$T_1 = \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} (R(h_{I_n,i}, D_{t_{I_n,i}}) - \ell(h_{I_n,i}, z_{t_{I_n,i}})) \tag{A.87}$$

$$T_2 = \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} \ell(h_{I_n,i}, z_{t_{I_n,i}}) - \inf_{h\in\mathcal{H}} \sum_{i=1}^{s_{I_n,n}} \ell(h, z_{t_{I_n,i}}) \tag{A.88}$$

$$T_3 = \inf_{h\in\mathcal{H}} \sum_{i=1}^{s_{I_n,n}} \ell(h, z_{t_{I_n,i}}) - \inf_{h\in\mathcal{H}} \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h, D_{t_{I_n,i}}). \tag{A.89}$$

T_2 is in fact just $W_{I_n,n}$. For T_3 observe that

$$\inf_{h\in\mathcal{H}} \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h, D_{t_{I_n,i}}) \geq \inf_{h\in\mathcal{H}} \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} \ell(h, z_{t_{I_n,i}}) \tag{A.90}$$

$$+ \inf_{h\in\mathcal{H}} (\frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} (R(h, D_{t_{I_n,i}}) - \ell(h, z_{t_{I_n,i}})). \tag{A.91}$$

Therefore, T_3 is bounded by $\tilde{T}_3$:

$$\tilde{T}_3 = \sup_{h\in\mathcal{H}} (\frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} (\ell(h, z_{t_{I_n,i}}) - R(h, D_{t_{I_n,i}})). \tag{A.92}$$

Combining everything together,

$$\mathbb{P}\left[R(h_n, D_{n+1}) - \inf_h R(h, D_{n+1}) > \alpha + 4\varepsilon + W_{I_n,n}\right] \tag{A.93}$$

$$\leq \mathbb{P}\left[T_1 + \tilde{T}_3 > \alpha \wedge \mathcal{X}_{k,m}\right] + \mathbb{P}\left[\mathcal{X}_{k,m}^c\right] \tag{A.94}$$

$$\leq \mathbb{P}\left[T_1 > \alpha/2 \wedge \mathcal{X}_{k,m}\right] + \mathbb{P}\left[\tilde{T}_3 > \alpha/2 \wedge \mathcal{X}_{k,m}\right] + \mathbb{P}\left[\mathcal{X}_{k,m}^c\right]. \tag{A.95}$$

The terms involving T_1 and $\tilde{T}_3$ can be bounded using Lemma A.1.4 giving us the statement of the theorem.

A.9 Proof of Theorem 3.8.10

From Lemma A.1.6 we get that with probability at least $1 - \frac{k\delta}{m}$ on $\mathcal{X}_{k,m}$

$$R(h_n, D_{n+1}) \leq \min_{1\leq i\leq s_{I_n,n}} v(I_n, i) + 2\varepsilon. \tag{A.96}$$

Hence, we focus on bounding $\min_{1\leq i\leq s_{I_n,n}} v(I_n, i)$. Observe that

$$\min_{1\leq i\leq s_{I_n,n}} v(I_n, i) \leq \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} (R(h_{I_n,i}, D_{n+1}) + c_{I_n,\delta/2}(s_{I_n,n} - i)) \tag{A.97}$$

$$\leq \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} (R(h_{I_n,i}, D_{t_{I_n,i}}) + 2\varepsilon + c_{I_n,\delta/2}(s_{I_n,n} - i)) \tag{A.98}$$

$$\leq \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h_{I_n,i}, t_{I_n,i}) + 2\varepsilon + 2\sqrt{\frac{1}{s_{I_n,n}} \log \frac{s_{I_n,n}^3(s_{I_n,n}+1)}{\delta}}. \tag{A.99}$$

Similarly to Lemma A.1.4, we have that with probability at least $1 - \frac{k\delta}{m}$ on $\mathcal{X}_{k,m}$:

$$\frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} R(h_{I_n,i}, D_{t_{I_n,i}}) \leq \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} \ell(h_{I_n,i}, z_{t_{I_n,i}}) + \sqrt{\frac{1}{s_{I_n,n}} \log \frac{s^2_{I_n,n}}{\delta}}. \tag{A.100}$$

As in the proof of Theorem 3.8.9, we get with probability at least $1 - \frac{k\delta}{m}$ on $\mathcal{X}_{k,m}$:

$$-\inf_{h\in\mathcal{H}} R(h, D_{n+1}) \leq -\inf_{h} \frac{1}{s_{I_n,n}} \sum_{i=1}^{s_{I_n,n}} \ell(h, z_{t_{I_n,i}}) + \sqrt{\frac{1}{s_{I_n,n}} \log \frac{s^2_{I_n,n} \mathcal{S}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{\delta}} + 4\beta. \tag{A.101}$$

Therefore, we can conclude that

$$\mathbb{P}\left[R(h_n, D_{n+1}) - \inf_{h\in\mathcal{H}} R(h, D_{n+1}) > 2\varepsilon + W_{I_n,n} + U_\delta(I_n, \beta) \wedge \mathcal{X}_{k,m}\right] \leq \frac{3k\delta}{m}. \tag{A.102}$$

A.10 Examples from Sections 3.8.4 and 3.8.5

Markov chains. First, we bound the probability of A_k using a union bound:

$$\mathbb{P}[A_k] = \mathbb{P}[J_{n+1} > k] \leq \mathbb{P}[F_{z_n} > k] \leq |S| \max_s \mathbb{P}[F_s > k]. \tag{A.103}$$

On the event $B_{k,m}$ we have the following chain of inequalities.

$$\sum_{t=J_n}^{n} \mathbb{I}[d_{t,J_{n+1}} \leq \varepsilon_n] \geq \sum_{t=k}^{n} \mathbb{I}[d_{t,J_{n+1}} \leq \varepsilon_n] \geq \sum_{t=k}^{n} \mathbb{I}[d_{t,J_{n+1}} = 0] \geq \sum_{t=k}^{n} \mathbb{I}[z_t = z_{J_{n+1}}], \tag{A.104}$$

which gives us

$$\begin{aligned}
\mathbb{P}[B_{k,m}] &= \mathbb{P}\left[J_n \leq k \wedge \sum_{t=J_{n+1}}^{n} \mathbb{I}[d_{t,J_{n+1}} \leq \varepsilon_n] < m\right] && \text{(A.105)}\\
&\leq \mathbb{P}\left[J_n \leq k \wedge \sum_{t=k}^{n} \mathbb{I}[z_t = z_{J_{n+1}}] < m\right] && \text{(A.106)}\\
&\leq |S| \max_s \mathbb{P}\left[J_{n+1} \leq k \wedge \sum_{t=k}^{n} \mathbb{I}[z_t = s] < m \wedge z_{J_{n+1}} = s\right]. && \text{(A.107)}
\end{aligned}$$

Now, for a given state s, $\sum_{t=k}^{n} \mathbb{I}[z_t = s]$ can be lower bounded by the number of times we hit the state s again. Let T_s^i, $i \geq 1$, be independent copies of the recurrence times. Then $\sum_{t=k}^{n} \mathbb{I}[z_t = s] \geq m$ for any $m \geq 0$, such that $\sum_{i=1}^{m} T_s^i \leq n - k$. We also have the following sequence of inclusions.

$$\begin{aligned}
&\left\{1 \leq i \leq m : T_s^i \leq \lfloor \frac{n-k}{m} \rfloor \wedge J_{n+1} \leq k \wedge z_{J_{n+1}} = s\right\} && \text{(A.108)}\\
&\subseteq \left\{\sum_{i=1}^{m} T_s^i \leq n - k \wedge J_{n+1} \leq k \wedge z_{J_{n+1}} = s\right\} && \text{(A.109)}\\
&\subseteq \left\{\sum_{t=k}^{n} \mathbb{I}[z_t = s] \geq m \wedge J_{n+1} \leq k \wedge z_{J_{n+1}} = s\right\}. && \text{(A.110)}
\end{aligned}$$

And this gives us

$$\mathbb{P}\left[J_{n+1} \leq k \wedge \sum_{t=k}^{n} \mathbb{I}\left[z_t = s\right] < m \wedge z_{J_{n+1}} = s\right] \tag{A.111}$$

$$\leq \mathbb{P}\left[\exists\, 1 \leq i \leq m : T_s^i > \lfloor \frac{n-k}{m} \rfloor\right] \tag{A.112}$$

$$\leq m\mathbb{P}\left[T_s > \lfloor \frac{n-k}{m} \rfloor\right]. \tag{A.113}$$

Dynamical systems. The bound on $\mathbb{P}\left[A_k\right]$ follows from the fact that $J_{n+1} \leq F(C_n)$. For the $B_{k,m}$ we get

$$\mathbb{P}\left[B_{k,m}\right] \leq k \max_{1 \leq j \leq k} \mathbb{P}\left[J_n = j \wedge \sum_{t=j}^{n} \mathbb{I}\left[d_{t,j} \leq b_n\right]\right]. \tag{A.114}$$

And similarly to the Markov chain example, we can obtain for each fixed j

$$\mathbb{P}\left[J_n = j \wedge \sum_{t=j}^{n} \mathbb{I}\left[d_{t,j} \leq \varepsilon_n\right]\right] \leq \mathbb{P}\left[T(C_j) > \lfloor \frac{n-j}{m} \rfloor\right]. \tag{A.115}$$

General stationary processes. The bound for this case is done analogously to the previous two examples.

B Proofs from Chapter 5

B.1 Technical results for MTLAB

General results for KL-divergence

Lemma B.1.1 (e.g. [Catoni, 2004], 5.2.1). *For any function $f : \mathcal{H} \to \mathbb{R}$ and any distribution P over $\mathcal{H}$, we have*

$$\sup_Q \{\mathbb{E}_{h\sim Q}[f(h)] - KL(Q|P)\} = \log \mathbb{E}_{h\sim P}\left[e^{f(h)}\right]. \tag{B.1}$$

Definition B.1.2. *Given a set Δ of distributions and an arbitrary distribution P, we define*

$$\Pi_\Delta(P) = \operatorname*{argmin}_{Q\in\Delta} KL(Q|P) \tag{B.2}$$

as a KL-projection of P onto Δ.

Lemma B.1.3 (Pythagorean inequality for KL divergence, e.g. [Csiszár, 1975], Theorem 2.2). *Let Δ be a convex subset of distributions, then for any distribution P and $Q \in \Delta$*

$$KL(Q|P) \geq KL(Q|\Pi_\Delta(P)) + KL(\Pi_\Delta(P)|P). \tag{B.3}$$

B.2 Proof of Theorem 5.2.1

Step 1. We start by proving the following fact.

For any sequence of distributions $\{Q_{t,i}\}$ such that $Q_{t,i}$ depends only on the samples before and including $(x_{t,i}, y_{t,i})$, for any $\lambda > 0$, we have with probability $1-\delta$

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}R(Q_{t,i}, D_t) \leq \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}[\ell_t(h, z_{t,i})] + \frac{1}{\lambda}\sum_{t=1}^{n}\sum_{i=1}^{m_t} KL(Q_{t,i}|Q_{t,i-1}) \tag{B.4}$$

$$+ \frac{\lambda n}{8\bar{m}} + \frac{1}{\lambda}\log\frac{1}{\delta}. \tag{B.5}$$

The same holds for the reverse statement. Let

$$u_{t,i}(h) = \ell_t(h, z_{t,i}) - R(h, D_t) \tag{B.6}$$

and apply Lemma B.1.1 to $\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}[u_{t,i}(h)]$ to get for any $\lambda > 0$:

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}[u_{t,i}(h)] \leq \frac{1}{\lambda}\sum_{t=1}^{n}\sum_{i=1}^{m_t}\mathsf{KL}(Q_{t,i}|Q_{t,i-1}) \tag{B.7}$$

$$+ \log \mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t,i}} \exp(\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{\lambda}{m_t}u_{t,i}(h_{t,i})). \tag{B.8}$$

Now let $v_{t,i} = e^{\frac{\lambda}{m_t}u_{t,i}(h_{t,i})}$ and we write

$$\mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t,i}}\prod_{t=1}^{n}\prod_{i=1}^{m_t}v_{t,i}\right] \tag{B.9}$$

$$= \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t,i}}\prod_{t=1}^{n}\prod_{i=1}^{m_t}v_{t,i}\middle| k_{1:n}, S_{1:n-1}\right]\right] \tag{B.10}$$

$$= \mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t<n,i}}\left[\prod_{t=1}^{n-1}\prod_{i=1}^{m_t}v_{t,i}\right]\mathbb{E}\left[\mathbb{E}_{\{h_{n,i}\sim Q_{n,i-1}\}_i}\prod_{i=1}^{m_n}v_{n,i}\middle| k_{1:n}, S_{1:n-1}\right]\right]. \tag{B.11}$$

Conditioned on $k_{1:n}$, $u_{n,i}(h_{n,i})$ forms a martingale difference sequence when $h_{n,i}$ is sampled from $Q_{n,i-1}$. Therefore, we can bound

$$\mathbb{E}\left[\mathbb{E}_{\{h_{n,i}\sim Q_{n,i-1}\}_i}\prod_{i=1}^{m_n}v_{n,i}\middle| k_{1:n}, S_{1:n-1}\right] \leq \exp(\frac{\lambda^2}{8m_n}). \tag{B.12}$$

Applying the same argument iteratively, we obtain

$$\mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t,i}}\prod_{t=1}^{n}\prod_{i=1}^{m_t}v_{t,i}\right] \leq \prod_{t=1}^{n}\exp(\frac{\lambda^2}{8m_t}) = \exp(\frac{\lambda^2 n}{8\bar{m}}). \tag{B.13}$$

We get the final statement of the theorem by using Markov inequality. The reverse statement can be obtained by applying the same arguments to $-u_{t,i}$.

Step 2. The next fact is similar fact to the previous one, but for a fixed posterior distribution.

Let P be a fixed prior distribution that is independent of the data. Then for any $\lambda > 0$, we have with probability $1-\delta$ uniformly over Q

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}R(Q, D_t) \leq \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_Q[\ell_t(h, z_{t,i})] + \frac{1}{\lambda}KL(Q|P) \tag{B.14}$$

$$+ \frac{\lambda n}{8\bar{m}} + \frac{1}{\lambda}\log\frac{1}{\delta}. \tag{B.15}$$

The same holds for the reverse statement.

Observe that

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}R(Q,D_t)-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_Q\left[\ell_t(h,z_{t,i})\right] \tag{B.16}$$

$$=\mathbb{E}_Q\left[\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}R(h,D_t)-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\ell_t(h,z_{t,i})\right]. \tag{B.17}$$

Now we can apply the lemma B.1.1 to the whole sum under the expectation to get for any $\lambda>0$

$$\mathbb{E}_Q\left[\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}R(h,D_t)-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\ell_t(h,z_{t,i})\right] \tag{B.18}$$

$$\leq\frac{1}{\lambda}\mathrm{KL}(Q|P)+\frac{1}{\lambda}\log\mathbb{E}_P e^{\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{\lambda}{m_t}R(h,D_t)-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{\lambda}{m_t}\ell_t(h,z_{t,i})} \tag{B.19}$$

and the second term can be bounded in the same way as in the step 1.

Step 3. Now we focus on the empirical regret of the MTLAB algorithm.

The sequence of distributions produced by MTLAB satisfies for any $Q\in\Delta$:

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h,z_{t,i})\right]-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_Q\left[\ell_t(h,z_{t,i})\right]+\frac{1}{\eta}\sum_{t=1}^{n}\sum_{i=1}^{m_t}KL(Q_{t,i}|Q_{t,i-1}) \tag{B.20}$$

$$\leq\frac{1}{\eta}KL(Q|P). \tag{B.21}$$

We start by proving that the optimization in 5.2 can be performed in two steps: first computing a minimizer over the set of all distributions and then taking a KL-projection onto Δ. Let

$$\tilde{Q}_{t,i}=\operatorname*{argmin}_Q\{\frac{\eta}{m_t}\mathbb{E}_{h\sim Q}\left[\ell_t(h,z_{t,i})\right]+\mathrm{KL}(Q|Q_{t,i-1})\}, \tag{B.22}$$

where the minimization is over the set of all distributions. We claim that $Q_{t,i}=\Pi_\Delta(\tilde{Q}_{t,i})$. To prove this, first observe that $\tilde{Q}_{t,i}$ can be computed explicitly, see e.g. [Catoni, 2004], Chapter 5.2,

$$d\tilde{Q}_{t,i}(h)=\frac{e^{-\frac{\eta}{m_t}\ell_t(h,z_{t,i})}dQ_{t,i-1}(h)}{\int e^{-\frac{\eta}{m_t}\ell_t(h',z_{t,i})}dQ_{t,i-1}(h')}=\frac{e^{-\frac{\eta}{m_t}\ell_t(h,z_{t,i})}dQ_{t,i-1}(h)}{W_t}. \tag{B.23}$$

Moreover, from this can compute for any Q

$$\mathrm{KL}(Q|\tilde{Q}_{t,i})=\log W_t+\frac{\eta}{m_t}\mathbb{E}_{h\sim Q}\left[\ell_t(h,z_{t,i})\right]+\mathrm{KL}(Q|Q_{t,i-1}), \tag{B.24}$$

hence, any minimizer of one side also minimizes the other and $Q_{t,i}=\Pi_\Delta(\tilde{Q}_{t,i})$. Now, using again (B.24), we can write

$$\frac{\eta}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h,z_{t,i})\right]-\frac{\eta}{m_t}\mathbb{E}_Q\left[\ell_t(h,z_{t,i})\right] \tag{B.25}$$

$$=\mathrm{KL}(Q_{t,i}|\tilde{Q}_{t,i})-\mathrm{KL}(Q_{t,i}|Q_{t,i-1})-\mathrm{KL}(Q|\tilde{Q}_{t,i})+\mathrm{KL}(Q|Q_{t,i-1}). \tag{B.26}$$

Notice that from Lemma B.1.1: $-\mathrm{KL}(Q|\tilde{Q}_{t,i}) \leq -\mathrm{KL}(Q|Q_{t,i}) - \mathrm{KL}(Q_{t,i}|\tilde{Q}_{t,i})$ and, hence, we get

$$\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h, z_{t,i})\right] - \frac{1}{m_t}\mathbb{E}_{Q}\left[\ell_t(h, z_{t,i})\right] \tag{B.27}$$

$$\leq \frac{1}{\eta}\mathrm{KL}(Q|Q_{t,i-1}) - \frac{1}{\eta}\mathrm{KL}(Q|Q_{t,i}) - \frac{1}{\eta}\mathrm{KL}(Q_{t,i}|Q_{t,i-1}). \tag{B.28}$$

Summing both sides from 1 to n, we obtain a telescoping sum on the right-hand side and get the statement of the theorem.

Step 4. Now we combine the steps above to get the final theorem statement. Steps 1 and 2 with $\lambda = \eta$ get us

$$\mathcal{R}_n(Q) \leq \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h, z_{t,i})\right] - \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q}\left[\ell_t(h, z_{t,i})\right] \tag{B.29}$$

$$+ \frac{1}{\eta}\sum_{t=1}^{n}\sum_{i=1}^{m_t}\mathrm{KL}(Q_{t,i}|Q_{t,i-1}) + \frac{1}{\eta}\mathrm{KL}(Q|P) + \frac{\eta n}{4m} + \frac{1}{\eta}\log\frac{2}{\delta}. \tag{B.30}$$

And then Step 3 bounds :

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h, x_{t,i}, y_{t,i})\right] - \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q}\left[\ell_t(h, x_{t,i}, y_{t,i})\right] \tag{B.31}$$

$$\leq \frac{1}{\eta}\mathrm{KL}(Q|P) - \frac{1}{\eta}\sum_{t=1}^{n}\sum_{i=1}^{m_t}\mathrm{KL}(Q_{t,i}|Q_{t,i-1}). \tag{B.32}$$

B.3 Proof of Theorem 5.6.2

From the definition of discrepancies we can bound

$$R(\bar{Q}_n, D_n) = \frac{1}{n}\sum_{t=1}^{n}R(\hat{Q}_t, D_n) \leq \frac{1}{n}\sum_{t=1}^{n}R(\hat{Q}_t, D_t) + \frac{1}{n}\sum_{t=1}^{n}\mathrm{disc}(k_t, k_n). \tag{B.33}$$

Similarly,

$$-R(Q, D_n) \leq -\frac{1}{n}\sum_{t=1}^{n}R(Q, D_t) + \frac{1}{n}\sum_{t=1}^{n}\mathrm{disc}(k_t, k_n). \tag{B.34}$$

Then the statement follows from Corollary 5.2.2.

B.4 Technical results for MTLAB.MS

We will use the following notations for the MTLAB.MS algorithm:

- $\mathcal{J}(j)$ is the set of indices assigned to j-th subsequence.
- $s_n(j) = |\mathcal{J}(j)|$.
- I_n is the index of the subsequence chosen at the step n.
- $Q_{t,i}(j)$ is the distribution produced by the transfer algorithm on the subsequence j on the i-th sample of t-th task.
- $\bar{m}(j)$ is the harmonic mean of the sizes of tasks in the j-th subsequence.
- $\hat{Q}_t(j)$ the distribution outputted by the transfer algorithm for t-th task of j-th subsequence.

Lemma B.4.1. *For each subsequence produced by MTLAB.MS, for any* $\lambda > 0$

$$\mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}(j)\}} e^{\sum_{t\in\mathcal{J}(j)}\sum_{i=1}^{m_t}\frac{\lambda}{m_t}(R(h_{t,i},D_t)-\ell_t(h_{t,i},z_{t,i}))-\frac{\eta^2 s_n(j)}{\bar{m}(j)}}\right] \leq 1. \tag{B.35}$$

Proof of Lemma B.4.1. This a simpler version of Lemma A.1.3. □

The next lemma is an analog of Step 1 in the proof of Theorem 5.2.1 adjusted to the fact that we use only a randomly chosen subsequence of the tasks.

Lemma B.4.2. *The sequence of distributions produced by the MTLAB.MS algorithm for the chosen subsequence satisfies for any* $\lambda(j) > 0, \forall j$ *with probability* $1-\delta$

$$\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\frac{1}{m_t}(R(Q_{t,i}(I_n),D_t) - \mathbb{E}_{Q_{t,i}(I_n)}[\ell_t(h,z_{t,i})]) \tag{B.36}$$

$$\leq \frac{1}{\lambda(I_n)}\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t} KL(Q_{t,i}|Q_{t,i-1}) + \frac{\lambda(I_n)s_n(I_n)}{\bar{m}(I_n)} + \frac{1}{\lambda(I_n)}\log\frac{n}{\delta}. \tag{B.37}$$

The same holds for the reverse statement.

Proof of Lemma B.4.2. We start with the Lemma S1, which applies even if the distributions are random, to get

$$\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\frac{1}{m_t}(R(Q_{t,i}(I_n),D_t) - \mathbb{E}_{Q_{t,i}(I_n)}[\ell_t(h,z_{t,i})]) \tag{B.38}$$

$$\leq \frac{1}{\lambda(I_n)}\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t} \mathrm{KL}(Q_{t,i}|Q_{t,i-1}) \tag{B.39}$$

$$+ \frac{1}{\lambda(I_n)}\log\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}(I_n)\}} e^{\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\frac{\lambda(I_n)}{m_t}(R(h_{t,i},D_t)-\ell_t(h_{t,i},z_{t,i}))}. \tag{B.40}$$

Now

$$\frac{1}{\lambda(I_n)} \log \mathbb{E}_{\{h_{t,i} \sim Q_{t,i-1}(I_n)\}} e^{\sum_{t \in \mathcal{J}(I_n)} \sum_{i=1}^{m_t} \frac{\lambda(I_n)}{m_t} (R(h_{t,i}, D_t) - \ell_t(h_{t,i}, z_{t,i}))} \tag{B.41}$$

$$\leq \frac{1}{\lambda(I_n)} \log \mathbb{E}_{\{h_{t,i} \sim Q_{t,i-1}(I_n)\}} e^{\sum_{t \in \mathcal{J}(I_n)} \sum_{i=1}^{m_t} \frac{\lambda(I_n)}{m_t} (R(h_{t,i}, D_t) - \ell_t(h_{t,i}, z_{t,i})) - \frac{\lambda^2(I_n) s_n(I_n)}{\bar{m}(I_n)}} \tag{B.42}$$

$$+ \frac{\lambda(I_n) s_n(I_n)}{\bar{m}(I_n)}. \tag{B.43}$$

Denote

$$\Omega(j) = \mathbb{E}_{\{h_{t,i} \sim Q_{t,i-1}(j)\}} e^{\sum_{t \in \mathcal{J}(j)} \sum_{i=1}^{m_t} \frac{\lambda(j)}{m_t} (R(h_{t,i}, D_t) - \ell_t(h_{t,i}, z_{t,i})) - \frac{\lambda^2(j) s_n(j)}{\bar{m}(j)}}, \tag{B.44}$$

so that from Lemma B.4.1 we get $\mathbb{E}[\Omega(j)] \leq 1$. To finish the proof we need to bound $\frac{1}{\lambda(I_n)} \log \Omega(I_n)$. For this observe

$$\mathbb{P}[\Omega(I_n) > \alpha] \leq \sum_j \mathbb{P}[\Omega(j) > \alpha] \leq \sum_j \frac{\mathbb{E}[\Omega(j)]}{\alpha} \leq \frac{n}{\alpha}. \tag{B.45}$$

This gives us that with probability $1 - \delta$:

$$\log \Omega(I_n) \leq \log \frac{n}{\delta} \tag{B.46}$$

and finishes the proof. □

Lemma B.4.3. *Let P be a fixed prior distribution independent of the data. For any $\lambda(j) > 0, \forall j$ it holds with probability $1 - \delta$ uniformly in Q*

$$\sum_{t \in \mathcal{J}(I_n)} \sum_{i=1}^{m_t} \frac{1}{m_t} (R(Q, D_t) - \mathbb{E}_Q[\ell_t(h, z_{t,i})]) \leq \frac{1}{\lambda(I_n)} KL(Q|P) + \frac{\lambda(I_n) s_n(I_n)}{\bar{m}(I_n)} + \frac{1}{\lambda(I_n)} \log \frac{n}{\delta}. \tag{B.47}$$

The same holds for the reverse statement.

Proof. The proof is an analogue of Step 2 of the proof of Theorem 5.2.1, but using arguments from Lemma B.4.2 for the last part. □

B.5 Proof of Theorem 5.6.3

Observe that from the definition of discrepancies we get.

$$R(\bar{Q}_n, D_n) - R(Q, D_n) \leq \frac{1}{s_n} \sum_{t \in \mathcal{J}(I_n)} (R(\hat{Q}_t(I_n), D_t) - R(Q, D_t)) + \frac{2}{s_n} \sum_{t \in \mathcal{J}(I_n)} \mathrm{disc}(k_t, k_n). \tag{B.48}$$

From the way MTLAB.MS constructs the subsequences, we have $\frac{1}{s_n}\sum_{t\in\mathcal{J}(I_n)}\mathrm{disc}(k_t,k_n)\leq\varepsilon$. Using Lemma B.4.2 and B.4.3 with $\lambda(I_n)=\eta_n$, we get with probability $1-\delta$

$$\sum_{t\in\mathcal{J}(I_n)}(R(\hat{Q}_t(I_n),D_t)-R(Q,D_t))\leq\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\frac{1}{m_t}(\mathbb{E}_{Q_{t,i}}[\ell_t(h,z_{t,i})]-\mathbb{E}_Q[\ell(h,z_{t,i})]) \tag{B.49}$$

$$+\frac{1}{\eta_n}\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\mathrm{KL}(Q_{t,i}|Q_{t,i-1})+\frac{1}{\eta_n}\mathrm{KL}(Q|P) \tag{B.50}$$

$$+2\frac{\eta_n s_n}{\bar{m}_n}+\frac{1}{\eta_n}\log\frac{n}{\delta}. \tag{B.51}$$

The final statement follows from the regret bound of Step 2 in the proof of Theorem 5.2.1.

B.6 Proof of Theorem 5.6.4

Proof of Theorem 5. Observe that from the definition of discrepancies we get.

$$R(\bar{Q}_n,D_n)-R(Q,D_n)\leq\frac{1}{s_n}\sum_{t\in\mathcal{J}(I_n)}(R(\hat{Q}_t(I_n),D_t)-R(Q,D_t))+\frac{2}{s_n}\sum_{t\in\mathcal{J}(I_n)}\mathrm{disc}(k_t,k_n). \tag{B.52}$$

From (22) we get with probability $1-\delta$ that

$$\frac{2}{s_n}\sum_{t\in\mathcal{J}(I_n)}\mathrm{disc}(k_t,k_n)\leq 2\varepsilon+\frac{2}{s_n}\sum_{t\in\mathcal{J}(I_n)}B_L(t,n,\frac{\delta}{n}). \tag{B.53}$$

Using Lemma B.4.2 and Lemma B.4.3 with $\lambda(I_n)=\eta_n$, we get with probability $1-\delta$

$$\sum_{t\in\mathcal{J}(I_n)}(R(\hat{Q}_t(I_n),D_t)-R(Q,D_t))\leq\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\frac{1}{m_t}(\mathbb{E}_{Q_{t,i}}[\ell(h,z_{t,i})]-\mathbb{E}_Q[\ell(h,z_{t,i})]) \tag{B.54}$$

$$+\frac{1}{\eta_n}\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\mathrm{KL}(Q_{t,i}|Q_{t,i-1})+\frac{1}{\eta_n}\mathrm{KL}(Q|P) \tag{B.55}$$

$$+2\frac{\eta_n s_n}{\bar{m}_n}+\frac{1}{\eta_n}\log\frac{n}{\delta}. \tag{B.56}$$

The final statement follows from the regret bound of Step 2 in the proof of Theorem 5.2.1. □

B.7 Proof of Theorem 5.6.6

Proof of Theorem 6. We start by applying the standard arguments from the transfer learning literature, see e.g. [Ben-David *et al.*, 2007].

$$R(\bar{Q}_n, D_n) - R(Q, D_n) \leq \frac{1}{s_n} \sum_{t \in \mathcal{J}(I_n)} (R(\hat{Q}_t(I_n), D_t) - R(Q, D_t)) \tag{B.57}$$

$$+ \frac{2}{s_n} \sum_{t \in \mathcal{J}(I_n)} e(k_t, k_n) + \frac{2}{s_n} \sum_{t \in \mathcal{J}(I_n)} \lambda_{t,n}. \tag{B.58}$$

MTLAB.MS is specifically designed to control the estimated discrepancies, so that we get with probability $1 - \delta$

$$\frac{2}{s_n} \sum_{t \in \mathcal{J}(I_n)} e(k_t, k_n) \leq 2\varepsilon + \frac{2}{s_n} \sum_{t \in \mathcal{J}(I_n)} B_U(U_t, U_n, \frac{\delta}{n}). \tag{B.59}$$

Using Lemma B.4.2 and Lemma B.4.3 with $\lambda(I_n) = \eta_n$, we get with probability $1 - \delta$

$$\sum_{t \in \mathcal{J}(I_n)} (R(\hat{Q}_t(I_n), D_t) - R(Q, D_t)) \leq \sum_{t \in \mathcal{J}(I_n)} \sum_{i=1}^{m_t} \frac{1}{m_t} (\mathbb{E}_{Q_{t,i}} [\ell(h, f_{t,i}, x_{t,i})] - \mathbb{E}_Q [\ell(h, f_{t,i}, x_{t,i})]) \tag{B.60}$$

$$+ \frac{1}{\eta_n} \sum_{t \in \mathcal{J}(I_n)} \sum_{i=1}^{m_t} \mathrm{KL}(Q_{t,i} | Q_{t,i-1}) + \frac{1}{\eta_n} \mathrm{KL}(Q|P) \tag{B.61}$$

$$+ 2\frac{\eta_n s_n}{\bar{m}_n} + \frac{1}{\eta_n} \log \frac{n}{\delta}. \tag{B.62}$$

The final statement follows from the regret bound of Step 2 in the proof of Theorem 5.2.1. □