

UNIVERSIDAD AUTÓNOMA DE MADRID

# Advanced Kernel Methods for Multi-Task Learning

by

Carlos Ruiz Pastor

A thesis submitted in partial fulfillment for the  
degree of Doctor of Philosophy

in the

Escuela Politécnica Superior  
Computer Science Department

under the supervision of José R. Dorronsoro Ibero

August 2021



*What is the essence of life? To serve others and to do good.*

Aristotle.

## *Abstract*

## *Resumen*

## *Acknowledgements*

.

# Contents

<b>Abstract</b>	<b>iv</b>
<b>Resumen</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>Abbreviations</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Publications . . . . .	1
1.3 Summary by Chapters . . . . .	1
1.4 Definitions and Notation . . . . .	3
<b>2 Foundations and Concepts</b>	<b>5</b>
2.1 Introduction . . . . .	6
2.2 Kernels . . . . .	6
2.2.1 Motivation and Definition . . . . .	6
2.2.2 Reproducing Kernel Hilbert Spaces . . . . .	6
2.2.3 Examples and Properties . . . . .	6
2.3 Risk Functions and Regularization . . . . .	6
2.3.1 Empirical and Expected Risk . . . . .	6
2.3.2 Regularized Risk Functional . . . . .	6
2.3.3 Representer Theorem . . . . .	6
2.4 Optimization . . . . .	6
2.4.1 Convex Optimization . . . . .	6
2.4.2 Unconstrained Problems . . . . .	6
2.4.3 Constrained Problems . . . . .	6
2.5 Statistical Learning . . . . .	6
2.5.1 Uniform Convergence and Consistency . . . . .	6
2.5.2 VC dimension and Structural Learning . . . . .	6
2.6 Support Vector Machines . . . . .	6
2.6.1 Linearly Separable Case . . . . .	6
2.6.2 Non-Linearly Separable Case . . . . .	6

2.6.3	Kernel Extension . . . . .	6
2.6.4	SVM properties . . . . .	6
2.6.5	Connection with Structural Learning . . . . .	6
2.6.6	SVM Variants . . . . .	6
2.7	Conclusions . . . . .	6
<b>3</b>	<b>Multi-Task Learning</b>	<b>7</b>
3.1	Introduction . . . . .	7
3.2	Why does Multi-Task Learning work? . . . . .	7
3.2.1	Inductive Bias Learning Problem . . . . .	7
3.2.2	Learning with Related Tasks . . . . .	14
3.2.3	Learning Under Privileged Information . . . . .	15
3.3	Multi-Task Learning Methods: An Overview . . . . .	15
3.3.1	Bias Learning Approach . . . . .	15
3.3.2	Low-Rank Approach . . . . .	15
3.3.3	Learning Task Relations . . . . .	15
3.3.4	Decomposition Approach . . . . .	15
3.4	Deep Multi-Task Learning . . . . .	15
3.4.1	Hard Parameter Sharing . . . . .	15
3.4.2	Soft Parameter Sharing . . . . .	15
3.5	Multi-Task Learning with Kernel Methods . . . . .	15
3.6	Conclusions . . . . .	15
<b>4</b>	<b>A Convex Formulation for Regularized Multi-Task Learning</b>	<b>17</b>
4.1	Introduction . . . . .	17
4.2	Convex Multi-Task Learning Support Vector Machines . . . . .	17
4.2.1	Convex Formulation . . . . .	17
4.2.2	L1 Support Vector Machine . . . . .	17
4.2.3	L2 Support Vector Machine . . . . .	17
4.2.4	LS Support Vector Machine . . . . .	17
4.3	Optimal Convex Combination of trained models . . . . .	17
4.4	Experiments . . . . .	17
4.5	Conclusions . . . . .	17
<b>5</b>	<b>Adaptive Graph Laplacian Multi-Task Support Vector Machine</b>	<b>19</b>
5.1	Introduction . . . . .	19
5.2	Graph Laplacian Multi-Task Support Vector Machine . . . . .	19
5.3	Adaptive Graph Laplacian Algorithm . . . . .	19
5.4	Experiments . . . . .	19
5.5	Conclusions . . . . .	19



# Abbreviations

<b>ADF</b>	<b>A</b> ssumed <b>D</b> ensity <b>F</b> iltering
<b>AF</b>	<b>A</b> cquisition <b>F</b> unction
<b>BO</b>	<b>B</b> ayesian <b>O</b> ptimization
<b>DGP</b>	<b>D</b> ee <b>P</b> <b>G</b> aussian <b>P</b> rocess
<b>EI</b>	<b>E</b> xpected <b>I</b> mprovement
<b>EP</b>	<b>E</b> xpectation <b>P</b> ropagation
<b>GP</b>	<b>G</b> aussian <b>P</b> rocess
<b>KL</b>	<b>K</b> ullback <b>L</b> iebler
<b>MCMC</b>	<b>M</b> arkov <b>C</b> hain <b>M</b> onte <b>C</b> arlo
<b>PPESMOC</b>	<b>P</b> arallel <b>P</b> redictive <b>E</b> ntropy <b>S</b> earch for <b>M</b> ultiobjective <b>O</b> ptimization with <b>C</b> onstraints
<b>PES</b>	<b>P</b> redictive <b>E</b> ntropy <b>S</b> earch
<b>PESMOC</b>	<b>P</b> redictive <b>E</b> ntropy <b>S</b> earch for <b>M</b> ultiobjective <b>O</b> ptimization with <b>C</b> onstraints
<b>RS</b>	<b>R</b> andom <b>S</b> earch
<b>UCB</b>	<b>U</b> pper <b>C</b> onfidence <b>B</b> ound



*To my family*



# Chapter 1

## Introduction

We begin this manuscript...

### 1.1 Introduction

### 1.2 Publications

This section presents, in chronological order, the work published during the doctoral period in which this thesis was written. We also include other research work related to this thesis, but not directly included on it. Finally, this document includes content that has not been published yet and is under revision.

### Related Work

### Work In Progress

### 1.3 Summary by Chapters

In this section...

**Chapter 3** provides an introduction to GPs and the expectation propagation algorithm.

Both are necessary concepts for the BO methods that we will describe in the following chapters. This chapter reviews the fundamentals of GPs and why they are so interesting for BO. More concretely, we review the most popular kernels, the analysis of the posterior and predictive distribution and how to tune the hyper-parameters of GPs: whether by maximizing the marginal likelihood or by generating samples from the hyper-parameter posterior distribution. Other alternative probabilistic surrogate models are also described briefly. Some of the proposed approaches of this thesis are extensions of an acquisition function called predictive entropy search, that is based on the expectation propagation approximate inference technique. That is why we provide in this chapter an explanation of the expectation propagation algorithm.

**Chapter 5** introduces the basics of BO and information theory. BO works with probabilistic models such as GPs and with acquisition functions such as predictive entropy search, that uses information theory. Having studied GPs in Chapter 3, BO can be now understood and it is described in detail. This chapter will also

describe the most popular acquisition functions, how information theory can be applied in BO and why BO is useful for the hyper-parameter tuning of machine learning algorithms.

**Chapter ??** describes an information-theoretical mechanism that generalizes BO to simultaneously optimize multiple objectives under the presence of several constraints. This algorithm is called predictive entropy search for multi-objective BO with constraints (PESMOC) and it is an extension of the predictive entropy search acquisition function that is described in Chapter 5. The chapter compares the empirical performance of PESMOC with respect to a state-of-the-art approach to constrained multi-objective optimization based on the expected improvement acquisition function. It is also compared with a random search through a set of synthetic, benchmark and real experiments.

**Chapter ??** addresses the problem that faces BO when not only one but multiple input points can be evaluated in parallel that has been described in Section ???. This chapter introduces an extension of PESMOC called parallel PESMOC (PPESMOC) that adapts to the parallel scenario. PPESMOC builds an acquisition function that assigns a value for each batch of points of the input space. The maximum of this acquisition function corresponds to the set of points that maximizes the expected reduction in the entropy of the Pareto set in each evaluation. Naive adaptations of PESMOC and the method based on expected improvement for the parallel scenario are used as a baseline to compare their performance with PPESMOC. Synthetic, benchmark and real experiments show how PPESMOC obtains an advantage in most of the considered scenarios. All the mentioned approaches are described in detail in this chapter.

**Chapter ??** addresses a transformation that enables standard GPs to deliver better results in problems that contain integer-valued and categorical variables. We can apply BO to problems where we need to optimize functions that contain integer-valued and categorical variables with more guarantees of obtaining a solution with low regret. A critical advantage of this transformation, with respect to other approaches, is that it is compatible with any acquisition function. This transformation makes the uncertainty given by the GPs in certain areas of the space flat. As a consequence, the acquisition function can also be flat in these zones. This phenomenon raises an issue with the optimization of the acquisition function, that must consider the flatness of these areas. We use a one exchange neighbourhood approach to optimize the resultant acquisition function. We test our approach in synthetic and real problems, where we add empirical evidence of the performance of our proposed transformation.

**Chapter ??** shows a real problem where BO has been applied with success. In this problem, BO has been used to obtain the optimal parameters of a hybrid Grouping Genetic Algorithm for attribute selection. This genetic algorithm is combined with an Extreme Learning Machine (GGA-ELM) approach for prediction of ocean wave features. Concretely, the significant wave height and the wave energy flux at a goal marine structure facility on the Western Coast of the USA is predicted. This chapter illustrates the experiments where it is shown that BO improves the performance of the GGA-ELM approach. Most importantly, it also outperforms a random search of the hyper-parameter space and the human expert criterion.

**Chapter ??** provides a summary of the work done in this thesis. We include the conclusions retrieved by the multiple research lines covered in the chapters. We also illustrate lines for future research.

## 1.4 Definitions and Notation





# Chapter 2

## Foundations and Concepts

This chapter presents...

## 2.1 Introduction

## 2.2 Kernels

### 2.2.1 Motivation and Definition

### 2.2.2 Reproducing Kernel Hilbert Spaces

### 2.2.3 Examples and Properties

## 2.3 Risk Functions and Regularization

### 2.3.1 Empirical and Expected Risk

### 2.3.2 Regularized Risk Functional

### 2.3.3 Representer Theorem

## 2.4 Optimization

### 2.4.1 Convex Optimization

### 2.4.2 Unconstrained Problems

### 2.4.3 Constrained Problems

## 2.5 Statistical Learning

### 2.5.1 Uniform Convergence and Consistency

### 2.5.2 VC dimension and Structural Learning

## 2.6 Support Vector Machines

### 2.6.1 Linearly Separable Case

### 2.6.2 Non-Linearly Separable Case

### 2.6.3 Kernel Extension

### 2.6.4 SVM properties

### 2.6.5 Connection with Structural Learning

### 2.6.6 SVM Variants

## 2.7 Conclusions

In this chapter, we covered...

# Multi-Task Learning

This chapter presents...

## 3.1 Introduction

## 3.2 Why does Multi-Task Learning work?

### 3.2.1 Inductive Bias Learning Problem

Typically in Machine Learning the goal is to find the best hypothesis  $h(x, \alpha)$  from a space of hypothesis  $\mathcal{H} = \{h(x, \alpha), \alpha \in \Lambda\}$ , where  $\Lambda$  is any set of parameters. This best candidate can be selected according to different inductive principles, which define a method of approximating a global function  $f(x)$  from a training set:  $z := \{(x_i, y_i), i = 1, \dots, n\}$  where  $(x_i, y_i)$  are sampled from a distribution  $F$ . In the classical statistics we find the Maximum Likelihood approach, where the goal is to estimate the density  $f(x) = p(y | x)$  and the hypothesis space is parametric, i.e.  $\mathcal{H} = \{h(x, \alpha), \alpha \in \Lambda \subset \mathbb{R}^m\}$ . The learner select the parameter  $\alpha$  that maximizes the probability of the data given the hypothesis. Another more direct inductive principle is Empirical Risk Minimization (ERM), which is the most common one. In ERM the densities are ignored and an empirical error  $\hat{R}_z$  is minimized with the hope of minimizing the true expected error  $R_F$ , which would result in a good generalization. Several models use the ERM principle to generalize from data such as Neural Networks or Support Vector Machines. These methods are designed to find a good hypothesis  $h(x, \alpha)$  from a given space  $\mathcal{H}$ . The definition of such space  $\mathcal{H}$  define the bias for these problems. If  $\mathcal{H}$  does not contain any good hypothesis, the learner will not be able to learn.

The best hypothesis space we can provide is the one containing only the optimal hypothesis, but this is the original problem that we want to solve. Therefore, in the single task scenario, there is no difference between bias learning and ordinary learning. Instead, we focus on the situation where we want to solve multiple related tasks. In that case, we can obtain a good space  $\mathcal{H}$  that contains good solutions for the different tasks. In [Baxter \(2000\)](#) an effort is made to define the concepts needed to construct the theory about inductive bias learning, which can be seen as a generalization of strict multi-task learning. This is done by defining an environment of tasks and extending the work of [Vapnik \(2013\)](#), which defines the capacity of space of hypothesis, Baxter defines the capacity of a family of spaces of hypothesis.

Before presenting the concepts defined for Bias Learning, and to establish an analogy to those of ordinary learning, we briefly review some statistical learning concepts.

### Ordinary Learning

In the ordinary statistical learning, some theoretical concepts are used:

- an *input space*  $\mathcal{X}$  and an *output space*  $\mathcal{Y}$ ,
- a *probability distribution*  $F$ , which is unknown, defined over  $\mathcal{X} \times \mathcal{Y}$ ,
- a *loss function*  $\ell(\cdot, \cdot) : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ , and
- a *hypothesis space*  $\mathcal{H} = \{h(x, \alpha), \alpha \in \Lambda \subset \mathbb{R}^m\}$  with hypothesis  $h(\cdot, \alpha) : \mathcal{X} \rightarrow \mathcal{Y}$ .

The goal for the learner is to select a hypothesis  $h(x, \alpha) \in \mathcal{H}$ , or equivalently  $\alpha \in \Lambda$ , that minimizes the expected risk

$$R_F(\alpha) = \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(x, \alpha), y) dF(x, y).$$

The distribution  $F$  is unknown, but we have a training set  $z = \{(x_1, y_1), \dots, (x_n, y_n)\}$  of samples drawn from  $F$ . The approach is then is to apply the ERM inductive principle, that is to minimize the empirical risk

$$\hat{R}_z(\alpha) = \frac{1}{n} \sum_{i=1}^n l(h(x_i), y_i).$$

Thus, a learner  $\mathcal{A}$  maps the set of training samples to a set of hypothesis:

$$\mathcal{A} : \bigcup (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{H}.$$

Although  $\hat{R}_z(\alpha)$  is an unbiased estimator of  $R_F(\alpha)$ , it has been shown [Vapnik \(2013\)](#) that this approach, despite being the most evident one, is not the best principle that can be followed. This has relation with two facts: the first one is that the unbiased property is an asymptotical one, the second one has to do with overfitting. Vapnik answers to the question of what can be said about  $R_F$  when  $\alpha$  minimizes  $\hat{R}_z(\alpha)$ , and moreover, his results are valid also for small number of training samples  $n$ . More specifically, Vapnik sets the sufficient and necessary conditions for the consistency of an inductive learning process, i.e. for  $\hat{R}_z(\alpha) \xrightarrow{P} R_F(\alpha)$  uniformly. Vapnik also defines the capacity of a hypothesis space and use it to derive bounds on the rate of this convergence for any  $\alpha \in \Lambda$  and, more importantly, bounds on the difference  $\inf_{\alpha \in \Lambda} \hat{R}_z(\alpha) - \inf_{\alpha \in \Lambda} R_F(\alpha)$ . Under some general conditions, he proves that

$$\inf_{\alpha \in \Lambda} \hat{R}_z(\alpha) - \inf_{\alpha \in \Lambda} R_F(\alpha) \leq B(n/\text{VCdim}(\mathcal{H})) \quad (3.1)$$

where  $B$  is some non-decreasing function and  $\text{VCdim}(\mathcal{H})$  is the capacity of the space  $\mathcal{H}$ , also named the VC-dimension  $\mathcal{H}$ . This means that the generalization ability of a learning process can be controlled in terms of two factors:

- The number of training samples  $n$ . A greater number of training samples assures a better generalization of the learning process. This looks intuitive and could be already inferred from the asymptotical properties.

- The VC-dimension  $\text{VCdim}(\mathcal{H})$  of the hypothesis space  $\mathcal{H}$ , which is desirable to be small. This term is not intuitive and is the most important term in Vapnik theory.

The VC-dimension measures the capacity of a set of hypothesis  $\mathcal{H}$ . If the capacity of the set  $\mathcal{H}$  is too large, we may find a hypothesis  $h(x, \alpha^*)$  that minimizes  $\hat{R}_z$  but does not generalize well and therefore, does not minimize  $R_F$ . This is the overfitting problem. On the other side, if we use a simple  $\mathcal{H}$ , with low capacity, we could be in a situation where there is not a good hypothesis  $h(x, \alpha) \in \mathcal{H}$ , so the empirical risk  $\inf_{\alpha \in \Lambda} R_F$  is too large. This is the underfitting problem.

### Bias Learning: Concept and Components

In [Baxter \(2000\)](#) two main concepts are presented: the *family of hypothesis spaces* and an *environment* of related tasks. For simplicity we write  $h(x)$  instead of  $h(x, \alpha)$ , and since  $\alpha$  completely defines  $h$ , we also substitute  $\alpha$  by  $h$  for an easier notation. Using these concepts, the bias learning problem has the following components:

- an *input space*  $\mathcal{X}$  and an *output space*  $\mathcal{Y}$ ,
- an *environment*  $(\mathcal{P}, Q)$  where  $\mathcal{P}$  is a set of distributions  $P$  defined over  $\mathcal{X} \times \mathcal{Y}$ , and we can sample from  $\mathcal{P}$  according to a distribution  $Q$ ,
- a *loss function*  $\ell(\cdot, \cdot) : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ , and
- a *family of hypothesis spaces*  $\mathbb{H} = \{\mathcal{H}_\delta, \delta \in \Delta\}$ , where each element  $\mathcal{H}_\delta$  is a set of hypothesis.

Analogous to ordinary learning, the goal is to minimize the expected risk, defined as

$$R_Q(\delta) = \int_{\mathcal{P}} \inf_{h \in \mathcal{H}_\delta} R_P(h) dQ(P) = \int_{\mathcal{P}} \inf_{h \in \mathcal{H}_\delta} \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(x), y) dP(x, y) dQ(P). \quad (3.2)$$

Again, we do not know  $\mathcal{P}$  nor  $Q$ , but we have a training set samples from the environment  $(\mathcal{P}, Q)$  obtained in the following way:

1. Sample  $T$  times from  $Q$  obtaining  $P_1, \dots, P_T \in \mathcal{P}$
2. For  $r = 1, \dots, T$  sample  $m$  pairs  $z_r = \{(x_1^r, y_1^r), \dots, (x_m^r, y_m^r)\}$  according to  $P_r$  where  $(x_i^r, y_i^r) \in \mathcal{X} \times \mathcal{Y}$ .

We obtain a sample  $z = \{(x_i^r, y_i^r), r = 1, i = 1, \dots, m = 1, \dots, T\}$ , with  $m$  examples from  $T$  different learning tasks, and

$$z := \begin{pmatrix} (x_1^1, y_1^1) & \dots & (x_m^1, y_m^1) \\ \vdots & \ddots & \vdots \\ (x_1^T, y_1^T) & \dots & (x_m^T, y_m^T) \end{pmatrix}$$

is named as a  $(T, m)$ -sample. Using  $z$  we can define the empirical loss as

$$\hat{R}_z(\delta) = \sum_{r=1}^T \inf_{h \in \mathcal{H}_\delta} \hat{R}_{z_r}(h) = \sum_{r=1}^T \inf_{h \in \mathcal{H}_\delta} \sum_{i=1}^m \ell(h(x_i^r), y_i^r), \quad (3.3)$$

which is an average of the empirical losses of each task. Note, however, that in the case of the bias learner, this estimate is biased, since  $R_{P_r}(h)$  does not coincide with  $\hat{R}_{z_r}(h)$ .

Putting all together, a bias learner  $\mathcal{A}$  maps the set of all  $(T, m)$ -samples to a family of hypothesis spaces:

$$\mathcal{A} : \bigcup (\mathcal{X} \times \mathcal{Y})^{(T, m)} \rightarrow \mathbb{H}.$$

To follow an analogous path to that of ordinary learning, the milestones in bias learning theory should include:

- Checking the consistency of the Bias Learning methods, i.e. proving that  $\hat{R}_{\mathbf{z}}(\delta)$  converges uniformly in probability to  $R_Q(\delta)$ .
- Defining a notion of capacity of hypothesis space families  $\mathbb{H}$ .
- Finding a bound of  $\hat{R}_{\mathbf{z}}(\delta) - R_Q(\delta)$  for any  $\delta$  using the capacity of the hypothesis space family. If possible, finding also a bound for  $\inf_{\delta \in \Delta} \hat{R}_{\mathbf{z}}(\delta) - \inf_{\delta \in \Delta} R_Q(\delta)$ .

To achieve these goals some previous definitions are needed. From this point, since any  $\mathcal{H}$  is defined by a  $\delta \in \Delta$ , we omit  $\delta$  and write just  $\mathcal{H}$  for simplicity.

### Bias Learning: Capacities and Uniform Convergence

In first place, a *sample-driven* pseudo-metric of  $(T, 1)$ -empirical risks is defined. Consider a sequence of  $T$  probabilities  $\mathbf{P} = (P_1, \dots, P_T)$  sampled from  $\mathcal{P}$  according the the distribution  $Q$ . Consider also the set of sequences of  $T$  hypothesis

$$\mathcal{H}^T := \{\mathbf{h} = (h_1, \dots, h_T), h_1, \dots, h_T \in \mathcal{H}\}.$$

We can define then the set of  $(T, 1)$ -empirical risks as

$$\mathcal{H}_\ell^T := \left\{ \mathbf{h}_\ell(x_1, y_1, \dots, x_T, y_T) = \sum_{r=1}^T \ell(h_r(x_r), y_r), h_1, \dots, h_T \in \mathcal{H} \right\}$$

The family of the set of  $T$ -risks of hypothesis is then  $\mathbb{H}^T = \bigcup_{\mathcal{H} \in \mathbb{H}} \mathcal{H}^T$ . Now we can define

$$d_{\mathbf{P}}(\mathbf{h}_\ell, \mathbf{h}'_\ell) = \int_{(\mathcal{X} \times \mathcal{Y})^T} |\mathbf{h}_\ell(x_1, y_1, \dots, x_T, y_T) - \mathbf{h}'_\ell(x_1, y_1, \dots, x_T, y_T)| \\ dP_1(x_1, y_1) \dots dP_T(x_T, y_T)$$

for  $\mathbf{h}_\ell, \mathbf{h}'_\ell \in \mathcal{H}_\ell, \mathcal{H}_\ell'$  as a pseudo-metric in  $\mathbb{H}^T$ .

Then, a *distribution-driven* pseudo-metric is defined. Given a distribution  $P$  on  $\mathcal{X} \times \mathcal{Y}$ . Consider the set of infimum expected risk for each  $\mathcal{H}$ :

$$\mathcal{H}^* := \inf_{h \in \mathcal{H}} R_P(h).$$

The family of such sets is defined as  $\mathbb{H}^* = \{\mathcal{H}^*, \mathcal{H} \in \mathbb{H}\}$ . The pseudo-metric in this space is given by  $Q$ :

$$d_Q = \int_{\mathcal{P}} |\mathcal{H}_1^* - \mathcal{H}_2^*| dQ$$

With these two pseudo-metrics, two capacities for families of hypothesis spaces are defined. For that the definition of  $\epsilon$ -cover is needed. Given a pseudo-metric  $d_S$  in a space  $\mathcal{S}$ , a set of  $l$  elements  $s_1, \dots, s_l \in \mathcal{S}$  is an  $\epsilon$ -cover of  $\mathcal{S}$  if  $d_S(s, s_i) \leq \epsilon$  for some  $i = 1, \dots, l$ . Let  $\mathcal{N}(\epsilon, \mathcal{S}, d_S)$  denote the size of the smallest  $\epsilon$ -cover. Then, we can define the following capacities of a family space  $\mathbb{H}$ :

- The *sample-driven capacity*  $C(\epsilon, \mathbb{H}^T) := \sup_{\mathbf{P}} \mathcal{N}(\epsilon, \mathbb{H}^T, d_{\mathbf{P}})$ .
- The *distribution-driven capacity*  $C(\epsilon, \mathbb{H}^*) := \sup_Q \mathcal{N}(\epsilon, \mathbb{H}^*, d_Q)$ .

Using these capacities, the convergence (uniformly over all  $\mathcal{H} \in \mathbb{H}$ ) of bias learners can be proved (Baxter, 2000, Theorem 2). Moreover, the bias expected risk is bounded

$$\hat{R}_{\mathbf{z}}(\mathcal{H}) \leq R_Q(\mathcal{H}) + \epsilon$$

with probability  $1 - \delta$ , given sufficiently large  $T$  and  $m$ ,

$$T \geq \max \left( \frac{256}{T\epsilon^2} \log \frac{8C(\frac{\epsilon}{32}, \mathbb{H}^*)}{\delta}, \frac{64}{\epsilon^2} \right), \quad m \geq \max \left( \frac{256}{T\epsilon^2} \log \frac{8C(\frac{\epsilon}{32}, \mathbb{H}^T)}{\delta}, \frac{64}{\epsilon^2} \right).$$

It should be noted that the bound for  $m$  is inversely proportional to  $T$ , that is, the more tasks we have, the less samples we need for each task.

### Multi-Task Learning

The previous result is a result for pure Bias Learning, where we have an  $(\mathbb{H}^*, Q)$ -environment of tasks. In Multi-Task Learning, we have a fixed number of tasks  $T$  and a fixed sequence of distributions  $\mathbf{P} = (P_1, \dots, P_T)$ , where  $P_i$  is a distribution over  $(\mathcal{X} \times \mathcal{Y})^m$ . The goal is not learning a hypothesis space  $\mathcal{H}$  but a sequence of hypothesis  $\mathbf{h} = (h_1, \dots, h_T)$ ,  $h_1, \dots, h_T \in \mathcal{H}$ . Thus, the Multi-Task expected risk is

$$R_{\mathbf{P}}(\mathbf{h}) = \sum_{r=1}^T R_{P_r}(h_r) = \sum_{r=1}^T \int_{\mathcal{X} \times \mathcal{Y}} l(h_r(x), y) dP_r(x, y), \quad (3.4)$$

and the empirical risk is defined as

$$\hat{R}_{\mathbf{z}}(\mathbf{h}) = \sum_{r=1}^T \hat{R}_{z_r}(h_r) = \sum_{r=1}^T \sum_{i=1}^m l(h_r(x_i^r), y_i^r). \quad (3.5)$$

A similar result to that of Bias Learning is given for Multi-Task Learning (Baxter, 2000, Theorem 4):

$$\hat{R}_{\mathbf{z}}(\mathbf{h}) \leq R_{\mathbf{P}}(\mathbf{h}) + \epsilon,$$

with probability  $1 - \delta$  given that the number of samples per task

$$m \geq \max \left( \frac{64}{T\epsilon^2} \log \frac{4C(\frac{\epsilon}{16}, \mathbb{H}^T)}{\delta}, \frac{16}{\epsilon^2} \right).$$

Observe that we do not need the *distribution-driven* capacity in this case, just the *sample-driven* capacity.

### Feature Learning

Feature Learning is a common way to encode bias. The most popular example are Neural Networks, where all the hidden layers can be seen as a Feature Learning engine that learns a mapping from the original space to a space with "strong" features. In general, a set of "strong" feature maps is defined as  $\mathcal{F} = \{f, f : \mathcal{X} \rightarrow \mathcal{V}\}$ . Using these features, functions  $g \in \mathcal{G}$  (which are typically simple) are built:  $\mathcal{X} \rightarrow_f \mathcal{V} \rightarrow_g \mathcal{Y}$ . Thus, for each

map  $f$ , the hypothesis space can be expressed as  $\mathcal{H}_f = \{h = \mathcal{G} \circ f, g \in \mathcal{G}\}$ , and the family of hypothesis spaces is  $\mathbb{H} = \{\mathcal{H}_f, f \in \mathcal{F}\}$ . Now, the Bias Learning problem is the problem of finding a good mapping  $f$ . It is proved (Baxter, 2000, Theorem 6) that in the Feature Learning case the capacities of  $\mathbb{H}$  can be bounded by the capacities of  $\mathcal{F}$  and  $\mathcal{G}$  as

$$\begin{aligned} C(\epsilon, \mathbb{H}^T) &\leq C(\epsilon_1, \mathcal{G}^T)^T C_{\mathcal{G}_\ell}(\epsilon_2, \mathcal{F}), \\ C(\epsilon, \mathbb{H}^*) &\leq C_{\mathcal{G}_\ell}(\epsilon, \mathcal{F}) \end{aligned}$$

with  $\epsilon = \epsilon_1 + \epsilon_2$ . Here,  $C_{\mathcal{G}_\ell}(\epsilon, \mathcal{F})$  is defined as  $C_{\mathcal{G}_\ell}(\epsilon, \mathcal{F}) := \sup_P \mathcal{N}(\epsilon, \mathcal{F}, d_{[P, \mathcal{G}_\ell]})$ , where

$$d_{[P, \mathcal{G}_\ell]}(f, f') = \int_{\mathcal{X} \times \mathcal{Y}} \sup_{g \in \mathcal{G}} |\ell(g \circ f(x), y) - \ell(g \circ f'(x), y)| dP(x, y)$$

is a pseudo-metric. Using these results alongside those presented for Bias Learning is useful to establish bounds for Feature Learning models like Neural Networks.

### Generalized VC-Dimension for Multi-Task Learning

The concepts presented until now rely on the concepts of two capacities of a family of hypothesis spaces  $\mathbb{H}$  to establish bounds in the difference  $\hat{R}_z(\mathbf{h}) - R_Q(\mathbf{h})$ , that is, the probability of large deviations between the empirical and expected risks for a given hypothesis sequence. However, it would be more interesting to establish some bounds between the empirical error and the *best expected error*. To overcome this limitations, a generalized VC-dimension is developed in Baxter (2000) for Multi-Task Learning with Boolean hypothesis.

Let  $\mathcal{H}$  be a space of boolean functions and  $\mathbb{H}$  a boolean hypothesis space family. Denote the set of  $T \times m$  matrices in  $\mathcal{X}$  as  $\mathcal{X}^{T \times m}$ . For each  $X \in \mathcal{X}^{T \times m}$  and each  $\mathcal{H} \in \mathbb{H}$  define the set of binary  $T \times m$  matrices

$$\mathcal{H}_{|X} := \left\{ \begin{pmatrix} h(x_1^1) & \dots & h(x_m^1) \\ \vdots & \ddots & \vdots \\ h(x_1^T) & \dots & h(x_m^T) \end{pmatrix}, h \in \mathcal{H} \right\},$$

and the corresponding family of such sets as

$$\mathbb{H}_{|X} = \bigcup_{\mathcal{H} \in \mathbb{H}} \mathcal{H}_{|X}.$$

For each  $T, m \geq 0$  define the number of binary matrices obtainable with  $\mathbb{H}$  as

$$\Pi_{\mathbb{H}}(T, m) := \max_{X \in \mathcal{X}^{T \times m}} |\mathbb{H}_{|X}|.$$



Note that  $\Pi_{\mathbb{H}}(T, m) \leq 2^{Tm}$  and if  $\Pi_{\mathbb{H}}(T, m) \leq 2^{Tm}$  we say that  $\mathbb{H}$  shatters  $\mathcal{X}^{T \times m}$ . For each  $n > 0$  define

$$\begin{aligned} d_{\mathbb{H}}(T) &:= \max_{m: \Pi_{\mathbb{H}}(T, m) = 2^{Tm}} m, \\ \bar{d}(\mathbb{H}) &:= \text{VCdim}(\mathbb{H}^1) = \text{VCdim}\left(\bigcup_{\mathcal{H} \in \mathbb{H}} \mathcal{H}\right), \\ \underline{d}(\mathbb{H}) &:= \max_{\mathcal{H} \in \mathbb{H}} \text{VCdim}(\mathcal{H}). \end{aligned}$$

Here,  $d_{\mathbb{H}}(T)$  is the generalized VC-dimension, and

$$d_{\mathbb{H}}(T) \geq \max\left(\left\lfloor \frac{\bar{d}(\mathbb{H})}{T} \right\rfloor, \underline{d}(\mathbb{H})\right)$$

where is trivial to see that  $\bar{d}(\mathbb{H}) \geq d_{\mathbb{H}}(T) \geq \underline{d}(\mathbb{H})$ . Now we can present the relevant result expressed in (Baxter, 2000, Corollary 13).

**Theorem 3.1.** *Given a sequence  $\mathbf{P} = (P_1, \dots, P_T)$  on  $(\mathcal{X} \times \{0, 1\})^T$ , and a sample  $\mathbf{z}$  from this distribution. Consider also a sequence  $\mathbf{h} = (h_1, \dots, h_T)$  of boolean hypothesis  $h_i \in \mathcal{H}$ , then*

$$R_Q(\mathbf{h}) \leq \hat{R}_{\mathbf{z}}(\mathbf{h}) + \epsilon,$$

with probability  $1 - \delta$  given that the number of samples per task

$$m \geq \frac{88}{\epsilon^2} \left[ 2d_{\mathbb{H}}(T) \log \frac{22}{\epsilon} + \frac{1}{T} \log \frac{4}{\delta} \right]. \quad (3.6)$$

Here, since  $d_{\mathbb{H}}(T) \geq d_{\mathbb{H}}(T+1)$ , it is easy to see that as the number of task  $T$  increases, the number of examples needed per task can decrease. Moreover, as shown in (Baxter, 2000, Theorem 14), if this bound on  $m$  is not fulfilled, then we can always find a sequence of distributions  $\mathbf{P}$  such that

$$\inf_{\mathbf{h} \in \mathcal{H}} \hat{R}_{\mathbf{z}}(\mathbf{h}) > \inf_{\mathbf{h} \in \mathcal{H}} R_{\mathbf{P}}(\mathbf{h}) + \epsilon.$$

With this results we can see that the condition (3.6) has some important properties:

- It is a computable bound, given that we know how to compute  $d_{\mathbb{H}}(T)$ .
- It provides a sufficient condition for the uniform convergence (in probability) of the empirical risk to the expected risk.
- It provides a necessary condition for the consistency of Multi-Task Learners, i.e. uniform convergence of the best empirical risk to the best expected risk.

## Conclusion

In Baxter (2000) several new concepts are developed. The  $(\mathcal{P}, Q)$ -environment of tasks is useful to characterize the concept of related tasks. Moreover, using this definition, Baxter is able to give some important results of uniform convergence in the Bias Learning paradigm. From this general view, Multi-Task Learning is a particular case and the uniform convergence results are also valid. The Feature Learning approach, which can be seen as a more particular method of Multi-Task Learning has some interesting results

splitting the analysis into the feature learning process and the construction of models over these features. Finally, the most important result is the definition of a generalized VC-dimension and the uniform convergence of Multi-Task Learning models using this concept. Although this is a result only valid for boolean hypothesis, it helps to shed some light on Multi-Task Learning and the reasons of its effectiveness.

### 3.2.2 Learning with Related Tasks

Using the work of [Baxter \(2000\)](#) as the foundation, several important notions and results are presented in [Ben-David and Borbely \(2008\)](#) for boolean hypothesis functions defined over  $\mathcal{X} \times \{0, 1\}$ . One of the main contributions of this work is a notion of task relatedness. In [Baxter \(2000\)](#) the tasks are related by sharing a common inductive bias that can be learned. In [Ben-David and Borbely \(2008\)](#) a precise mathematical definition for task relatedness is given. The other important contribution is the focus on the individual risk of each task. In [Baxter \(2000\)](#) all the results are given for the Multi-Task empirical and expected risks, which are an average of the risks of each task. However, bounding this average does not bound the risk of each particular task. This is specially relevant if we are in a Transfer Learning scenario, where there is a target task that we want to solve and the remaining tasks can be seen as an aid to improve the performance in the target one.

#### A Notion of Task Relatedness: $\mathcal{F}$ -Related Tasks

The main concept for the theory developed in [Ben-David and Borbely \(2008\)](#) is a set of  $\mathcal{F}$  of transformations  $f : \mathcal{X} \rightarrow \mathcal{X}$ . Given a probability distribution  $F$  over  $\mathcal{X} \times \{0, 1\}$ , a set of tasks with distributions  $P_1, \dots, P_T$  are  $\mathcal{F}$ -related if, for each task there exists some  $f_i \in \mathcal{F}$  such that  $P_i = f_i(F)$ .

**Definition 3.2** ( $\mathcal{F}$ -related task). Consider a measurable space  $(\mathcal{X}, \mathcal{A})$  and the corresponding measurable product space  $(\mathcal{X} \times \{0, 1\}, \mathcal{A} \times \wp(\{0, 1\}))$ . Consider  $P$  a probability distribution over this product space and a function  $f : \mathcal{X} \rightarrow \mathcal{X}$ , then we define the distribution  $f[P]$  such that for any  $S \in \mathcal{A}$ ,

$$f[P](S) := P(\{(f(x), b), (x, b) \in S\}).$$

Let  $\mathcal{F}$  be a set of transformations  $f : \mathcal{X} \rightarrow \mathcal{X}$ , and let  $P_1, P_2$  be distributions over  $(\mathcal{X} \times \{0, 1\}, \mathcal{A} \times \wp(\{0, 1\}))$ , then the distributions  $P_1, P_2$  are  $\mathcal{F}$ -related if  $f[P_1] = P_2$  or  $f[P_2] = P_1$  for some  $f \in \mathcal{F}$ .

This notion establishes a clear definition of related tasks but we are interested in how a learner can use this relatedness to improve the learning process. For that, considering that  $\mathcal{F}$  is a group under function composition, we regard at the action of the group  $\mathcal{F}$  over the set of hypothesis  $\mathcal{H}$ . This action defines the following equivalence relation in  $\mathcal{H}$ :

$$h_1 \sim_{\mathcal{F}} h_2 \iff \exists f \in \mathcal{F}, h_1 \circ f = h_2.$$

This equivalence relation defines equivalence classes  $[h]$ , that is let  $h' \in \mathcal{H}$  be an hypothesis, then  $h' \in [h]$  iff  $h' \sim_{\mathcal{F}} h$ . We consider the quotient space  $\mathcal{H}_{\mathcal{F}} := \mathcal{H} / \sim_{\mathcal{F}} = \{[h], h \in \mathcal{H}\}$ . This equivalence classes are useful to divide the learning process in two stages: a first one to determine the best equivalence class, and a second one to find the best representative hypothesis of such equivalence class.

For example, consider the handwritten digits recognition problem, we might integrate  $T$  different datasets designed in different conditions. Each dataset have been created using certain conditions of light and some specific scanner for getting the images. Even different pens or pencils might be influential in the stroke of the numbers. All these conditions are the  $\mathcal{F}$  transformations, and each  $f \in \mathcal{F}$  generate a different bias for the dataset. However, there exists a probability for "pure" digits, e.g. the pixels of digit one have higher probability around a line in the middle of the picture than in the sides. This "pure" probability distribution  $P_0$  and all the distributions  $P_1, \dots, P_T$  from which our datasets have been sampled might be  $\mathcal{F}$ -related among them and with  $P_0$ . If we first determine the  $\mathcal{F}$ -equivalent class of hypothesis  $[h]$  suited for digit recognition in the first stage, then it will be easier to select  $h_1, \dots, h_T \in [h]$  for each dataset in the second one.

### Bounds for $\mathcal{F}$ -Related Tasks

The results of Theorem 3.1 can be applied to the hypothesis space of equivalent classes  $\mathcal{H}_{\mathcal{F}}$ . However the following results is needed first. Let  $P_1, P_2$  be  $\mathcal{F}$ -related distributions, then this statement can be proved (Ben-David and Borbely, 2008, Lemma 2):

$$\inf_{h \in \mathcal{H}} R_{P_1}(h) = \inf_{h \in \mathcal{H}} R_{P_2}(h).$$

This indicates that the expected risk is invariant under transformations of  $\mathcal{F}$ . Now, one of the main results (Baxter, 2000, Theorem 2) can be given.

### Theorem 3.3.

#### Analysis of generalized VC-dimension

#### 3.2.3 Learning Under Privileged Information

### 3.3 Multi-Task Learning Methods: An Overview

#### 3.3.1 Bias Learning Approach

#### 3.3.2 Low-Rank Approach

#### 3.3.3 Learning Task Relations

#### 3.3.4 Decomposition Approach

### 3.4 Deep Multi-Task Learning

#### 3.4.1 Hard Parameter Sharing

#### 3.4.2 Soft Parameter Sharing

### 3.5 Multi-Task Learning with Kernel Methods

### 3.6 Conclusions

In this chapter, we covered. . .



# Chapter 4

## A Convex Formulation for Regularized Multi-Task Learning

### 4.1 Introduction

### 4.2 Convex Multi-Task Learning Support Vector Machines

#### 4.2.1 Convex Formulation

#### 4.2.2 L1 Support Vector Machine

#### 4.2.3 L2 Support Vector Machine

#### 4.2.4 LS Support Vector Machine

### 4.3 Optimal Convex Combination of trained models

### 4.4 Experiments

### 4.5 Conclusions

In this chapter, we have...



# Chapter 5

## Adaptive Graph Laplacian Multi-Task Support Vector Machine

### 5.1 Introduction

### 5.2 Graph Laplacian Multi-Task Support Vector Machine

### 5.3 Adaptive Graph Laplacian Algorithm

### 5.4 Experiments

### 5.5 Conclusions

In this chapter, we have...





# Bibliography

- Baxter, J. (2000). A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198.
- Ben-David, S. and Borbely, R. S. (2008). A notion of task relatedness yielding provable multiple-task learning guarantees. *Mach. Learn.*, 73(3):273–287.
- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.