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

Abbreviations

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

Chapter 3

Multi-Task Learning

This chapter presents...

3.1 Introduction

3.2 Why does Multi-Task Learning work?

3.2.1 Inductive Bias Learning Problem

Tipically in Machine Learning the goal is to find the best hypothesis $h(x, \alpha_0)$ from a space of hypothesis $\mathcal{H} = \{h(x, \alpha), \alpha \in \Lambda\}$, where Λ 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: $z := \{(x_i, y_i),\ i = 1, \ldots, n\}$ 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 $f(x) = p(y \mid x)$ 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 α 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* $\ell(\cdot, \cdot) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, and
- a *hypothesis space* $\mathcal{H} = \{h(x, \alpha), \alpha \in \Lambda \subset \mathbb{R}^m\}$ with hypothesis $h(\cdot, \alpha) : \mathcal{X} \to \mathcal{Y}$.

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 $z = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ 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:

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

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 α 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

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

where B is some non-decreasing function and $\mathrm{VCdim}(\mathcal{H})$ 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 $\text{VCdim}(\mathcal{H})$, or capacity, 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 consider $h(x) = h(x, \alpha)$, that is α completely defines h, we also substitute α by h for an easier formulation when necessary. 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* $\ell(\cdot, \cdot) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, 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

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

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

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

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

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

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

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:

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

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 $\hat{R}_{\boldsymbol{z}}(\delta)$ converges uniformly in probability to $R_Q(\delta)$.
- Defining a notion of capacity of hypothesis space families $\mathbb{H}$.
- Finding a bound of $\hat{R}_{\boldsymbol{z}}(\delta) - R_Q(\delta)$ for any δ using the capacity of the hypothesis space family. If possible, finding also a bound for $\inf_{\delta\in\Delta} \hat{R}_{\boldsymbol{z}}(\delta) - \inf_{\delta\in\Delta} R_Q(\delta)$.

To achieve these goals some previous definitions are needed.

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 $\boldsymbol{P} = (P_1, \ldots, P_T)$ sampled from $\mathcal{P}$ according the the distribution Q. Consider also the set of sequences of T hypothesis $\mathcal{H}^T := \{\boldsymbol{h} = (h_1, \ldots, h_T), h_1, \ldots, h_T \in \mathcal{H}\}$. We can define then the set of $(T, 1)$-empirical risks as

$$\mathcal{H}_\ell^T := \left\{ \boldsymbol{h}_\ell(x_1, y_1, \ldots, x_T, y_T) = \sum_{r=1}^{T} \ell(h(x_i), y_i), h_1, \ldots, h_T \in \mathcal{H} \right\}$$

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

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

for $\boldsymbol{h}_\ell, \boldsymbol{h}'_\ell \in \mathcal{H}_\ell, \mathcal{H}'_\ell$ 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 ϵ-cover is needed. Given a pseudo-metric d_S in a space $\mathcal{S}$, a set of l elements $s_1, \ldots, s_l \in \mathcal{S}$ is an ϵ-cover of $\mathcal{S}$ if $d_S(s, s_i) \leq \epsilon$ for some $i = 1, \ldots, l$. Let $\mathcal{N}(\epsilon, \mathcal{S}, d_S)$ denote the size of the smallest ϵ-cover. Then, we can define the following capacities of a family space $\mathbb{H}$:

- The *sample-driven capacity* $C\left(\epsilon, \mathbb{H}^T\right) := \sup_{\boldsymbol{P}} \mathcal{N}(\epsilon, \mathbb{H}^T, d_{\boldsymbol{P}})$.

- 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

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

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

$$T \geq O\left(\frac{1}{\epsilon^2} \log \frac{C\left(\frac{\epsilon}{32}, \mathbb{H}^*\right)}{\delta}\right), \; m \geq O\left(\frac{1}{T\epsilon^2} \log \frac{C\left(\frac{\epsilon}{32}, \mathbb{H}^T\right)}{\delta}\right).$$

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. As with $h(x)$ and α, since any $\mathcal{H}$ is defined by a $\delta \in \Delta$, we omit δ and write just $\mathcal{H}$ for simplicity.

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. The goal is not learning a hypothesis space $\mathcal{H}$ but a sequence of hypothesis $\boldsymbol{h} = (h_1, \ldots, h_T), h_1, \ldots, h_T \in \mathcal{H}$.

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

and the empirical risk is defined as

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

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

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

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

$$m \geq O\left(\frac{1}{T\epsilon^2} \log \frac{C\left(\frac{\epsilon}{16}, \mathbb{H}^T\right)}{\delta}\right).$$

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 $\mathcal{F} = \{f, f : \mathcal{X} \to \mathcal{V}\}$. Using these features, functions $g \in \mathcal{G}$ (which are tipically simple) are built: $\mathcal{X} \to_f \mathcal{V} \to_g \mathcal{Y}$. 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

$$C\left(\epsilon, \mathbb{H}^T\right) \leq C\left(\epsilon_1, \mathcal{G}^T\right)^T C_{\mathcal{G}_\ell}(\epsilon_2, \mathcal{F}),$$
$$C\left(\epsilon, \mathbb{H}^*\right) \leq C_{\mathcal{G}_\ell}(\epsilon, \mathcal{F})$$

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

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

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-Dimensions 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 $\hat{R}_{\boldsymbol{z}}(\boldsymbol{h}) - R_Q(\boldsymbol{h})$, 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 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

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

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

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

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

$$d_{\mathbb{H}}(T) := \max_{m:\Pi_{\mathbb{H}}(T,m)=2^{Tm}} m$$
$$\overline{d}(\mathbb{H}) :=$$
$$\underline{d}(\mathbb{H}) :=$$

3.2.2 A notion of Task relatedness

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.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.