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

February 2022

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 Multi-Task Learning Methods: An Overview

3.2.1 Feature-Based MTL

The feature-based methods try to find a set of features that are useful for all tasks. Two main approaches are taken: Feature Learning, which tries to learn new features from the original ones, and Feature Selection which selects a subset of the original features. Deep Learning is a particular case of Feature Learning that has had a tremendous relevance and that is treated separately.

Feature Learning approach

Apart from the Multi-Task feedforward Neural Network first shown in Caruana (1997), which will be described later, the first work of Multi-Task Feature Learning is presented in Argyriou et al. (2006). Argyriou et. al assume a multi-task linear model in some RKHS $\mathcal{H}$, where that the task parameters w_t lie in a linear subspace, i.e. $W = UA$ where w_t are the columns of W, U is an orthogonal matrix and A is a row-sparse matrix. The minimization problem is

$$\underset{U \in \mathbb{R}^{d \times d}, A \in \mathbb{R}^{d \times T}}{\arg\min} \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle U a_r, x_i^r \rangle) + \lambda \left\| A^\intercal \right\|_{2,1}^2 \text{ s.t. } U^\intercal U = I. \tag{3.1}$$

Here, the $L_{2,1}$ regularizer is used to impose row-sparsity across tasks, i.e. forcing some rows of A to be zero while the matrix U is restricted to be orthonormal. Although problem (3.1) is not jointly convex in U and A, in Argyriou et al. (2006) and Argyriou et al. (2008) is shown to be equivalent to the convex problem

$$\underset{W \in \mathbb{R}^{d \times T}, D \in \mathbb{R}^{d \times d}}{\arg\min} \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r \rangle) + \lambda \sum_{r=1}^{T} \langle w_r, D^{-1} w_r \rangle + \lambda\mu \operatorname{tr}\left(D^{-1}\right) \tag{3.2}$$
$$\text{s.t. } D \succeq 0, \ \operatorname{tr}(D) \leq 1.$$

Here D^+ denotes the pseudo-inverse of D and $D \succeq 0$ restricts the D to the semidefinite positive matrices and the trace of D^{-1} is added for numerical stability and convenient convergence properties. Problem (3.2) is solved using an alternating two-step optimization algorithm in W and D. When $\mu = 0$ in (3.2) the optimization problem can be reformulated as

$$\operatorname*{arg\,min}_{W \in \mathbb{R}^{d \times T}} \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r \rangle) + \lambda \left\| W \right\|_*^2 . \tag{3.3}$$

Here, $\left\| W \right\|_* = \operatorname{tr}\left((WW^\intercal)^{\frac{1}{2}} \right)$ denotes the trace norm (or nuclear norm), which can be seen as the convex envelope of the rank, thus favouring low-rank solutions of W. In Maurer (2009) some bounds on the excess risks are given for this Multi-Task Feature Learning method. In Argyriou et al. (2007) this idea is extended to any spectral funcion $F : \mathbb{S}_{++}^d \to \mathbb{S}_{++}^d$ where $\mathbb{S}_{++}^d$ is the set of matrices $A \in \mathbb{R}^{d \times d}$ symmetric and positive definite. The definition for the spectral function $F(D)$, where we can write $D = U^\intercal \operatorname{diag}(\lambda_1, \dots, \lambda_d) U$ is: $F(D) = U^\intercal \operatorname{diag}(f(\lambda_1), \dots, f(\lambda_d)) U$. Then, a generalized regularizer for problem (3.2) can be expressed as

$$\sum_t \langle w_t, F(D) w_t \rangle = \operatorname{tr}\left(F(D) W W^\intercal \right) .$$

It is easy to see that problem (3.2) is a particular case where $f(\lambda) = \lambda^{-1}$. Another relevant extension is shown in Agarwal et al. (2010), where instead of assuming that the task parameters w_r lie in a linear subspace, the authors generalize this idea by assuming that w_r lies in a manifold $\mathcal{M} \in \mathcal{H}$.

$$\operatorname*{arg\,min}_{W, \mathcal{M}, \boldsymbol{b}} \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r \rangle + b_r) + \lambda \sum_{r=1}^{T} \mathcal{P}_{\mathcal{M}}(w_r), \tag{3.4}$$

where $\mathcal{P}_{\mathcal{M}}(w_t)$ represents the distance between w_t and its projection on the manifold $\mathcal{M}$. Again, an approximation of (3.4) is used to obtain a convex problem and it is solved using a two-step optimization algorithm.

Other distinct, relevant approach for Feature Learning is the one described in Maurer et al. (2013), where a sparse-coding method, Maurer and Pontil (2010), is used for MTL. Maurer et al. present the problem

$$\operatorname*{arg\,min}_{D \in \mathcal{D}_k, A \in \mathbb{R}^{k \times T}} \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle D a_r, x_i^r \rangle) + \lambda \left\| D \right\|_{2,\infty} + \mu \left\| A \right\|_{1,\infty} . \tag{3.5}$$

Here, $\mathcal{D}_k$ is the set of k-dimensional dictionaries and every $D \in \mathcal{D}_k$ is a linear map $D : \mathbb{R}^k \to \mathcal{H}$; in the linear case, where $\mathcal{H} = \mathbb{R}^d$, the set $\mathcal{D}_k$ is the set of matrices $\mathbb{R}^{d \times k}$, such that

$$\underset{d \times T}{W} = \underset{d \times k}{D} \underset{k \times T}{A} .$$

Although (3.1) and (3.5) share a similar form, there are crucial differences. The matrix U in (3.1) is an orthogonal square matrix, while the matrix D of (3.5) is overcomplete with $k > d$ columns of bounded norm. A problem very similar to (3.5) is presented in Kumar and III (2012) where the idea is the same but the regularizers are the $L_{2,2}$ (Frobenius) norm for D and the $L_{1,1}$ norm for A:

$$\underset{D\in\mathcal{D}_k, A\in\mathbb{R}^{k\times T}}{\arg\min} \sum_{r=1}^{T}\sum_{i=1}^{m_r} \ell(y_i^r, \langle Da_r, x_i^r\rangle) + \lambda \left\|D\right\|_{2,2} + \mu \left\|A\right\|_{1,1} . \tag{3.6}$$

Feature Selection approach

The feature selection is also driven by learning a good set of features for all tasks, however it focuses on subsets of the original features. This is a more rigid approach than that of Feature Learning but is also more interpretable.

Most works on Multi-Task Feature Selection uses an $L_{p,q}$ regularization of the weights matrix W. The first work Obozinski et al. (2006) solves the problem

$$\underset{W}{\arg\min} \sum_{r=1}^{T}\sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r\rangle) + \lambda \left\|W^\intercal\right\|_{2,1}^2 , \tag{3.7}$$

where the $L_{2,1}$ regularization enforces row sparsity and forces different tasks to share some features W^i, which are the rows of W. In Liu et al. (2009) the $L_{\infty,1}$ regularization is used for the same goal. Then, in Gong et al. (2012a) this idea is generalized with a capped-$L_{p,1}$ penalty of $W^\intercal$, which is defined as $\sum_{i=1}^{d} min(\theta, \left\|W^i\right\|_p)$. That is, the parameter θ enables a more flexible regularization, with small values of θ many rows should be zero, since only the small rows will be fully penalized and the rows with the largest norm will not dominate the sum, as θ grows this penalty will degenerate to the standard $L_{p,1}$ norm.

To allow more flexibility, a multi-level selection is presented in Lozano and Swirszcz (2012). The main idea is to decompose each w_r^i, that is, the i-th feature of the r-th task as $w_r^i = \theta^i \gamma_r^i$ and using an L_1 regularizer on the vector $\boldsymbol{\theta}$ and the vectors $\boldsymbol{\gamma}^i$, which are the concatenations of θ^i and γ_r^i for all features and all tasks, respectively. By doing this, the features i such that $\theta^i = 0$, are discarded for all tasks, but the rest may be shared among tasks or not depending on γ_r^i.

The feature selection methods based on $L_{p,1}$ regularization are shown to be equivalent to a Bayesian approximation with a generalized Gaussian prior in Zhang et al. (2010). Moreover, this approach also allows to find the relationship among tasks and to identify outliers. In Hernández-Lobato and Hernández-Lobato (2013) a horseshoe prior is used instead to learn feature covariance; and Hernández-Lobato et al. (2015) this prior is also used to identify outlier tasks.

Deep Learning approach

3.2.2 Parameter-Based MTL

The parameter-based MTL does not focus on shared sets of features across tasks, instead other types of dependencies among the task (the task-parameters w_r specifically) are taken into account. Some approaches rely on the assumption that the Multi-Task weight matrix W has a low rank, others try to learn the pairwise task relations or to cluster the tasks. A different approach is the decomposition one, where the assumption is that the matrix W can be expressed as the summation of multiple matrices.

Low-Rank approach

In the low-rank approach, the assumption is that the task parameters w_r share a low-dimensional space, or are close to this subspace. This is similar to the Feature Learning approach, but it is not that rigid, since it allows for some flexibility. The idea in Ando and Zhang (2005) is that the task parameters can be decomposed as

$$w_r = u_r + \Theta^\intercal v_r,$$

where $\Theta \in \mathbb{R}^{k\times d}$ spans a shared low dimensional space, that is $\Theta^\intercal\Theta = I_k$ with $k < d$, and d is the dimension of the data. Under this consideration, the proposed model is

$$\underset{\Theta\in\mathbb{R}^{k\times d},\boldsymbol{u},\boldsymbol{v}}{\arg\min} \sum_{r=1}^{T}\sum_{i=1}^{m_r} \ell(y_i^r, \langle u_r + \Theta^\intercal v_r, x_i^r\rangle) + \lambda\sum_{r=1}^{T}\|u_r\|^2 \text{ s.t. } \Theta\Theta^\intercal = I_k. \tag{3.8}$$

Observe that this problem shares some similarities with (3.1). However, this is a more flexible approach, since the vectors u_r allow for deviations of the task parameters from the shared subspace. Problem (3.8) is solved using a two-step optimization iterating between minimizing in $\{\Theta, \boldsymbol{v}\}$ and minimizing in $\boldsymbol{u}$. However problem (3.8) is not convex. Note that (3.8) can be reformulated as

$$\underset{\Theta\in\mathbb{R}^{k\times d},\boldsymbol{w},\boldsymbol{v}}{\arg\min} \sum_{r=1}^{T}\sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r\rangle) + \lambda\sum_{r=1}^{T}\|w_r - \Theta^\intercal v_r\|^2 \text{ s.t. } \Theta\Theta^\intercal = I_k, \tag{3.9}$$

where the terms $\|w_r - \Theta^\intercal v_r\|^2$ enforces the similarity across tasks by binging them closer to the shared subspace. In Chen et al. (2009) the following extension is proposed:

$$\underset{\Theta\in\mathbb{R}^{k\times d},\boldsymbol{w},\boldsymbol{v}}{\arg\min} \sum_{r=1}^{T}\sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r\rangle) + \lambda\sum_{r=1}^{T}\|w_r - \Theta^\intercal v_r\|^2 + \mu\sum_{r=1}^{T}\|w_r\|^2 \text{ s.t. } \Theta\Theta^\intercal = I_k. \tag{3.10}$$

And it is shown that (3.10), when relaxing the orthogonality constraint, can be expressed as a convex minimization problem. Note that the difference between (3.9) and (3.10) is the addition of the Frobenius norm of matrix W, also known as trace norm. This norm penalizes $\sum_{i=1}^{d}\lambda_i(W)^2$, where $\lambda_i(W)$ is an eigenvector of W, and thus, forcing W to be low-rank. In the work of Pong et al. (2010) new formulations for problems with this trace norm penalty and a primal-dual method for solving the problem is developed. A modification of the trace norm can be found in Han and Zhang (2016), where a capped-Frobenius norm is defined as $\sum_{i=1}^{d}\min(\theta, \lambda_i(W)^2)$. This capped norm, as in the capped-$L_{p,q}$ norm, can enforce a lower rank matrix for small θ and also degenerates to the trace norm for large enough θ.

Task-Relation Learning approach

In other approaches, like the Feature Learning approach or the Low-Rank approach, the assumption is that all task parameters share the same subspace, which may be detrimental when there exists a negative or neutral transfer. The Task-Relation Learning approach aims to find the pairwise dependencies among tasks and to possibly model positive, neutral and negative transfers between tasks.

One of the first works with the goal of explicitly modelling the pairwise task-relations is Bonilla et al. (2007), where a Multi-Task Gaussian Process (GP) formulation is presented. Assuming a Gaussian noise model $y_i^r \sim N\left(f_r(x_i^r), \sigma_r^2\right)$, Bonilla et al. place a GP prior over the latent functions $\boldsymbol{f}_r$ to induce correlation among tasks:

$$\boldsymbol{f} \sim N\left(\mathbf{0}_{dT}, K^f \otimes K_\theta^{\mathcal{X}}\right)$$

where $\boldsymbol{f} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_T)$ and $A \otimes B$ is the Kronecker product of two matrices, so $\operatorname{Cov}(f_r(x), f_s(x')) = K_{rs}^f k_\theta^{\mathcal{X}}(x, x')$. That is, instead of assuming a block-diagonal covariance matrix for $\boldsymbol{f}$ as in previous works of Joint Learning Lawrence and Platt (2004), the authors in Bonilla et al. (2007) model the covariance as a product of inter-task covariance and inter-feature covariance. The inference of this model can be done using the standard GP inference for the mean and the variance of the prediction distribution. The mean prediction for a new data point x_* of task s is:

$$f_s(x_*) = (K_{:,l}^f \otimes K^{\mathcal{X}}(:, x_*))^\intercal \Sigma^{-1} \boldsymbol{y},\ \Sigma = K^f \otimes K_\theta^{\mathcal{X}} + \operatorname{diag}(\sigma_1, \ldots, \sigma_T) \otimes I_n.$$

However the interest resides in learning the task-covariance matrix K^f, but this leads to a non-convex problem. The authors propose a low-rank approximation of K^f, which weaken its expressive power. To overcome this disadvantage, in Zhang and Yeung (2010, 2013), using the idea of the Multi-Task GP they consider linear models $f(x_i^r) = \langle w_r, x_i^r \rangle + b_r$ and the prior on matrix $W = (w_1 \ldots, w_T)$ is defined as

$$W|\sigma_r \sim \left(\prod_{r=1}^{T} N\left(\mathbf{0}_d, \theta_i^2 I_d\right)\right) MN\left(0_{d\times m}, I_d \otimes \Omega\right)$$

where $MN\left(0, A \otimes B\right)$ denotes the matrix-variate normal distribution with mean M, row covariance matrix A and column covariance matrix B. It is shown that the problem of selecting the maximum a posteriori estimation of W and the maximum Likelihood estimations of Ω and $\boldsymbol{b}$ has a regularized minimization problem that, when relaxing the restrictions on Ω, can be expressed as

$$\begin{aligned} &\operatorname*{arg\,min}_{\Omega \in \mathbb{R}^{d\times T}, W, \boldsymbol{b}} \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r \rangle + b_r) + \lambda \sum_{r=1}^{T} \|w_r\|^2 + \mu \sum_{r,s=1}^{T} \left(\Omega^{-1}\right)_{rs} \|w_r - w_s\|^2 \\ &\text{s.t. } \Omega \succeq 0, \operatorname{tr} \Omega = 1. \end{aligned} \tag{3.11}$$

This is a convex formulation and a two-step procedure is developed to find the solution.

Other approaches like Argyriou et al. (2013) reach a similar problem from other perspective. Argyriou et al. assume a representation of the structure of the tasks as a graph, then the graph Laplacian in the optimization problem can incorporate the knowledge about the task structure as shown in Evgeniou et al. (2005):

$$\operatorname*{arg\,min}_{W, \boldsymbol{b}} \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r \rangle + b_r) + \mu \sum_{r,s=1}^{T} \left(L^+\right)_{rs} \langle w_r, w_s \rangle. \tag{3.12}$$

Here the regularization can be also written using the adjacency matrix A as

$$\sum_{r,s=1}^{T} \left(L^+\right)_{rs} \langle w_r, w_s \rangle = \sum_{r,s=1}^{T} \left(A^+\right)_{rs} \|w_r - w_s\|^2,$$

and μ is a tuning parameter. The goal of Argyriou et al. (2013) is to jointly learn the task parameters and the graph Laplacian L. For that, they add the penalty some small perturbation to the diagonal of L so they obtain a convenient positive kernel; and then, according to Evgeniou et al. (2005), the problem (3.12) can be solved in the dual space using

$$k_L(x_i^r, x_j^s) = (L + \lambda I)^{-1}_{rs}\, k(x_i^r, x_j^s).$$

Argyriou et al. propose the joint learning in this dual space by solving

$$\begin{aligned} &\arg\min_{\boldsymbol{\alpha}, L} \boldsymbol{\alpha}^\intercal K_L \boldsymbol{\alpha} + \nu \boldsymbol{\alpha}^\intercal \boldsymbol{y} + \operatorname{tr}(L + \lambda I)^{-1} \\ &\text{s.t. } 0 \preceq (L + \lambda I)^{-1} \preceq \frac{1}{\lambda} I,\ (L + \lambda I)^{-1}_{\text{off}} \leq 0,\ (L + \lambda I)^{-1} \mathbf{1}_n = \frac{1}{\lambda} \mathbf{1}_n, \end{aligned} \tag{3.13}$$

where A_{off} denotes the off diagonal entries of A. The restrictions of problem (3.13) are to ensure that L is a valid graph Laplacian, and the trace term in the objective function is to force L to be low-rank so it is easier to find clusters of tasks. The objective function is not jointly convex but is convex in each $\boldsymbol{\alpha}$ and L when we fix the other. The step to optimize the Laplacian matrix requires two projection steps using proximal operators. Another work focused on learning the task-relations is Dinuzzo (2013), where the approach the problem a learning problem in an RKHS of vector-valued functions $g : \mathcal{X} \to \mathbb{R}^T$, where the associated reproducing kernel is:

$$H(x_1, x_2) = K_{\mathcal{X}}(x_1, x_2) \cdot L$$

and L is a symmetric positive matrix called *output* kernel.

In Ciliberto et al. (2015) some results are developed for the convergence of alternating minimization algorithms in convex problems as that used in Zhang and Yeung (2013). However, since the problems presented in Argyriou et al. (2013); Dinuzzo (2013) are not convex, this result of convergence does not hold.

Task Clustering approach

The Task Clustering approach tries to find K clusters or groups among the original set of T tasks. Usually, the goal is to learn jointly only the tasks in the same cluster, so no negative transfer takes place. The first clustering approach Thrun and O'Sullivan (1996) divides the optimization process in two separate steps: independently learning the task-parameters and jointly learning the clusters of tasks. Using models that involves distances among points, e.g. kernel methods, we define for each task the distance

$$\operatorname{dist}_{\phi_r}(x, x') = \sqrt{\sum_{i=1}^{d} \phi_r^i \left(x^i - x'^i\right)^2}.$$

That is, $\boldsymbol{\phi}_r$ parametrizes a distance with a different weight for each feature. Then, for each task $r = 1, \ldots, T$ an optimal $\boldsymbol{\phi}_r^*$ is computed minimizing the distance between examples of the same class and maximizing the distance among different classes:

$$A_r(\phi) = \sum_{i,j}^{m_r} (y_i^r y_j^r) \operatorname{dist}_\phi(x_i^r, x_j^r).$$

Then, the procedure is the following. First the empirical loss on task r of a model fitted on data of task r using a distance parametrized by ϕ_s^* is defined as e_{rs}. Then, for $K = 1, 2, \dots, T$, the following functional is minimized

$$J(K) = \sum_{\kappa=1}^{K} \sum_{r \in B_\kappa} \frac{1}{|B_r|} \sum_{s \in B_\kappa} e_{r,s},$$

and the number of clusters K with minimum $J(K)$ is selected. That is, the clusters are selected using the results of independently trained tasks and the transfer is done through the parameters ϕ.

A different approach is the Bayesian proposal of Bakker and Heskes (2003) where a Multi-Task Neural Network Caruana (1997) with one shared hidden layer is used. The task-dependent parameters can be modeled together using a prior that is a mixture of K Gaussians $\boldsymbol{a}_r \sim \sum_{\kappa=1}^{K} \alpha_\kappa N\left(\boldsymbol{\mu}_\kappa, \Sigma_\kappa\right)$; and by learning the variables α_i using Bayesian inference, the tasks can be clustered. In this model, unlike in Thrun and O'Sullivan (1996) the clusters and task parameters are jointly learned.

From the regularization approach some proposals have also been made. In Jacob et al. (2008) a problem based on Evgeniou and Pontil (2004) is proposed. Considering $U = \frac{1}{T}\mathbf{1}\mathbf{1}^\intercal$, E the $T \times K$ cluster assignment binary matrix, and defining the adjacency matrix $M = E(E^\intercal E)^{-1} E^\intercal$ the problem is

$$\arg\min_{W} \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r \rangle) + \lambda(\mu_{\mathrm{m}} \Omega_{\mathrm{m}}(W) + \mu_{\mathrm{b}} \Omega_{\mathrm{b}}(W) + \mu_{\mathrm{w}} \Omega_{\mathrm{w}}(W)), \tag{3.14}$$

where $\Omega_{\mathrm{m}} = \operatorname{tr}(WUW^\intercal)$ is the mean regularization, $\Omega_{\mathrm{b}} = \operatorname{tr}(W(M-U)W^\intercal)$ is the inter-cluster variance regularization and $\Omega_{\mathrm{w}} = \operatorname{tr}(W(I-M)W^\intercal)$ is the intra-cluster variance regularization. This problem cannot be solved using the results of Evgeniou et al. (2005) because the regularization used is not convex, so a convex relaxation is needed.

A similar approach is presented in Kang et al. (2011) where, using the results from Argyriou et al. (2008), they propose a trace norm regularizer of the matrices $W_\kappa = WQ_\kappa$, where Q_κ are diagonal matrices and the r-th element of the diagonal indicates if task r corresponds to cluster κ. They consider the problem

$$\begin{aligned} \arg\min_{W, Q_1, \dots, Q_T, \boldsymbol{b}} \quad & \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle w_r, x_i^r \rangle + b_r) + \lambda \sum_{\kappa=1}^{K} \|WQ_\kappa\|_*^2 \\ \text{s.t.} \quad & \sum_{\kappa=1}^{K} Q_\kappa = I \text{ with } 0 \le Q_{\kappa t} \le 1 \end{aligned} \tag{3.15}$$

This can be seen as a clusterized version of Multi-Task Feature Learning Argyriou et al. (2006), that is, instead of assuming that all tasks share the same subspace, only the tasks in the same cluster do, however the explicit learned features cannot be recovered.

Another approximation to clusterized MTL is provided in Crammer and Mansour (2012), where a two-step procedure if described as follows. Considering that K initial clusters are fixed containing the T tasks, then two steps are repeated: First K single task models f_κ are fitted using the pooled data from tasks in cluster κ. Secondly each task r is assigned to the cluster κ whose function f_κ obtains the lowest error in task r. The proposal of Barzilai and Crammer (2015) takes the idea of the cluster assignation step from Crammer and Mansour (2012) and is also inspired by the work of Kumar and III (2012). In this work a model is presented where $W = DA$, where $D \in \mathbb{R}^{d\times K}$ contains as columns the hypothesis for each cluster and $G \in \mathbb{R}^{K\times T}$ is the task assignment matrix, that is $g_r = G_r \in \{0,1\}$ and $\|g_r\|^2 = 1$. The corresponding optimization problem is

$$\begin{aligned} &\underset{D\in\mathbb{R}^{d\times K}, G\in\mathbb{R}^{K\times T}}{\arg\min} \sum_{r=1}^{T}\sum_{i=1}^{m_r} \ell(y_i^r, \langle Dg_r, x_i^r\rangle) + \lambda \|D\|_{2,1} \\ &\text{s.t. } g_r \in [0,1],\ \|g_r\|^2 = 1, \end{aligned} \tag{3.16}$$

where the constraints on g_r have been relaxed to be in the $[0,1]$ interval in order to make problem (3.16) convex. Observe that (3.16) is similar to (3.6) but different restrictions are used to ensure that G can be seen as a clustering assignment matrix.

Decomposition approach

The Decomposition approach considers that assuming that the task parameters resides in the same subspace or that the parameter matrix W is too restrictive for real world scenarios. The proposition is then to decompose the parameter matrix in the sum of two matrices, i.e. $W = U + V$ where usually U captures the shared properties of the tasks and V accounts for the information that cannot be shared among tasks. This models also receive the name of *dirty models* because they assume that the data is *dirty* and does not place perfectly in constrained subspaces. The optimization problem is

$$\underset{U,V\in\mathbb{R}^{d\times T}}{\arg\min} \sum_{r=1}^{T}\sum_{i=1}^{m_r} \ell(y_i^r, \langle u_r + v_r, x_i^r\rangle) + \lambda g(U) + \mu h(V), \tag{3.17}$$

where $g(U)$ and $h(V)$ are different regularizers for U and V, respectively. In Jalali et al. (2010) $g(U) = \|U^\intercal\|_{1,\infty}$ to enforce block-sparsity and $h(V) = \|V\|_{1,1}$ to enforce element-sparsity. In Chen et al. (2010) $g(U) = \|U\|_{2,2}$ to enforce low-rank while maintaining $h(V) = \|V\|_{1,1}$. In Chen et al. (2011) both regularizers seek properties shared among all tasks, $g(U) = \|U\|_{2,2}$ to enforce a low-rank and $h(V) = \|V^\intercal\|_{1,2}$ for row-sparsity. In Gong et al. (2012b) they propose $g(U) = \|U^\intercal\|_{1,2}$ to enforce row-sparsity, i.e. the tasks share a common subspace; and $h(V) = \|V\|_{1,2}$ which penalizes the orthogonal parts to the common subspace of task-parameter, the authors state that it penalizes outlier tasks.

Other approaches generalize the decomposition method by assuming that the parameter matrix can be expressed as $W = \sum_{l=1}^{L} W_l$, then the problem to solve has the form

$$\underset{W_1,\ldots,W_L\in\mathbb{R}^{d\times T}}{\arg\min} \sum_{r=1}^{T}\sum_{i=1}^{m_r} \ell(y_i^r, \langle \sum_{l=1}^{L} (W_l)_r, x_i^r\rangle) + \sum_{l=1}^{L} \lambda_l r(W_l), \tag{3.18}$$

In Zweig and Weinshall (2013) the regularizer used is $r(W_l) = \|W_l^\intercal\|_{2,1} + \|W_l\|_{1,1}$ to enforce the row and element-sparsity. In Han and Zhang (2015) the regularizer is $r(W_l) = \sum_{r,s=1}^{T} \|(W_l)_r - (W_l)_s\|^2$, which alongside some constraints it allows to build a tree of task groups, where the root contains all the tasks and the leafs only correspond to one task.

3.2.3 Joint MTL

The Joint Learning methods for MTL can be divided into the frequentist and the Bayesian approaches. The frequentist approaches uses a combination of task-specific models and models that are common to all tasks. These two models are learned simultaneously with the goal of leveraging the common and specific information. In the Bayesian approaches common prior is shared for all the tasks models, and the diverse sources of data define different posterior distributions for each task.

Bayesian Approach

The work of Lawrence and Platt (2004) presents a GP model where all the tasks share a common prior. That is, given T tasks, a noise model with latent variables $\boldsymbol{f}_r$ is considered, i.e. $y_i^r = f_i^r + \epsilon$, and $\boldsymbol{f}_r$ follows a GP prior

$$p(\boldsymbol{f}_r|X,\boldsymbol{\theta}) = N\left(0, K_{\boldsymbol{\theta}}\right)$$

where K is a kernel matrix parametrized by $\boldsymbol{\theta}$ and evaluated at the points X. Note that a single $\boldsymbol{\theta}$ parameter is used to model a prior shared for all tasks. Using this idea the posterior probability can be expressed as

$$p(\boldsymbol{y}^1,\ldots,\boldsymbol{y}^T|\boldsymbol{f}_1,\ldots,\boldsymbol{f}_T,\boldsymbol{\theta},X^1,\ldots,X^T) \propto p(\boldsymbol{f}_1,\ldots,\boldsymbol{f}_T,\boldsymbol{\theta}|X^1,\ldots,X^T)\prod_{r=1}^{T} p(\boldsymbol{y}^r|\boldsymbol{f}_r,\boldsymbol{\theta})$$

where it is assumed the distribution for the latent parameters factorizes as

$$p(\boldsymbol{f}_1,\ldots,\boldsymbol{f}_T,\boldsymbol{\theta}|X^1,\ldots,X^T) \propto \prod_{r=1}^{T} p(\boldsymbol{f}_r|X^r,\boldsymbol{\theta}),$$

where $p(\boldsymbol{f}_r|X^r,\boldsymbol{\theta}) = N\left(0, K_\theta\right)$, that is $\text{Cov}(f^r(x), f^r(x')) = k_\theta(x,x')$. Although this idea is interesting for MTL, it presents a rigid framework since we use a fixed model for the prior $p(\boldsymbol{f}_r|X,\boldsymbol{\theta})$. To use Bayesian induction over the prior too, the hierarchical Bayesian model is considered. That is, we consider a different prior $p(\boldsymbol{f}_r|X,\boldsymbol{\theta}^r)$ for each task and a hyperprior for $\boldsymbol{\theta}^r$, $p(\boldsymbol{\theta}^r|\boldsymbol{\phi})$. Then, the distribution for latent parameters is expressed as

$$p(\boldsymbol{f}_1,\ldots,\boldsymbol{f}_T,\boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T,\boldsymbol{\phi}|X^1,\ldots,X^T) \propto \prod_{r=1}^{T} p(\boldsymbol{f}_r|X^r,\boldsymbol{\theta}^r,\boldsymbol{\phi})p(\boldsymbol{\theta}^r|\boldsymbol{\phi}).$$

In Yu et al. (2005), a Gaussian hyperprior $p(\boldsymbol{\theta}^r|\boldsymbol{\phi})$ is considered. Note that in this formulation, each task parameter $\boldsymbol{\theta}^r$ is an independent from the rest of parameters $\boldsymbol{\theta}^s, s \neq r$ given $\boldsymbol{\phi}$. That is, the joint distribution factorizes as

$$p(\boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T|\boldsymbol{\phi}) = \prod_{r=1}^{T} p(\boldsymbol{\theta}^r|\boldsymbol{\phi}).$$

Then, in Xue et al. (2007) a Dirichlet Process Prior is considered for modelling the task parameters. An explicit dependence is then defined over the task parameters

$$p(\boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T|\boldsymbol{\phi}) = \prod_{r=1}^{T} p(\boldsymbol{\theta}^r|\boldsymbol{\theta}^{-r},\boldsymbol{\phi}).$$

where $\boldsymbol{\theta}^{-r} = \{\boldsymbol{\theta}^s, s \neq r\}$. This formulation converts this model in a task-clustering approach, where the clusters are learned jointly with the task parameters $\boldsymbol{\theta}$. Following this approach of hierarchical Bayes, III (2009) uses a prior for the task parameters $\boldsymbol{\theta}^r$ that learns backwards a genealogy tree. That is, beginning at the leafs, which are the task parameters $\boldsymbol{\theta}^r$, the branches merge until a common root to all the tasks. Thus, by selecting different thresholds or levels of this tree, we can obtain different clusters.

Frequentist Approach

The first proposal of the frequentist approach, which uses the SVM as the foundation, is found in Evgeniou and Pontil (2004) where the *regularized MTL* SVM is presented. The goal is to find a decision function for each task, each being defined by a vector

$$w_r = w + v_r,$$

where w is common to all tasks and v_r is task-specific. Instead of imposing some restrictions such as low-rank or inter-task regularization the idea is to impose the coupling by directly placing a model w that is common to all tasks. The v_r part is added so each model can be adapted to the specific task.

Multiple extensions of the work of Evgeniou and Pontil (2004) have been presented: in Li et al. (2015); Xu et al. (2014) the method is extended to the Proximal SVM Fung and Mangasarian (2001) and Least Squares SVM Suykens and Vandewalle (1999), respectively. Also, in Parameswaran and Weinberger (2010) the idea is adapted for the Large Margin Nearest Neighbor model Weinberger and Saul (2009). However, in this work we are interested mainly in two extensions: one is the work of Evgeniou et al. (2005) and the other is developed in Cai and Cherkassky (2009); Liang and Cherkassky (2008), both will be described in detail in Section 3.3. Using the unified formulation we can express the more general primal problem as

$$\begin{array}{ll}
\underset{w, b, v_r, b_r, \xi_i^r}{\arg\min} & C \sum_{r=1}^{T} \sum_{i=1}^{m_r} \xi_i^r + \frac{1}{2} \langle w, w \rangle + \frac{\mu}{2} \sum_{r=1}^{T} \langle v_r, v_r \rangle \\
\text{s.t.} & y_i(\langle w, \phi(x_i^r) \rangle + b + \langle v_r, \phi_r(x_i^r) \rangle + b_r) \geq p_i^r - \xi_i^r, \\
 & \xi_i^r \geq 0, \\
\text{for} & r = 1, \dots, T;\ i = 1, \dots, m_r.
\end{array}$$

where μ is a tradeoff parameter to leverage the common and task-specific information.

3.2.4 Multi-Task Problems

TODO: explicación de tabla

TABLE 3.1: A sample long table.

Name	n	d	T	Goal	References
school	15362	27	139	single-reg	Evgeniou and Pontil (2004) Evgeniou et al. (2005) Argyriou et al. (2006, 2008, 2007) Bonilla et al. (2007) Zhang and Yeung (2010) Agarwal et al. (2010) Chen et al. (2011) Zhou et al. (2011) Gong et al. (2012b) Kumar and III (2012) Zhang and Yeung (2013) Han and Zhang (2016) Jeong and Jun (2018)
20-newsgroup	18000	t.v.	20	multi-clas	Ando and Zhang (2005) III (2009)
Reuters-RCV1	800000	t.v.	103	multi-clas	Yu et al. (2005) Ando and Zhang (2005)
computer	3600	13	180	single-reg	Argyriou et al. (2006, 2008, 2007) Evgeniou et al. (2005) Agarwal et al. (2010) Kumar and III (2012) Jeong and Jun (2018)
landmine	14820	10	29	bin-clas	Xue et al. (2007) Jebara (2011) III (2009) Jawanpuria and Nath (2012)
MHC-I	32302	184	47	bin-clas	Jacob et al. (2008) Jawanpuria and Nath (2012)
dermatology	366	33	6	multi-clas	Jebara (2004) Argyriou et al. (2008)
sentiment	2000	473856	4	bin-clas	III (2009) Zhang and Yeung (2010) Crammer and Mansour (2012) Zhang and Yeung (2013) Barzilai and Crammer (2015)

Continued on next page

Table 3.1 – continued from previous page

Name	n	d	T	Goal	References
sarcos	44484	21	7	multi-reg	Zhang and Yeung (2010) Chen et al. (2011) Zhou et al. (2011) Jawanpuria and Nath (2012) Zhang and Yeung (2013) Ciliberto et al. (2015)
isolet	7797	617	5	MT multi-clas	Parameswaran and Weinberger (2010) Gong et al. (2012a)
mnist	70000	400	10	multi-clas	Kang et al. (2011) Kumar and III (2012) Zweig and Weinshall (2013) Jeong and Jun (2018)
usps	9298	256	10	multi-clas	Kang et al. (2011) Kumar and III (2012) Zweig and Weinshall (2013) Jeong and Jun (2018)
adni	675	306	6	single-reg	Gong et al. (2012a) Gong et al. (2012b)
microarray	131	21	19	multi-reg	Lozano and Swirszcz (2012) Han and Zhang (2016)
cifar10	50000	1024	10	multi-clas	Zweig and Weinshall (2013) Han and Zhang (2016)
parkinson	5875	19	42	single-reg	Jawanpuria and Nath (2012) Jeong and Jun (2018)

3.3 Multi-Task Learning with Kernel Methods

The Multi-Task Learning paradigm can be seen as learning a vector-valued function $f : \mathcal{X} \to \mathbb{R}^T$, where each element of the vector corresponds to a different task.

3.3.1 Operator-Valued Kernel Hilbert Spaces

Consider $\mathcal{Y}$ a Hilbert space with inner product $(.,.)$, then we can study the Hilbert spaces of functions with values in $\mathcal{Y}$. The kernels in such spaces are operator-valued functions $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$. We look at these spaces from three different and equivalent perspectives: continuous evaluation functionals, positive-definite kernels and feature maps.

Continuous evaluation functionals. The first approach considered is that of continuous evaluation functionals. That is, given a Hilbert space with continuous evaluation functionals, we can find the reproducing kernel operator. Consider the vector-valued Hilbert space $\mathcal{H}$ of functions defined in $\mathcal{X}$ and values in $\mathcal{Y}$

$$\begin{aligned} \mathcal{H} &\rightarrow \mathcal{Y} &&\rightarrow \mathbb{R} \\ f &\rightarrow f(x) &&\rightarrow (y, f(x)) \end{aligned}$$

Consider the functionals $L_{x,y}f = (y, f(x))$, if this functionals are continuous, we can apply Riesz Representation theorem. That is, for every $x \in \mathcal{X}, y \in \mathcal{Y}$ we can find an unique $g_{x,y} \in \mathcal{H}$ such that for all $f \in \mathcal{H}$,

$$L_{x,y}f = (y, f(x)) = \langle g_{x,y}, f \rangle_{\mathcal{H}}. \tag{3.19}$$

We can now give the definition of vector-valued Hilbert space from the point of view of continuous functionals (Micchelli and Pontil, 2005, Definition 2.1)

Definition 3.1 (vector-valued RKHS)**.** We say that $\mathcal{H}$ is a vector-valued RKHS when for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, the functional $L_{x,y}f = (y, f(x))$ is continuous.

Note that this is a definition similar to the scalar case but we use the inner product of $\mathcal{Y}$ to construct the scalar-valued functionals $L_{x,y}$. The price we pay is that it is necessary to express the Riesz representation as dependent of the elements $y \in \mathcal{Y}$. To get rid of this dependence for every $x \in \mathcal{X}$ we can define the linear operator

$$\begin{aligned} g_x : \mathcal{Y} &\rightarrow \mathcal{H} \\ y &\rightarrow g_x y = g_{x,y}. \end{aligned}$$

This operator is well defined because $g_{x,y}$ is unique for every $x \in \mathcal{X}, y \in \mathcal{Y}$ and its linearity is easy to check from the linearity of the inner product $(.,.)$.

Using this results can now define the operator

$$\begin{aligned} K(x, \hat{x}) : \mathcal{Y} &\rightarrow \mathcal{Y} \\ y &\rightarrow K(x, \hat{x})y = (g_{\hat{x}}y)(x) \end{aligned} \tag{3.20}$$

for every $x, \hat{x} \in \mathcal{X}$. Observe that $K(x, \hat{x})$ is linear since g_x is linear. It is possible then to prove that $K(x, \hat{x})$ is a reproducing kernel in $\mathcal{H}$ as seen (Micchelli and Pontil, 2005, Propositon 2.1). To do that, first we have to define a vector-valued kernel and the corresponding reproducing property.

Definition 3.2 (operator-valued Kernel)**.** If $\mathcal{Y}$ is a finite-dimensional Hilbert space, an operator-valued kernel is a function

$$K : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{L}(\mathcal{Y})$$

which is symmetric and positive definite.

For the clarity of the text an operator-valued K will be referred just as kernel unless an explicit distinction is needed.

Definition 3.3 (Reproducing Property of operator-valued operators)**.** If $\mathcal{Y}$ is a Hilbert space, a function

$$K : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{L}(\mathcal{Y})$$

has the reproducing property if $\forall x \in \mathcal{X}, y \in \mathcal{Y}, \forall f \in \mathcal{H}$, $(y, f(x)) = \langle K(\cdot, x)y, f\rangle$.

A kernel with the reproducing property is also called a reproducing kernel. The next proposition shows how we can build a reproducing kernel for a space $\mathcal{H}$ in which the evaluation functionals are continuous.

Proposition 3.4. *If for every $x, \hat{x} \in \mathcal{X}$, the function $K(x, \hat{x})$ is defined as in Equation (3.20), then the function*

$$K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$$

is a reproducing kernel.

Proof. To prove that K is a reproducing kernel we need to see

1. $K(x, \hat{x})$ is bounded for every $x, \hat{x} \in \mathcal{X}$ so K is well defined.
2. K is symmetric: $K(x, \hat{x})^* = K(\hat{x}, x)$.
3. K is positive definite: given $n \in \mathbb{N}$, for any $x_1, \dots, x_n \in \mathcal{X}, y_1, \dots, y_n \in \mathcal{Y}$,

$$\sum_{i,j=1}^{n} (y_i, K(x_i, x_j)y_j) \geq 0.$$

4. It has the reproducing property: $\forall x \in \mathcal{X}, y \in \mathcal{Y}, \forall f \in \mathcal{H}$, $(y, f(x)) = \langle K(\cdot, x)y, f\rangle$.

To prove 1 and 2,observe that by applying (3.19) to $f = g_{\hat{x}}\hat{y}$, for every $x \in \mathcal{X}, y \in \mathcal{Y}$ there exists an unique $g_{x,y} = g_x y$ such that

$$(y, (g_{\hat{x}}\hat{y})(x)) = \langle g_x y, g_{\hat{x}}\hat{y}\rangle. \tag{3.21}$$

and combining this result with (3.20) we get

$$(y, K(x, \hat{x})\hat{y}) = (y, (g_{\hat{x}}\hat{y})(x)) = \langle g_x y, g_{\hat{x}}\hat{y}\rangle.$$

Also, using the definitions,

$$(K(\hat{x}, x)y, \hat{y}) = ((g_x y)(\hat{x}), \hat{y}) = (\hat{y}, (g_x y)(\hat{x})) = \langle g_{\hat{x}}\hat{y}, g_x y\rangle.$$

Since both operators $K(x, \hat{x}), K(\hat{x}, x)$ are linear, by the Uniform Boundness Principle Akhiezer and Glazman (1961), $K(x, \hat{x}), K(\hat{x}, x)$ are bounded (hence continuous) and $K(x, \hat{x})^* = K(\hat{x}, x)$.
To prove 3 we write

$$\sum_{i,j=1}^{n} (y_i, K(x_i, x_j)y_j) = \sum_{i,j=1}^{n} \langle g_{x_i}y_i, g_{x_j}y_j\rangle = \left\| \sum_{i=1}^{n} g_{x_i}y_i \right\|^2 \geq 0$$

Finally, to prove 4 we use that $\forall x \in \mathcal{X}, y \in \mathcal{Y}, \forall f \in \mathcal{H}, \exists g_{x,y} \in \mathcal{H}$ such that

$$(y, f(x)) = \langle g_{x,y}, f\rangle = \langle g_x y, f\rangle = \langle K(\cdot, x)y, f\rangle.$$

□

Semi-positive definite kernels. The second approach changes the point of view. Given a kernel K, the Hilbert space from which K is the reproducing kernel is built. To do this, we use (Micchelli and Pontil, 2005, Theorem 2.1) which extends the Moore-Aronszanj's Theorem:

Theorem 3.5. *If $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is a kernel, then there exists a unique (up to an isometry) RKHS which admits K as its reproducing kernel.*

The proof is similar to that of the Moore-Aronszanj's Theorem, considering the space of the completion of the span of $\{K_x = K(\cdot, x), x \in \mathcal{X}\}$.

Feature map. The last approach is based on feature maps, which provide a very simple way of generating kernels.

Lemma 3.6. *Any feature map $\Phi : \mathcal{X} \to \mathcal{L}(\mathcal{W}, \mathcal{Y})$ defines a kernel as*

$$K(x, \hat{x}) = \Phi(x)\Phi(\hat{x})^* : \mathcal{Y} \to \mathcal{Y}. \tag{3.22}$$

Proof. We need to prove that it is bounded, symmetric and positive definite:

1. Since $\Phi(x)$ is continuous, the adjoint is $\Phi(x)^*$ is continuous and the composition $\Phi(x) \circ \Phi(\hat{x})^*$ is also continuous.

2. It is symmetric since $(\Phi(x) \circ \Phi(\hat{x})^*)^* = ((\Phi(x)^*)^* \circ \Phi(\hat{x})^*) = (\Phi(x) \circ \Phi(\hat{x})^*)$.

3. Is positive definite since, given any $x_1, \ldots, x_n \in \mathcal{X}, y_1, \ldots, y_n \in \mathcal{Y}$

$$\begin{aligned}\sum_{i,j=1}^{n} (y_i, K(x_i, x_j)y_j) &= \sum_{i,j=1}^{n} (y_i, \Phi(x_i)\Phi(x_j)^* y_j) \\ &= \sum_{i,j=1}^{n} (\Phi(x_i)^* y_i, \Phi(x_j)^* y_j) = \left\| \sum_{i=1}^{n} \Phi(x_i)^* y_i \right\|^2 \geq 0.\end{aligned}$$

□

Since K as defined in (3.22) is a kernel, according to Theorem 3.5 we can find its corresponding vector-valued hilbert space $\mathcal{H}$.

Representer Theorem for Operator-Valued Kernels

The Representer Theorem is a crucial result in Optimization and Machine Learning. Given a regularized empirical risk, under some assumptions, the theorem gives a precise description of the minimizer f^* as a finite linear combination of functions $K(\cdot, x_i)$ where x_i are part of the empirical sample. This result is extended in (Micchelli and Pontil, 2005, Theorem 4.2) for operator-valued kernels.

Theorem 3.7. *Let $\mathcal{Y}$ be a Hilbert space and let $\mathcal{H}$ be the Hilbert space of $\mathcal{Y}$-valued functions with an operator-valued reproducing kernel K. Let $V : \mathcal{Y}^n \times \mathbb{R}_+ \to \mathbb{R}$ be a function strictly increasing in its second variable and consider the problem of minimizing the functional*

$$E(f) := V((f(x_1), \ldots, f(x_n)), \|f\|^2) \tag{3.23}$$

in $\mathcal{H}$. If f_0 minimizes the E, then $f_0 = \sum_{j=1}^{n} K(\cdot, x_j)c_j$ where $c_j \in \mathcal{Y}$. In addition, if V is strictly convex, the minimizer is unique.

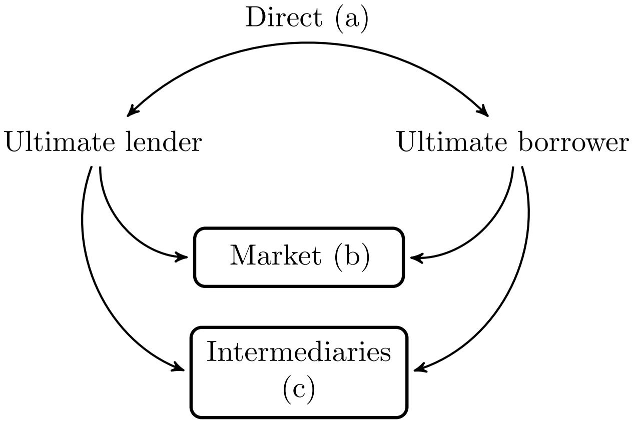

FIGURE 3.1: Bijections between kernel functions.

FIGURE 3.2: Bijections between kernel functions.

A useful bijection between kernel functions

The scalar-valued kernels are well known and studied but this is not the case for operator-valued kernels. However, as shown in Baldassarre et al. (2012); Hein et al. (2004) we can find a bijection between operator-valued kernels and scalar-valued ones.

Lemma 3.8. *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be an operator-valued kernel and consider the scalar-valued $L : (\mathcal{X}, \mathcal{Y}) \times (\mathcal{X}, \mathcal{Y}) \to \mathbb{R}$ such that $L((x, z), (\hat{x}, \hat{z})) = (z, K(x, \hat{x})\hat{z})$, then the map $K \to L$ is a bijection.*

Moreover, if we focus on finite dimensional Hilbert spaces, that is, isomorphic to $\mathbb{R}^d$ given an operator-valued kernel K the corresponding scalar-valued kernel L is defined using the normal basis as

$$L((x, e_r), (\hat{x}, e_s)) = (e_r, K(x, \hat{x})e_s) = K(x, \hat{x})_{rs}.$$

That is, each pair $(x, \hat{x})$ defines a matrix which contains the information of how the different outputs, or tasks, are related.

3.3.2 Tensor product of Hilbert Spaces

TODO: completar, comprobar que los funcionales de evaluación están acotados y que todo está bien definido

Consider the space of the tensor product of two Hilbert spaces $\mathcal{H}_1 \otimes \mathcal{H}_2$ where the functions $f \in \mathcal{H}_i$ are defined as $f : \mathcal{X}_i \to \mathcal{Y}_i$. This tensor space is also a Hilbert space endowed with the inner product:

$$\begin{array}{rcc} \langle , \rangle : (\mathcal{X}_1 \otimes \mathcal{X}_2) \times & (\mathcal{X}_1 \otimes \mathcal{X}_2) \to & \mathcal{L}(\mathcal{Y}_1) \otimes \mathcal{L}(\mathcal{Y}_2) \\ (f_1 \otimes f_2) & (\hat{f}_1 \otimes \hat{f}_2) \to & \langle f_1, \hat{f}_1 \rangle \otimes \langle f_2, \hat{f}_2 \rangle \end{array}$$

When $\mathcal{H}_1$ and $\mathcal{H}_2$ are scalar-valued Hilbert spaces, also is $\mathcal{H}_1 \otimes \mathcal{H}_2$. The reproducing kernel of such space is:

$$\begin{aligned} K_1 \otimes K_2 : (\mathcal{H}_1 \otimes \mathcal{H}_2) \times \quad & (\mathcal{H}_1 \otimes \mathcal{H}_2) \to & \mathcal{L}(\mathcal{Y}_1) \otimes \mathcal{L}(\mathcal{Y}_2) \\ (x_1 \otimes x_2) \quad & (\hat{x}_1 \otimes \hat{x}_2) \to & K_1(x_1, x_2) \otimes K_2(x_1, x_2) \end{aligned}$$

Another useful bijection between kernels

To understand how this tensor product space we study a special case of operator-valued kernels. A standard assumption is that the relation among the different outputs is independent of the pair $(x, \hat{x})$:

$$K(x, \hat{x}) = k(x, \hat{x})M$$

where k is a scalar-valued kernel and M is some fixed operator $M \in \mathcal{L}(\mathcal{Y})$. That is, the operator $K(x, \hat{x})$ decouples in two parts: the similarity between x and $\hat{x}$ measured by $k(\cdot, \cdot)$ and the interaction between the different outputs expressed by M.

Consider the tensor product space $\mathcal{X} \otimes \mathcal{T}$, where $\mathcal{T}$ is the space of tasks. We assume that $\mathcal{T}$ has finite dimension, therefore it is isomorphic to $\mathbb{R}^k$. We consider $\mathcal{H}$ as the Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ with reproducing kernel $K_{\mathcal{X}}$, and $\mathcal{T} = (\mathbb{R}^k)^*$ as the space of linear functions $g : \mathbb{R}^k \to \mathbb{R}$, which is isomorphic to $\mathbb{R}^k$. In $\mathbb{R}^k$ we consider a general inner product induced by the positive definite matrix A. Then $(\mathbb{R}^k)^*$, by the Riesz Representation Theorem, $\forall x \in \mathbb{R}^k, \exists u_x \in (\mathbb{R}^k)^*$ such that $\forall w \in (\mathbb{R}^k)^*$

$$\langle w, u_x \rangle_A = wx,$$

that is, $u_x = A^{-1}x$. Then, the kernel is defined as $K_{\mathcal{T}}(x, y) = \langle u_x, u_y \rangle_A = x^{\intercal} A^{-1} y$. Now we can define the following tensor product of two scalar-valued kernels:

$$\begin{aligned} K_{\mathcal{X}} \otimes K_{\mathcal{T}} : (\mathcal{X} \otimes \mathbb{R}^k) \times \quad & (\mathcal{X} \otimes \mathbb{R}^k) \to & \mathcal{L}(\mathbb{R}) \otimes \mathcal{L}(\mathbb{R}) \\ (x \otimes z) \quad & (\hat{x} \otimes \hat{z}) \to & K_{\mathcal{X}}(x, \hat{x}) \otimes K_{\mathcal{T}}(z, \hat{z}). \end{aligned}$$

The function $K_{\mathcal{X}} \otimes K_{\mathcal{T}}$ is the reproducing kernel of $\mathcal{H} \otimes (\mathbb{R}^k)^*$. Note that using this kernel with $z = e_r, \hat{z} = e_s$, that is, two examples of tasks r and s respectively, we have that

$$K_{\mathcal{X}} \otimes K_{\mathcal{T}}((x \otimes e_r), (\hat{x} \otimes e_s)) = K_{\mathcal{X}}(x, \hat{x})(A^{-1})_{rs}.$$

This gives us a hint of a bijection that we can establish with previous

3.3.3 Multi-Task Learning as Single-Task Learning

Building upon the ideas discussed in Evgeniou and Pontil (2004), two useful results are presented in Evgeniou et al. (2005), which show how we can apply Single-Task Learning methods to Multi-Task Learning problems. There exists a plethora of work about Single-Task Learning within regularization theory, some general formulation is

$$\sum_{i=1}^{n} \ell(y_i, \langle w, \phi(x_i) \rangle) + \lambda \langle w, w \rangle. \tag{3.24}$$

Here, ℓ is the loss function and ϕ is a transformation to include non-linearity. Popular models such as Ridge Regression or SVMs are particular cases of this formulation for different choices of ℓ. One crucial result for problems using this formulation is the *Representer Theorem*, which states that the any minimizer of problem (3.24) has the form

$$w = \sum_{i=1}^{N} c_j \phi(x_j). \tag{3.25}$$

Linear MTL Models

The first result (Evgeniou et al., 2005, Proposition 1) is defined for linear models and illustrates under which conditions we can adapt these results for MTL. Consider the linear MTL problem where we want to estimate the task parameters $u_r : r = 1, \dots, T$, so we define $\boldsymbol{u}^\intercal = (u_1^\intercal, \dots, u_T^\intercal) \in \mathbb{R}^{Td}$. Then we want to minimize

$$R(\boldsymbol{u}) = \sum_{r=1}^{T} \sum_{i=1}^{m} \ell(y_i^r, \langle u_r, x_i^r \rangle) + \mu J(\boldsymbol{u}), \tag{3.26}$$

where $J(\boldsymbol{u}) = \boldsymbol{u}^\intercal E \boldsymbol{u}$ is the regularizer where different choices of $J(\boldsymbol{u})$, i.e. choices of the matrix E, we can encode different beliefs about the task structure. For example, if $J(\boldsymbol{u}) = \sum_{r=1}^{T} \|u_r\|^2$ the problem decouples and we get independent task learning, so there is no relation among tasks; if $J(\boldsymbol{u}) = \sum_{r,s=1}^{T} \|u_r - u_s\|^2$ we are enforcing the parameters from different tasks to be close, so we expect all tasks to be similar.

Then, Evgeniou et al. propose to consider a vector $\boldsymbol{w} \in \mathbb{R}^p$ with $p \geq Td$ such that we can express $\langle u_r, x \rangle$ as $\langle B_r^\intercal \boldsymbol{w}, x \rangle$, where B_r is a $p \times d$ "selection" matrix yet to be specified. One condition for B_r is to be full rank so we can find such $\boldsymbol{w}$. Note that we can also interpret B_r as a feature map $f : \mathbb{R}^d \to \mathbb{R}^p$ such that $\langle u_r, x \rangle = \langle \boldsymbol{w}, B_r x \rangle$. Observe that using the matrices B_r we have the following MTL kernel

$$\hat{k}(x^r, y^s) = \hat{k}((x, r), (y, s)) = x^\intercal B_r^\intercal B_s y.$$

Using these feature maps we would like to write the MTL problem as a Single-Task problem

$$S(\boldsymbol{w}) = \sum_{r=1}^{T} \sum_{i=1}^{m} \ell(y_i^r, \langle \boldsymbol{w}, B_r x_i^r \rangle) + \mu \langle \boldsymbol{w}, \boldsymbol{w} \rangle, \tag{3.27}$$

We also define the feature matrix B as the concatenation $B = (B_r : r = 1, \dots, T) \in \mathbb{R}^{p \times Td}$, then we present the first result of Evgeniou et al. (2005).

Proposition 3.9. *If the feature matrix B is full rank and we define the matrix E in equation (3.26) as to be $E = (B^\intercal B)^{-1}$ then we have that*

$$S(\boldsymbol{w}) = R(B^\intercal \boldsymbol{w}).$$

and therefore $\boldsymbol{u}^* = B^\intercal \boldsymbol{w}^*$.

One important consequence of this result is that since we can solve the MTL problem (3.26) as the STL problem (3.24), then we can apply the *Representer Theorem*. That is, the solution $\boldsymbol{w}^*$ of problem (3.24) has the form

$$\boldsymbol{w} = \sum_{r=1}^{T} \sum_{i=1}^{m} c_i B_r x_i^r,$$

and the prediction can be expressed as

$$\langle \boldsymbol{w}, y^s \rangle = \sum_{r=1}^{T} \sum_{i=1}^{m} c_i (x_i^r)^\intercal B_r^\intercal B_s y^s = \sum_{r=1}^{T} \sum_{i=1}^{m} c_i \hat{k}(x_i^r, y^s).$$

Kernel Extension of MTL Models

Evgeniou et al. also extend this results to kernelized models. In the following lemma (Evgeniou et al., 2005, Lemma 2) the give the conditions under which the extension is possible.

Lemma 3.10. *If G is a kernel on $\mathcal{T} \times \mathcal{T}$ and, for every $r = 1, \dots, T$ there are prescribed mappings $z_r : \mathcal{X} \to \mathcal{T}$ such that*

$$K((x,r),(y,s)) = G(z_r(x), z_s(t)),\ x,t \in \mathcal{X},\ r,s \in [T] \tag{3.28}$$

then K is a multi-task kernel.

That defines K as a semi-positive functional over the product space $\mathcal{X} \times \mathcal{T}$, which is named multi-task kernel. The mappings described in Evgeniou et al. (2005) are

$$z_r(x) = B_r x$$

where B_r are the $p \times d$ matrices previously defined. Then, two examples of multi-task kernels using this lemma are given. The polynomial kernel is defined as

$$K((x,r),(y,s)) = (x^\intercal B_r^\intercal B_s y)$$

and the multi-task Gaussian kernel is defined as

$$K((x,r),(y,s)) = \exp\left(-\gamma \left\| B_r x - B_s y \right\|^2\right).$$

That is, using the result of Proposition 3.9, when the matrix E has a block structure, we can incorporate the task-regularizer information into the Gaussian kernel using that

$$\begin{aligned} \left\| B_r x - B_s y \right\|^2 &= x^\intercal B_r^\intercal B_r x + y^\intercal B_s^\intercal B_s y - 2 x_r^\intercal B_r^\intercal B_s y_s \\ &= x^\intercal E_{rr}^{-1} x + y^\intercal E_{ss}^{-1} y - 2 x_r^\intercal E_{rs}^{-1} y_s. \end{aligned}$$

That is, we use the task information in the original space and then apply the non-linear transformation.

Alternative Kernel Extension of MTL Models

The kernel extension proposed in Evgeniou et al. (2005) proposes to use a mapping in the original finite space to incorporate the task information and then applying the kernel trick over this new mapped features. However, these results do not permit to perform the, possibly infinite dimensional, mapping corresponding to a kernel and then incorporate the task information in the new space. Here we show another approach which makes it is possible to replicate the results for the infinite-dimensional case by using tensor products.

Consider a general Hilbert space $\mathcal{H}$ and the functional

$$R(u_1,\dots,u_T) = \sum_{r=1}^{T}\sum_{i=1}^{m} \ell(y_i^r, \langle u_r, \phi(x_i^r)\rangle) + \mu \sum_{r}\sum_{s} E_{rs}\langle u_r, u_s\rangle, \tag{3.29}$$

where $u_1,\dots,u_T \in \mathcal{H}$. Replicating the idea of Evgeniou et al. (2005), we can write

$$\boldsymbol{u} = \sum_{t=1}^{T} e_r \otimes u_r,$$

such that $\boldsymbol{u} \in \mathbb{R}^T \otimes \mathcal{H}$. Then, we can reformulate (3.29) as

$$R(\boldsymbol{u}) = \sum_{r=1}^{T}\sum_{i=1}^{m} \ell(y_i^r, \langle \boldsymbol{u}, e_r \otimes \phi(x_i^r)\rangle) + \mu\left(\boldsymbol{u}^\intercal (E\otimes I)\boldsymbol{u}\right), \tag{3.30}$$

Since $E \in \mathbb{R}^{T\times T}$ is positive definite, we can find express $B \in \mathbb{R}^{p\times T}, p \geq T$ and rank $B = T$ such that $E^{-1} = B^\intercal B$. Using the properties of the tensor product of linear maps,

$$E^{-1}\otimes I = (B^\intercal B)\otimes I = (B^\intercal \otimes I)(B\otimes I),$$

Consider the change of variable $\boldsymbol{u} = (B^\intercal \otimes I)\boldsymbol{w}$, where $\boldsymbol{w} \in \mathbb{R}^p \otimes \mathcal{H}$. Observe that this can always be done because B is full rank. Rewriting (3.30) using $\boldsymbol{w}$,

$$\begin{aligned} R((B^\intercal\otimes I)\boldsymbol{w}) &= \sum_{r=1}^{T}\sum_{i=1}^{m} \ell(y_i^r, \langle (B^\intercal\otimes I)\boldsymbol{w}, (e_r\otimes\phi(x_i^r))\rangle) + \mu\boldsymbol{w}^\intercal (B^\intercal\otimes I)^\intercal (E\otimes I)(B^\intercal \otimes I)\boldsymbol{w}, \\ &= \sum_{r=1}^{T}\sum_{i=1}^{m} \ell(y_i^r, \langle \boldsymbol{w}, (B\otimes I)(e_r\otimes\phi(x_i^r))\rangle) + \mu\boldsymbol{w}^\intercal\boldsymbol{w}, \end{aligned}$$

which is equivalent to

$$S(\boldsymbol{w}) = \sum_{r=1}^{T}\sum_{i=1}^{m} \ell(y_i^r, \langle \boldsymbol{w}, (B_r\otimes\phi(x_i^r))\rangle) + \mu\boldsymbol{w}^\intercal\boldsymbol{w},$$

We are thus considering a regularized functional $S(\boldsymbol{w})$ where we seek the minimum over functions w in the Hilbert space $\mathbb{R}^p \otimes \mathcal{H}$. Note that in this space the inner product is:

$$\begin{aligned} \langle,\rangle : (\mathbb{R}^p\otimes\mathcal{H})\times \quad & (\mathbb{R}^p\otimes\mathcal{H}) \to & \mathbb{R} \\ (z_1,\phi(x_1)), \quad & (z_2,\phi(x_2)) \to & \langle z_1, z_2\rangle k(x_1,x_2) \end{aligned}$$

Where $k(\cdot,\cdot)$ is the reproducing kernel of the space of functions $\phi(\cdot)$. However we are only interested in those cases where $z = Be_r = B_r$ for some $r = 1,\dots,T$, then $\langle B_r, B_s\rangle = E_{rs}^{-1}$.

Since the regularizer is clearly increasing in $\|w\|^2$, we can apply the Representer theorem, which states that the minimizer of $S(\boldsymbol{w})$ has the form

$$\boldsymbol{w}^* = \sum_{r=1}^{T}\sum_{i=1}^{m} \alpha_i^r (B_r\otimes\phi(x_i^r)).$$

Examples of Multi-Task Kernels

The specific case where $\mathcal{Y} = \mathbb{R}^T$, let $K(x_i, x_j) : \mathbb{R}^T \to \mathbb{R}^T$ and $(e_r, k(x_i, x_j), e_s)$:

3.3.4 Beyond Multi-Task Learning with Kernel Methods

3.3.5 MTLSVM and SVM+

Another extension of the *regularized MTL* SVM model can be found in the multi-task problem described in Liang and Cherkassky (2008) for classification, also adapted for regression problems in Cai and Cherkassky (2009). Using the unified formulation

$$
\begin{array}{ll}
\underset{w,b,v_r,b_r,\xi_i^r}{\arg\min} & C \sum_{r=1}^{T} \sum_{i=1}^{m_r} \xi_i^r + \frac{1}{2}\langle w, w\rangle + \frac{\mu}{2} \sum_{r=1}^{T} \langle v_r, v_r \rangle \\
\text{s.t.} & y_i(\langle w, \phi(x_i^r)\rangle + b + \langle v_r, \phi_r(x_i^r)\rangle + b_r) \geq p_i^r - \xi_i^r, \\
 & \xi_i^r \geq 0, \\
\text{for} & r = 1, \dots, T;\ i = 1, \dots, m_r.
\end{array}
$$

Comparing (**??**) and (3.31) we observe that the subyacent idea is the same, but there exists some differences. In first place, note that (**??**) is described as a linear model, while in (3.31) not only non-linear transformations of the data are used, but different transformations can be selected ϕ, ϕ_r for the common part and for each task-specific term, respectively. Moreover, it is relevant to note the incorporation of the bias terms in (3.31). The dual form of (3.31) is

$$
\begin{array}{ll}
\underset{\alpha_i}{\arg\min} & \frac{1}{2} \sum_{r,s=1}^{T} \sum_{i,j=1}^{m_r} y_i^r y_j^s \alpha_i^r \alpha_j^s \left[k(x_i^r, x_j^s) + \delta_{rs} k_r(x_i^r, x_j^s)\right] - \sum_{r=1}^{T} \sum_{i=1}^{m_r} p_i^r \alpha_i^r \\
\text{s.t.} & 0 \leq \alpha_i^r \leq C \\
 & \sum_{i=1}^{m_r} \alpha_i^r y_i^r = 0, \\
\text{for} & r = 1, \dots, T;\ i = 1, \dots, m_r.
\end{array}
\tag{3.31}
$$

In (3.31), T equality constraints that are not present in (**??**) have been added. This a direct consequence of the incorporation of bias terms in the primal formulation. Since the original SMO algorithm does not account for multiple equality constraints, in Cai and Cherkassky (2012) a generalized SMO algorithm is developed. Also, it is important to observe the use of different kernel spaces through the functions k and k_r. This has connections with the LUPI paradigm Vapnik and Izmailov (2015) and the proposed SVM+ that embodies this paradigm, as described in Subsection 3.4.4. The kernel space for the common part is named the decision space, and the spaces corresponding to the kernel functions k_r are the correcting spaces. That is, each task can independently correct the similarity defined by the common decision space.

3.4 Why does Multi-Task Learning work?

TODO: Repasar sección entera después escribir capítulo 2

3.4.1 Learning to Learn: A notion of environment of tasks

Tipically in Machine Learning the goal is to find the best hypothesis $h\left(x, \alpha_0\right)$ from a space of hypothesis $\mathcal{H} = \left\{h\left(x, \alpha\right), \alpha \in A\right\}$, where A 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 $f\left(x\right) = P\left(y \mid x\right)$ and the hypothesis space is parametric, i.e. $\mathcal{H} = \left\{h\left(x, \alpha\right), \alpha \in A \subset \mathbb{R}^m\right\}$. 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 $h\left(x, \alpha\right)$ 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 equivalent the original problem. When we only want to estimate a single function $f(x)$, a single-task scenario, there is no difference between learning the optimal hypothesis space (bias learning) and ordinary learning of the optimal hypothesis function. Instead, we focus on the situation where we want to solve multiple related tasks, that is, estimating multiple functions $f_1(x), \dots, f_T(x)$. 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 or Learning to Learn, 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* $\ell\left(\cdot, \cdot\right) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, and
- a *hypothesis space* $\mathcal{H} = \left\{h\left(x, \alpha\right), \alpha \in A \subset \mathbb{R}^m\right\}$ with hypothesis $h\left(\cdot, \alpha\right) : \mathcal{X} \to \mathcal{Y}$.

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

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

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:

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

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 $\alpha \in A$ and, more importantly, bounds on the difference $\inf_{\alpha \in A} \hat{R}_z(\alpha) - \inf_{\alpha \in A} R_F(\alpha)$. Under some general conditions, he proves that

$$\inf_{\alpha \in A} \hat{R}_z(\alpha) - \inf_{\alpha \in A} R_F(\alpha) \leq B(n/\mathrm{VCdim}\,(\mathcal{H})) \tag{3.32}$$

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 $\mathrm{VCdim}\,(\mathcal{H})$ 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 $\inf_{\alpha \in A} R_F$ 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 α completely defines h, we also substitute α 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* $\ell(\cdot,\cdot) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, and
- a *family of hypothesis spaces* $\mathbb{H} = \{\mathcal{H}_\alpha, \alpha \in A\}$, where each element $\mathcal{H}_\alpha$ is a set of hypothesis.

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

$$R_Q(\alpha) = \int_{\mathcal{P}} \inf_{h \in \mathcal{H}_\alpha} R_P(h) dQ(P) = \int_{\mathcal{P}} \inf_{h \in \mathcal{H}_\alpha} \int_{\mathcal{X} \times Y} l(h(x), y) dP(x,y) dQ(P). \tag{3.33}$$

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 $(x_i^r, y_i^r) \in X \times Y$.

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

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

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

$$\hat{R}_{\boldsymbol{z}}(\alpha) = \sum_{r=1}^{T} \inf_{h \in \mathcal{H}_\alpha} \hat{R}_{z_r}(h) = \sum_{r=1}^{T} \inf_{h \in \mathcal{H}_\alpha} \sum_{i=1}^{m} l(h(x_i^r), y_i^r), \tag{3.34}$$

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

To achieve these goals some previous definitions are needed. From this point, since any $\mathcal{H}$ is defined by a $\alpha \in A$, we omit α 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 $\boldsymbol{P} = (P_1, \dots, P_T)$ sampled from $\mathcal{P}$ according the the distribution Q.

Definition 3.11 (*sample-driven* pseudometric)**.** Consider also the set of sequences of T hypothesis

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

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

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

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

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

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.

Definition 3.12 (*distribution-driven* pseudo-metric)**.** 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 d_Q:

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

for $\mathcal{H}_1^*, \mathcal{H}_2^* \in \mathbb{H}^*$. With these two pseudo-metrics, two capacities for families of hypothesis spaces are defined. For that the definition of ϵ-cover is needed.

Definition 3.13 (ϵ-cover)**.** 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 ϵ-cover of $\mathcal{S}$ if $\forall s \in S$ $d_S(s, s_i) \leq \epsilon$ for some $i = 1, \dots, l$. Let $\mathcal{N}(\epsilon, \mathcal{S}, d_S)$ denote the size of the smallest 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* $C\left(\epsilon, \mathbb{H}^*\right) := \sup_Q \mathcal{N}(\epsilon, \mathbb{H}^*, d_Q)$.

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

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

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

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

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 $(\mathcal{P}, Q)$-environment of tasks. In Multi-Task Learning, we have a fixed number of tasks T and a fixed sequence of distributions $\boldsymbol{P} = (P_1, \dots, P_T)$, 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 $\boldsymbol{h} = (h_1, \dots, h_T), h_1, \dots, h_T \in \mathcal{H}$. Thus, the Multi-Task expected risk is

$$R_{\boldsymbol{P}}(\boldsymbol{h}) = \sum_{r=1}^{T} R_{P_r}(h_r) = \sum_{r=1}^{T} \int_{\mathcal{X} \times Y} l(h_r(x), y) dP_r(x, y), \tag{3.35}$$

and the empirical risk is defined as

$$\hat{R}_{\boldsymbol{z}}(\boldsymbol{h}) = \sum_{r=1}^{T} \hat{R}_{z_r}(h_r) = \sum_{r=1}^{T} \sum_{i=1}^{m} l(h_r(x_i^r), y_i^r). \tag{3.36}$$

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

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

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

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

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 $\mathcal{F} = \{f, f : \mathcal{X} \to \mathcal{V}\}$. 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

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

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

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

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-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 $\hat{R}_{\boldsymbol{z}}(\boldsymbol{h}) - R_Q(\boldsymbol{h})$, that is, the probability of deviations between the empirical and expected risks for a given hypothesis sequence. However, it would be more useful to find some result concerning the empirical error and the *best expected error*. To achieve this, a generalized VC-dimension is developed in Baxter (2000) for Multi-Task Learning with Boolean hypothesis.

Definition 3.14. 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 $\Pi_{\mathbb{H}}(T,m) = 2^{Tm}$ we say that $\mathbb{H}$ shatters $\mathcal{X}^{T\times m}$. For each $T > 0$ define

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

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

$$\overline{d}(\mathbb{H}) := \mathrm{VCdim}\left(\mathbb{H}^1\right) = \mathrm{VCdim}\left(\bigcup_{\mathcal{H}\in\mathbb{H}} \mathcal{H}\right),$$
$$\underline{d}(\mathbb{H}) := \max_{\mathcal{H}\in\mathbb{H}} \mathrm{VCdim}\left(\mathcal{H}\right).$$

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

where it can be observed that

$$\overline{d}(\mathbb{H}) \geq d_{\mathbb{H}}(T) \geq \underline{d}(\mathbb{H}). \tag{3.37}$$

Proof. Let split the proof in three inequalities:

- $d_{\mathbb{H}}(T) \geq \underline{d}(\mathbb{H})$
Let $\mathcal{H}^{\max}$ be such that $\mathrm{VCdim}\left(\mathcal{H}^{\max}\right) = \underline{d}(\mathbb{H}) = \max_{\mathcal{H}\in\mathbb{H}} \mathrm{VCdim}\left(\mathcal{H}\right)$. Then we can construct a $T \times \underline{d}(\mathbb{H})$ matrix X with T rows where each row can be shattered

by $\mathcal{H}^{\max}$ and, thus, $\Pi_{\{\mathcal{H}^{\max}\}}(T,m) \geq T\underline{d}(\mathbb{H})$ (es esto verdad?) Also, for all $T, m \geq 0$

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

Then, since $\left\{m : \Pi_{\{\mathcal{H}^{\max}\}}(T,m) = 2^{Tm}\right\} \subset \left\{m : \Pi_{\mathbb{H}}(T,m) = 2^{Tm}\right\}$,

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

- $d_{\mathbb{H}}(T) \geq \left\lfloor \frac{\overline{d}(\mathbb{H})}{T} \right\rfloor$
  Given a sequence $x_1, \ldots, x_{\overline{d}(\mathbb{H})}$ shattered by $\mathbb{H}^1$ we can build a $T \times \left\lfloor \frac{\overline{d}(\mathbb{H})}{T} \right\rfloor$ matrix X that is shattered by $\mathbb{H}^1$ and therefore $d_{\mathbb{H}}(T) = \max_{m:\Pi_{\mathbb{H}}(T,m)=2^{Tm}} m \geq \left\lfloor \frac{\overline{d}(\mathbb{H})}{T} \right\rfloor$.

- $d_{\mathbb{H}}(T) \leq \overline{d}(\mathbb{H})$
  We prove the stronger result $d_{\mathbb{H}}(T) \leq \left\lceil \frac{\overline{d}(\mathbb{H})}{T} \right\rceil$. Suppose that $d_{\mathbb{H}}(T) > \left\lceil \frac{\overline{d}(\mathbb{H})}{T} \right\rceil$, then there exists a $T \times d_{\mathbb{H}}(T)$ matrix that is shattered by $\mathbb{H}^1$, then, $\mathrm{VCdim}\left(\mathbb{H}^1\right) \geq T d_{\mathbb{H}}(T) > \overline{d}(\mathbb{H})$, which is a contradiction.

□

Now we can present the relevant result expressed in (Baxter, 2000, Corollary 13).

Theorem 3.15. *Given a sequence* $\boldsymbol{P} = (P_1, \ldots, P_T)$ *on* $(\mathcal{X} \times \{0,1\})^T$*, and a sample* $\boldsymbol{z}$ *from this distribution. Consider also a sequence* $\boldsymbol{h} = (h_1, \ldots, h_T)$ *of boolean hypothesis* $h_i \in \mathcal{H}$*, then for every* $\epsilon > 0$

$$\left|R_{\boldsymbol{P}}(\boldsymbol{h}) - \hat{R}_{\boldsymbol{z}}(\boldsymbol{h})\right| \leq \epsilon,$$

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

$$m \geq \frac{88}{\epsilon^2}\left[2d_{\mathbb{H}}(T)\log\frac{22}{\epsilon} + \frac{1}{T}\log\frac{4}{\eta}\right]. \tag{3.38}$$

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 $\boldsymbol{P}$ such that

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

With this results we can see that the condition (3.38) 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.

3.4.2 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.

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 $f : \mathcal{X} \to \mathcal{X}$. Given a probability distribution F over $\mathcal{X} \times \{0, 1\}$, a set of tasks with distributions $P_1, \ldots, P_T$ are $\mathcal{F}$-related if, for each task there exists some $f_i \in \mathcal{F}$ such that $P_i = f_i(F)$.

Definition 3.16 ($\mathcal{F}$-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 $f : \mathcal{X} \to \mathcal{X}$, 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 $f : \mathcal{X} \to \mathcal{X}$, 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

$$\mathcal{H}_{\mathcal{F}} := \mathcal{H} / \sim_{\mathcal{F}} = \{[h], h \in \mathcal{H}\}.$$

It is important to observe that $\mathcal{H}_{\mathcal{F}} = \mathbb{H}'$ is a hypothesis space family, since it is a set of equivalence classes $[h] = \mathcal{H}'$, which are set of hypothesis.

The Multi-Task Empirical Risk Minimization

This equivalence classes are useful to divide the learning process in two stages, this is called the *Multi-Task ERM*. Consider the samples $z_1, \ldots, z_T$ from T different tasks, then

1. Select the best hypothesis class $[h^{\mathcal{F}}] \in \mathcal{H}_{\mathcal{F}}$:

$$[h^{\mathcal{F}}] := \min_{[h]\in\mathcal{H}_{\mathcal{F}}} \inf_{h_1,\ldots,h_T\in[h]} \frac{1}{T}\sum_{r=1}^{T} \hat{R}_{z_i}(h_i),$$

2. Select the best hypothesis $h^{\diamond}$ for the target task (without loss of generality, consider the first one):

$$h^{\diamond} = \inf_{h\in[h^{\mathcal{F}}]} \hat{R}_{z_1}(h).$$

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 $P_1, \ldots, P_T$ 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 $h_1, \ldots, h_T \in [h]$ for each dataset in the second one.

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

The results of Theorem 3.15 can be applied to the hypothesis quotient 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):

$$\inf_{h\in\mathcal{H}} R_{P_1}(h) = \inf_{h\in\mathcal{H}} R_{P_2}(h). \tag{3.39}$$

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

Theorem 3.17. *Let $\mathcal{F}$ be a set of transformations $f : \mathcal{X} \to \mathcal{X}$ that is a group under function composition. Let $\mathcal{H}$ be a hypothesis space so that $\mathcal{F}$ acts as a group over $\mathcal{H}$, and consider the quotient space $\mathcal{H}_{\mathcal{F}} = \{[h], h \in \mathcal{H}\}$. Consider $\boldsymbol{P} = (P_1, \ldots, P_T)$ a sequence of $\mathcal{F}$-related distributions over $\mathcal{X} \times \{0, 1\}$, and $\boldsymbol{z} = (z_1, \ldots, z_T)$ the corresponding sequence of samples where z_i is sampled using P_i, then for every $[h] \in \mathcal{H}_{\mathcal{F}}$ and $\epsilon > 0$*

$$\left| \inf_{h_1,\ldots,h_T\in[h]} \frac{1}{T}\sum_{r=1}^{T} \hat{R}_{z_r}(h_r) - \inf_{h'\in[h]} R_{P_1}(h') \right| \leq \epsilon$$

with probability greater than η if the number of samples from each distribution satisfies

$$|z_i| \geq \frac{88}{\epsilon^2}\left[2d_{\mathcal{H}_{\mathcal{F}}}(T)\log\frac{22}{\epsilon} + \frac{1}{T}\log\frac{4}{\eta}\right]. \tag{3.40}$$

Note that, in contrast to Theorem 3.15, this result bounds the expected risk of a single task, not the average risk. This is the consequence of applying Theorem 3.15 and substituting the average empirical error using the result from (3.39). Also observe that here the hypothesis space family used is the quotient space $\mathcal{H}_{\mathcal{F}}$, and the VC-dimension of such family is used. Using this result, a bound for learners using the Multi-Task ERM principle is given (Ben-David and Borbely, 2008, Theorem 3)

Theorem 3.18. *Consider $\mathcal{F}$ and $\mathcal{H}$ as in the previous theorem. Consider also the previous sequences of distributions $(P_1, \ldots, P_T)$ and corresponding samples $(z_1, \ldots, z_T)$. Consider $\underline{d}(\mathcal{H}_{\mathcal{F}}) = \max_{h \in \mathcal{H}} VCdim([h])$. Let $h^\diamond$ be the hypothesis selected using the Multi-Task ERM principle, then for every $\epsilon_1, \epsilon_2 > 0$*

$$\hat{R}_{z_1}(h^\diamond) - \inf_{h' \in \mathcal{H}} R_{P_1}(h') \leq 2(\epsilon_1 + \epsilon_2)$$

with probability greater than η if

$$|z_1| \geq \frac{64}{\epsilon^2}\left[2\underline{d}(\mathcal{H}_{\mathcal{F}}) \log \frac{12}{\epsilon} + \frac{1}{T} \log \frac{8}{\eta}\right], \tag{3.41}$$

and for $i \neq 1$

$$|z_i| \geq \frac{88}{\epsilon^2}\left[2d_{\mathcal{H}_{\mathcal{F}}}(T) \log \frac{22}{\epsilon} + \frac{1}{T} \log \frac{8}{\eta}\right]. \tag{3.42}$$

The idea of the proof of this theorem helps to understand how using different tasks can help to improve the performance in the target task. Consider $h^* = \inf_{h \in \mathcal{H}} R_{P_1}(h)$ the best hypothesis for the P_1 distribution. According to Theorem 3.17, for $[h^*]$ we have that

$$\inf_{h_1, \ldots, h_T \in [h^*]} \frac{1}{T} \sum_{r=1}^{T} \hat{R}_{z_r}(h_r) \leq \inf_{h' \in [h^*]} R_{P_1}(h') + \epsilon_1.$$

Also, in the first stage of Multi-Task ERM principle, we select the hypothesis class $[h^{\mathcal{F}}]$ that minimizes $\inf_{\boldsymbol{h} \in [h]} R_{\boldsymbol{P}}(\boldsymbol{h})$ where $\boldsymbol{h}$ is a sequence of hypothesis of $\mathcal{H}_{\mathcal{F}}$. According to Theorem 3.17, for $[h^{\mathcal{F}}]$ we have that

$$\inf_{h' \in [h^{\mathcal{F}}]} R_{P_1}(h') \leq \inf_{h_1, \ldots, h_T \in [h^{\mathcal{F}}]} \frac{1}{T} \sum_{r=1}^{T} \hat{R}_{z_r}(h_r) + \epsilon_1.$$

Using these two inequalities we get

$$\inf_{h' \in [h^{\mathcal{F}}]} R_{P_1}(h') \leq \inf_{h' \in [h^*]} R_{P_1}(h') + 2\epsilon_1$$

under the condition (3.40). This bounds the risk of the hypothesis space given by the equivalence class of $h^{\mathcal{F}}$ and establishes the inequality (3.42).

Once we select $[h^{\mathcal{F}}]$, the second stage is just ERM using this hypothesis space. According to the Vapnik (1982),

$$\inf_{h \in \mathcal{H}} R_{z_1}(h) - \inf_{h \in \mathcal{H}} R_{P_1}(h) \leq \epsilon_2$$

if

$$|z_1| \geq \frac{64}{\epsilon^2}\left[2\text{VCdim}\,(\mathcal{H}) \log \frac{12}{\epsilon} + \frac{1}{T} \log \frac{8}{\eta}\right].$$

Since the ERM will not use the whole space $\mathcal{H}$ but the subset $[h^{\mathcal{F}}] \subset \mathcal{H}$, and

$$\text{VCdim}\left([h^{\mathcal{F}}]\right) \leq \max_{[h] \in \mathcal{H}_{\mathcal{F}}} \text{VCdim}\,([h]) = \underline{d}(\mathcal{H}_{\mathcal{F}}).$$

then we can write the inequality (3.41) of the theorem. The advantage of using multiple tasks is then illustrated in this bound and it will be defined by the gap between $\text{VCdim}(\mathcal{H})$ and $\underline{d}(\mathcal{H}_{\mathcal{F}})$. If $\underline{d}(\mathcal{H}_{\mathcal{F}})$ is smaller than $\text{VCdim}(\mathcal{H})$, the number of samples needed to solve the target task will also be smaller. Also, the sample complexity of the rest of tasks is given by $d_{\mathcal{H}_{\mathcal{F}}}(T)$.

That is, Multi-Task Learning allows to select a subset of hypothesis from which a learner can use the ERM principle. In this stage, the sample complexity is controlled by the generalized VC-dimension of the set of equivalent classes of hypothesis. Once the best equivalent class has been selected, the VC-dimension of this subset, compared to the VC-dimension of the whole set of hypothesis, is what marks the difference between Single Task and Multi-Task Learning.

Analysis of generalized VC-dimension with $\mathcal{F}$-related tasks

As we have seen in Theorem 3.18, the VC-dimensions $\text{VCdim}(\mathcal{H}), \underline{d}(\mathcal{H}_{\mathcal{F}})$ and $d_{\mathcal{H}_{\mathcal{F}}}(T)$ are crucial for stating the advantage of Multi-Task over Single Task Learning. To understand better how these concepts interact, Ben-David et al. give some theoretical results. Recall that, given a hypothesis space $\mathcal{H}$, $\mathcal{H}_{\mathcal{F}}$ is a family of hypothesis spaces composed by the hypothesis spaces $[h], h \in \mathcal{H}$, then

$$\begin{aligned} d_{\mathcal{H}_{\mathcal{F}}}(T) &= \max_{\left\{m, \Pi_{\mathcal{H}_{\mathcal{F}}} = 2^{Tm}\right\}} m, \\ \underline{d}(\mathcal{H}_{\mathcal{F}}) &= max_{h \in \mathcal{H}} \text{VCdim}([h]), \\ \overline{d}(\mathcal{H}_{\mathcal{F}}) &= \text{VCdim}\left(\bigcup_{[h] \in \mathcal{H}_{\mathcal{F}}} [h]\right) = \text{VCdim}(\mathcal{H}). \end{aligned}$$

Using the result from (3.37) we observe that

$$\underline{d}(\mathcal{H}_{\mathcal{F}}) \leq d_{\mathcal{H}_{\mathcal{F}}}(T) \leq \text{VCdim}(\mathcal{H}).$$

That is, the best we can hope when bounding the sample complexity in Theorem 3.18 is $\underline{d}(\mathcal{H}_{\mathcal{F}}) = d_{\mathcal{H}_{\mathcal{F}}}(T)$. Ben-David et al. give evidence that, with some restrictions on $\mathcal{H}$, this lower bound can be achieved (Ben-David and Borbely, 2008, Theorem 4).

Theorem 3.19. *If the support of h is bounded, i.e. $|\{x \in \mathcal{X}, h(x) = 1\}| < M$, for all $h \in \mathcal{H}$, then there exists T_0 such that for all $T > T_0$*

$$d_{\mathcal{H}_{\mathcal{F}}}(T) = \underline{d}(\mathcal{H}_{\mathcal{F}}).$$

Thus, a sufficient condition on the hypothesis space $\mathcal{H}$ to achieve the lowest $d_{\mathcal{H}_{\mathcal{F}}}(T)$ is a bounded support of any hypothesis. Although this condition may be too restricting, it can also be proved that the upper limit of $d_{\mathcal{H}_{\mathcal{F}}}(T)$, that is, $\text{VCdim}(\mathcal{H})$, under some conditions on $\mathcal{F}$ is not achieved.

The following result (Ben-David and Borbely, 2008, Theorem 6) shows this.

Theorem 3.20. *If $\mathcal{F}$ is finite and $\frac{T}{\log(T)} \geq \text{VCdim}(\mathcal{H})$, then*

$$d_{\mathcal{H}_{\mathcal{F}}}(T) \leq 2\log(|\mathcal{F}|)$$

This inequality indicates that, given a finite set of transformation $\mathcal{F}$, there are scenarios when $\text{VCdim}(\mathcal{H})$ is arbitrarily large but $d_{\mathcal{H}_{\mathcal{F}}}(T)$ is bounded, and therefore, the right-hand side of inequality (3.42) is also bounded. That is, the Multi-Task bound, which substitutes $\mathrm{VCdim}\,(\mathcal{H})$ by $d_{\mathcal{H}_{\mathcal{F}}}(T)$ is a better one in this cases.

3.4.3 Other bounds for Multi-Task Learning

TODO: Completar después de escribir capítulo 2. Usar discusión de cotas en Maurer et al. 2016 para estructurar la subsección. The work of Baxter Baxter (2000) set the foundations for the theoretical analysis of MTL. In this work, an MTL extension to the VC-dimension is given, and it is used to develop some results bounding the difference between the Multi-Task empirical and expected risks

$$\left| R_{\boldsymbol{P}}(\boldsymbol{h}) - \hat{R}_{\boldsymbol{z}}(\boldsymbol{h}) \right|$$

for any sequence of hypothesis $\boldsymbol{h} \in \mathcal{H}^T$, see Theorem 3.15. This is a necessary condition for the consistency of Multi-Task Learners, but not a sufficient one. Then, Ben-David et al. Ben-David and Borbely (2008); Ben-David and Schuller (2003) defines a notion of task relatedness, see Definition 3.16. Using this notion and building an appropiate hypothesis space $\mathcal{H} = [h]$, they bound the *excess risk* of an Empirical Multi-Task Learner, that is, the difference between the best empirical risk, achieved by such Learner, and the best possible expected risk (Theorem 3.18)

$$\left| \inf_{\boldsymbol{h} \in \mathcal{H}^T} R_{\boldsymbol{P}}(\boldsymbol{h}) - \inf_{\boldsymbol{h} \in \mathcal{H}^T} \hat{R}_{\boldsymbol{z}}(\boldsymbol{h}) \right| .$$

Moreover, using this task relatedness definition not only the Multi-Task average excess risk is bounded, but the individual excess risk of each task (Theorem 3.20).

The works discussed until this point on the VC-dimension, and the corresponding extensions to the MTL framework expressed in Definition 3.14, to bound the differences between empirical and expected risks. However in Ando and Zhang (2005), Ando and et al. rely on another notion of complexity, the Rademacher Complexity Bartlett and Mendelson (2002), which measures how well a family of hypothesis can approximate random noise. The Rademacher complexity, unlike the VC-dimension, is distribution-dependent. That is the VC-dimension only uses properties of the hypothesis space $\mathcal{H}$, while the Rademacher complexity also depends on F. In first place, they use two pseudo-metrics different to those of Definitions 3.11 or 3.12 and they define the corresponding covering numbers. Then they bound the covering numbers of the set of hypothesis used in (3.8) using the Rademacher complexity of those sets of functions. However, this bounds are also dependent on the dimensions k and d, which make them unfeasible for kernel extensions.

To control the complexity or VC-dimension of the hypothesis space regularization is used, so bounds for regularized methods are also of interest.

In Maurer (2006), Maurer improves the bounds from Baxter (2000) and Ben-David and Borbely (2008); Ben-David and Schuller (2003) in the particular case of using a feature extractor g, i.e. $h_r = f_r \circ g$. Then, in Argyriou et al. (2009) an extension of the spectral theorem to vector-valued function is shown. This is useful to compute theTODO TODO: meter a maurer2009

In Pontil and Maurer (2013) bounds for trace-norm regularized methods are given.

Then, in Maurer et al. (2016) bounds for the Multi-Task Representation Learning (MTRL) Argyriou et al. (2006) are given. That is, methods where the group sparse regularization $\|W^{\intercal}\|_{2,1}$ is used. Moreover, this bounds are not limited to MTL but are also valid for the Learning to Learn paradigm.

3.4.4 Learning Under Privileged Information

The standard machine learning paradigm tries to find the hypothesis h from a set of hypothesis $\mathcal{H}$ that minimizes the expected risk $\hat{R}_z$ given a set of training samples. Vapnik is one of the main contributors to the theory of statistical learning Vapnik (2013). In this theory several important results are provided: necessary and sufficient conditions for the consistency of learning processes and bounds for the rate of convergence, which uses the notion of VC-dimension. A new inductive principle, Structural Risk Minimization (SRM), and an algorithm, Support Vector Machine (SVM), that makes use of this notion to improve the learning process.

Nowadays learning approaches based on Deep Neural Networks, which are not focused on controlling the capacity of the set of hypothesis, outperform the SVM approaches in many problems. However, these popular Deep Learning approaches require large amounts of data to learn good hypothesis. It is commonly believed that machines need much more samples to learn than humans do. Vapnik Vapnik and Izmailov (2015); Vapnik and Vashist (2009) reflects on this belief and states that humans typically learn under the supervision of an Intelligent Teacher. This Teacher shares important knowledge by providing metaphors, examples or clarifications that are helpful for the students.

LUPI Paradigm

The additional knowledge provided by the Teacher is the Privileged Information that is available only during the training stage. To incorporate the concept of Intelligent Teacher in the Machine Learning framework, Vapnik introduces the paradigm of Learning Under Privileged Information (LUPI). In the LUPI paradigm describes the following model. Given a set of i.i.d. triplets

$$z = \{(x_1, x_1^*, y_1), \dots, (x_n, x_n^*, y_n)\}, \ x \in \mathcal{X}, x^* \in \mathcal{X}^*, y \in \mathcal{Y}$$

generated according to an unknown distribution $P(x, x^*, y)$, the goal is to find the hypothesis $h(x, \alpha^*)$ from a set of hypothesis $\mathcal{H} = \{h(x, \alpha), \alpha \in A\}$ that minimizes some expected risk

$$R_F = \int \ell\left(h\left(x, \alpha\right), y\right) dF(x, y).$$

Note that the goal is the same that in the standard paradigm, however with the LUPI approach we are provided additional information, which is available only during the training stage. This additional information is encoded in the elements x^* of a space $\mathcal{X}^*$, which is different from $\mathcal{X}$. The goal of the Teacher is, given a pair (x_i, y_i), to provide a useful information $x^* \in \mathcal{X}^*$ given some probability $P(x^* \mid x)$. That is, the "intelligence" of the Teacher is defined by the choice of the space $\mathcal{X}^*$ and the conditional probability $P(x^* \mid x)$. To understand better this paradigm consider the following example.
Example. Consider that the goal is to find a decision rule that classifies biopsy images into cancer or non-cancer. Here, $\mathcal{X}$ is the space of images, i.e. the matrix of pixels, for example $[0, 1]^{64\times 64}$. The label space is $\mathcal{Y} = \{0, 1\}$. An Intelligent Teacher might provide a student of medicine with commentaries about the images, for example: "There is an area of unusual concentration of cells of Type A." or "There is an aggresive proliferation

of cells of Type B". These commentaries are the elements x^* of certain space X^* and the Teacher also chooses the probability $P(x^* \mid x)$.

Analysis of convergence rates

To get a better insight of how the Privileged Information can help in the Learning process, Vapnik provides a theoretical analysis of its influence on the learning rates. In the standard learning paradigm, how well the expected risk R_F can be bounded is controlled by two factors: the empirical risk $\hat{R}_z$ and the VC-dimension of the set of hypothesis $\mathcal{H}$. In the case of classification, where $\mathcal{Y} = \{-1, 1\}$ and the loss $\ell(h(x,\alpha), y) = \mathbf{1}_{h(x,\alpha)y \leq 0}$, the risks can be expressed as

$$\begin{aligned} R_F(\alpha) &= \int \mathbf{1}_{yh(x,\alpha)\leq 0} dF(x,y) = P(h(x,\alpha)\, y \leq 0), \\ \hat{R}_z(\alpha) &= \sum_{i=1}^{n} \mathbf{1}_{y_i h(x_i,\alpha)\leq 0} = \nu(\alpha). \end{aligned}$$

In (Vapnik, 1982, Theorem 6.8) the following bound for the rate of convergence is given with probability $1-\eta$:

$$P(h(x,\alpha_n)\, y \leq 0) \leq \nu(\alpha_n) + O\left(\frac{d \log\left(\frac{2n}{d}\right) - \log\eta}{n} \sqrt{\nu(\alpha_n) \frac{n}{d \log\left(\frac{2n}{d}\right) - \log\eta}} \right).$$

That is, the bound is controlled by the ratio d/n, where d is the VCdim$(\mathcal{H})$. If this VC-dimension is finite, the bound goes to zero as n grows. However, two different cases can be considered.

Separable case: the training data can be classified in two groups without errors. That is, there exists $\alpha_n \in \eta$ such that $y_i h(x_i, \alpha_n) > 0$ for $i = 1, \dots, n$, and thus $\nu(\alpha_n) = 0$. In this case, the following bound for the rate of converge holds

$$P(h(x,\alpha_n)\, y \leq 0) \leq O\left(\frac{d \log\left(\frac{2n}{d}\right) - \log\eta}{n} \right).$$

Non-Separable case: the training data cannot be classified in two groups without errors. That is, for all $\alpha_n \in A$, there exists $i = 1, \dots, n$, such that $y_i h(x_i, \alpha_n) \leq 0$, and thus $\nu(\alpha_n) > 0$. In this case, the following bound for the rate of converge holds

$$P(h(x,\alpha_n)\, y \leq 0) \leq \nu(\alpha_n) + O\left(\sqrt{\frac{d \log\left(\frac{2n}{d}\right) - \log\eta}{n}} \right).$$

Note that there is an important difference here in the rate of convergence. The separable case has a convergence rate of d/n, while the non-separable case has a rate of $\sqrt{d/n}$. Vapnik tries to address the question of why there exists such difference.

Oracle SVM

Vapnik tries to answer these question by looking at Support Vector Machines. In the separable case, one has to minimize the functional

$$J(w) = \|w\|^2$$

subject to the constraints

$$y_i (wx_i + b) \geq 1.$$

However, in the non-separable case the functional to minimize is

$$J(w, \xi_1, \ldots, \xi_n) = \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to the constraints

$$y_i (wx_i + b) \geq 1 - \xi_i.$$

That is, in the separable case d parameters (of w) have to be estimated using n examples, while in the non-separable case $d + n$ parameters (considering w and the slack variables $\xi_1, \ldots, \xi_n$) have to be estimated with n examples.

What would happen if the parameters $\xi_1, \ldots, \xi_n$ were known? In Vapnik and Izmailov (2015) an *Oracle* SVM is considered. Here, the learner (Student) is supplied with a set of triplets

$$(x_1, \xi_1^0, y_1), \ldots, (x_n, \xi_n^0, y_n)$$

where $\xi_1^0, \ldots, \xi_n^0$ are the slack variables for the best decision rule $h(x, \alpha_0) = \inf_{\alpha \in A} R_F(\alpha)$:

$$\xi_i^0 = \max(0, 1 - h(x, \alpha_0)), \ \forall i = 1, \ldots, n.$$

An *Oracle* SVM has to minimize the functional

$$J(w) = \|w\|^2$$

subject to the constraints

$$y_i (wx_i + b) \geq 1 - \xi_i^0.$$

Since the slack variables ξ_i^0 are known in advance, it can be shown Vapnik and Vashist (2009) that for the *Oracle* SVM the following bound holds

$$P(h(x, \alpha_n) y \leq 0) \leq P(1 - \xi^0 \leq 0) + O\left(\frac{d \log\left(\frac{2n}{d}\right) - \log \eta}{n}\right),$$

where $P(1 - \xi^0 \leq 0)$ is the probability error of the hypothesis $h(x, \alpha_0)$. That is, we recover the rate d/n.

From Oracle to Intelligent Teacher

The *Oracle* SVM is a theoretical construct, but we can approximate it by modelling the slack variables with the information provided by the Teacher in the LUPI paradigm. That is, the Teacher defines a space $\mathcal{X}^*$ and a set of functions $\{f^*(x, \alpha^*), \alpha^* \in A^*\}$. Then model the slack variables as

$$\xi^* = f^*(x^*, \alpha^*).$$

From the pairs generated by some random generator in nature, the Teacher also defines the probability $P\left(x^* \mid x\right)$ to provide the triplets

$$(x_1, x_1^*, y_1), \dots, (x_n, x_n^*, y_n).$$

Then, we can consider the problem where the goal is to minimize

$$J(\alpha, \alpha^*) = \sum_{i=1}^{n} \max(0, f^*(x_i^*, \alpha^*))$$

subject to the constraints

$$h\left(x_i, \alpha\right) \geq 1 - f^*(x_i^*, \alpha^*).$$

Let $f(x, \alpha_n), h(x.\alpha_n)$ that minimize this problem. Then, in (Vapnik and Vashist, 2009, Proposition 2) the following results for the bound of convergence is given

$$P(h\left(x, \alpha_n\right) y \leq 0) \leq P(1 - f^*(x^*, \alpha_n^*) \leq 0) + O\left(\frac{(d + d^*)\log\left(\frac{2n}{(d+d^*)}\right) - \log\eta}{n}\right),$$

where d^* is the VC-dimension of the space of hypothesis $\{f(x, \alpha^*) \in A^*\}$. This result shows that, to maintain the best convergence rate d/n, we need to estimate the $P(1 - f^*(x^*, \alpha_n^*) \leq 0)$. Although this probability is unknown, we can control it. Considering

$$\alpha_0^* = \inf_{\alpha^* \in A^*} \int_{\mathcal{X}^*} \max(0, f^*(x^*, \alpha^*) - 1) dP(x^*)$$

and

$$\alpha_n^* = \inf_{\alpha^* \in A^*} \sum_{i=1}^{n} \max(0, f^*(x_i^*, \alpha^*) - 1).$$

Consider $\{f^*(x^*, \alpha^*), \alpha^* \in A^*\}$ such that $f^*(x^*, \alpha^*) < B, \alpha^* \in A^*$, then

$$\{\max(0, f^*(x^*, \alpha^*) - 1), \alpha^* \in A^*\}$$

is a set of totally bounded non-negative functions, then we have the standard bound Vapnik (2013)

$$P(1 - f^*(x^*, \alpha_0^*) \leq 0) \leq P(1 - f^*(x^*, \alpha_n^*) \leq 0) + O\left(\sqrt{\frac{d^*\log\left(\frac{2n}{d^*}\right) - \log\eta}{n}}\right).$$

with probability $1 - 2\eta$. That is, to have a rate of d/n for α_n, we need to estimate α_n^*, which has a rate of $\sqrt{d^*/n}$. However, observe that $\mathcal{X}^*$ is the space suggested by the Teacher, which hopefully has a much lower capacity, and thus, the convergence will be faster in this space.

SVM+

Vapnik describes an extension of the SVM that embodies the LUPI paradigm Vapnik and Izmailov (2015); Vapnik and Vashist (2009). Given a set of triplets

$$(x_1, x_1^*, y_1), \dots, (x_n, x_n^*, y_n),$$

the idea is to model the slack variables of the standard SVM using the elements $x^* \in \mathcal{X}^*$ as

$$\xi(x^*, y) = [y(w^* \phi^*(x^*) + b^*)]_+ = \max\left(y(w^* \phi^*(x^*) + b^*), 0\right).$$

The minimization problem is the following:

$$\begin{aligned} \underset{w, w^*, b, b^*}{\arg\min} \quad & C \sum_{i=1}^{n} [y_i(\langle w^*, \phi^*(x_i^*)\rangle + b^*)]_+ + \frac{1}{2}\langle w, w\rangle + \frac{\mu}{2}\langle w^*, w^*\rangle \\ \text{s.t.} \quad & y_i(\langle w, \phi(x_i)\rangle + b) \geq 1 - [y_i(\langle w^*, \phi^*(x_i^*)\rangle + b^*)]_+ . \end{aligned} \tag{3.43}$$

Here ϕ and ϕ^* are two transformations that can be different. However, note that problem (3.43) is not convex due to the positive part term in the objective function. Vapnik et al. propose a relaxation of this problem to obtain a convex one. The idea is to model the slack variables ξ as

$$\xi(x^*, y) = [y(w^* \phi^*(x^*) + b^*)] + \zeta(x^*, y),$$

where $\zeta(x^*, y) \geq 0$. The minimization problem is then

$$\begin{aligned} \underset{w, w^*, b, b^*, \zeta_i}{\arg\min} \quad & C \sum_{i=1}^{n} ([y_i(\langle w^*, \phi^*(x_i^*)\rangle + b^*)] + \zeta_i) + C\Delta \sum_{i=1}^{n} \zeta_i \\ & \quad + \frac{1}{2}\langle w, w\rangle + \frac{\mu}{2}\langle w^*, w^*\rangle \\ \text{s.t.} \quad & y_i(\langle w, \phi(x_i)\rangle + b) \geq 1 - [y_i(\langle w^*, \phi^*(x_i^*)\rangle + b^*) + \zeta_i] , \\ & y_i(\langle w^*, \phi^*(x_i^*)\rangle + b^*) + \zeta_i \geq 0, \\ & \zeta_i \geq 0, \\ \text{for} \quad & i = 1, \ldots, n. \end{aligned} \tag{3.44}$$

Problem (3.44) is convex and the corresponding dual problem is

$$\begin{aligned} \underset{\alpha_i, \delta_i}{\arg\min} \quad & \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j) + \frac{1}{2\mu} \sum_{i,j=1}^{n} y_i y_j (\alpha_i - \delta_i)(\alpha_j - \delta_i) k^*(x_i^*, x_j^*) - \sum_{i=1}^{n} \alpha_i \\ \text{s.t.} \quad & 0 \leq \delta_i \leq C \\ & 0 \leq \alpha_i \leq C + \delta_i, \\ & \sum_{i=1}^{n} \delta_i y_i = 0, \ \sum_{i=1}^{n} \alpha_i y_i = 0, \\ \text{for} \quad & i = 1, \ldots, n. \end{aligned} \tag{3.45}$$

where we use the kernel functions

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle, \ k^*(x_i^*, x_j^*) = \langle \phi^*(x_i^*), \phi^*(x_j^*)\rangle.$$

We can observe in Problem (3.45) that the LUPI paradigm exerts a similarity control, correcting the similarity in space $\mathcal{X}$ with the similarity in the privileged space $\mathcal{X}^*$. For that reason, $\mathcal{X}$ and $\mathcal{X}^*$ are named Decision Space and Correction Space, respectively.

Connection between SVM+ and MTLSVM

In Liang and Cherkassky (2008) the connection between SVM+ and Multi-Task Learning SVM (MTLSVM) is discussed. The MTLSVM proposed in Liang and Cherkassky (2008) is a Multi-Task Learning model based on the SVM. It solves the primal problem

$$\begin{aligned}
&\underset{w,b,v_r,b_r,\xi_i^r}{\arg\min} && C\sum_{r=1}^{T}\sum_{i=1}^{m_r}\xi_i^r + \frac{1}{2}\langle w, w\rangle + \sum_{r=1}^{T}\frac{\mu}{2}\langle v_r, v_r\rangle \\
&\text{s.t.} && y_i^r(\langle w, \phi(x_i^r)\rangle + b + \langle v_r, \phi_r(x_i^r)\rangle + b_r) \geq 1 - \xi_i^r, \\
& && \xi_i^r \geq 0, \\
&\text{for} && r = 1, \dots, T;\ i = 1, \dots, m_r.
\end{aligned} \tag{3.46}$$

Here, a combination of a common model for all tasks

$$\langle w, \phi(x_i)\rangle + b$$

and a task-specific model

$$\langle v_r, \phi_r(x_i^r)\rangle + b_r$$

is used. Here, the common transformation ϕ and the task-independent ones ϕ_r can be different The dual problem corresponding to (3.46) is

$$\begin{aligned}
&\underset{\alpha_i}{\arg\min} && \frac{1}{2}\sum_{r,s=1}^{T}\sum_{i,j=1}^{m_r} y_i^r y_j^s \alpha_i^r \alpha_j^s k(x_i^r, x_j^s) + \frac{1}{2\mu}\sum_{r,s=1}^{T}\sum_{i,j=1}^{m_r} y_i^r y_j^s \alpha_i^r \alpha_j^s \delta_{rs} k_r(x_i^r, x_j^s) \\
& && - \sum_{r=1}^{T}\sum_{i=1}^{m_r}\alpha_i^r \\
&\text{s.t.} && 0 \leq \alpha_i^r \leq C \\
& && \sum_{i=1}^{m_r}\alpha_i^r y_i^r = 0, \\
&\text{for} && r = 1, \dots, T;\ i = 1, \dots, m_r.
\end{aligned} \tag{3.47}$$

In Liang and Cherkassky (2008) some similarities between MTLSVM and SVM+ are pointed out. Problem (3.46) can be regarded as an adaptation of (3.44) to solve MTL problems, where different tasks are incorporated and multiple correcting spaces are defined using the transformations ϕ_r. If we consider the problem (3.46) with a single task, it is a modification of the SVM+ problem (3.44) where the slack variables are modeled as

$$\xi(x, y) = y(w^*\phi^*(x) + b^*).$$

That is, it is a relaxation of the original problem (3.44) where the second constraint to model the positive part of the slack variables disappears. This relaxation gives place to some important differences between both models. Since the auxiliary primal variables ζ_i are no longer required, this is reflected in a simpler dual form (3.47), where only n dual variables have to be estimated, instead of the $2n$ dual variables of (3.45). The Multi-Task element, is reflected on (3.47) through the δ_{rs} function, which makes the correction of similarity only possible between elements of the same task. A major remark can be make about the differences between MTLSVM and SVM+. The results for the improved rate of convergence with an Intelligent Teacher may not be valid with MTLSVM, since we are not modelling the slack variables ξ adequately. It is still a work in progress to study the rate of convergence of MTLSVM and to establish more clear links with SVM+.

3.5 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

Agarwal, A., III, H. D., and Gerber, S. (2010). Learning multiple tasks using manifold regularization. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 46–54. Curran Associates, Inc.

Akhiezer, N. I. and Glazman, I. M. (1961). Theory of linear operators in hilbert space.

Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853.

Argyriou, A., Clémençon, S., and Zhang, R. (2013). Learning the graph of relations among multiple tasks. *HAL*.

Argyriou, A., Evgeniou, T., and Pontil, M. (2006). Multi-task feature learning. In Schölkopf, B., Platt, J. C., and Hofmann, T., editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 41–48. MIT Press.

Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Mach. Learn.*, 73(3):243–272.

Argyriou, A., Micchelli, C. A., and Pontil, M. (2009). When is there a representer theorem? vector versus matrix regularizers. *J. Mach. Learn. Res.*, 10:2507–2529.

Argyriou, A., Micchelli, C. A., Pontil, M., and Ying, Y. (2007). A spectral regularization framework for multi-task structure learning. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 25–32. Curran Associates, Inc.

Bakker, B. and Heskes, T. (2003). Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99.

Baldassarre, L., Rosasco, L., Barla, A., and Verri, A. (2012). Multi-output learning via spectral filtering. *Mach. Learn.*, 87(3):259–301.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482.

Barzilai, A. and Crammer, K. (2015). Convex multi-task learning by clustering. In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org.

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.

Ben-David, S. and Schuller, R. (2003). Exploiting task relatedness for mulitple task learning. In Schölkopf, B. and Warmuth, M. K., editors, *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, volume 2777 of *Lecture Notes in Computer Science*, pages 567–580. Springer.

Bonilla, E. V., Chai, K. M. A., and Williams, C. K. I. (2007). Multi-task gaussian process prediction. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 153–160. Curran Associates, Inc.

Cai, F. and Cherkassky, V. (2009). SVM+ regression and multi-task learning. In *International Joint Conference on Neural Networks, IJCNN 2009, Atlanta, Georgia, USA, 14-19 June 2009*, pages 418–424. IEEE Computer Society.

Cai, F. and Cherkassky, V. (2012). Generalized SMO algorithm for svm-based multitask learning. *IEEE Trans. Neural Networks Learn. Syst.*, 23(6):997–1003.

Caruana, R. (1997). Multitask learning. *Mach. Learn.*, 28(1):41–75.

Chen, J., Liu, J., and Ye, J. (2010). Learning incoherent sparse and low-rank patterns from multiple tasks. In Rao, B., Krishnapuram, B., Tomkins, A., and Yang, Q., editors, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 1179–1188. ACM.

Chen, J., Tang, L., Liu, J., and Ye, J. (2009). A convex formulation for learning shared structures from multiple tasks. In Danyluk, A. P., Bottou, L., and Littman, M. L., editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 137–144. ACM.

Chen, J., Zhou, J., and Ye, J. (2011). Integrating low-rank and group-sparse structures for robust multi-task learning. In Apté, C., Ghosh, J., and Smyth, P., editors, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 42–50. ACM.

Ciliberto, C., Mroueh, Y., Poggio, T. A., and Rosasco, L. (2015). Convex learning of multiple tasks and their structure. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1548–1557. JMLR.org.

Crammer, K. and Mansour, Y. (2012). Learning multiple tasks using shared hypotheses. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1484–1492.

Dinuzzo, F. (2013). Learning output kernels for multi-task problems. *Neurocomputing*, 118:119–126.

Evgeniou, T., Micchelli, C. A., and Pontil, M. (2005). Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637.

Evgeniou, T. and Pontil, M. (2004). Regularized multi–task learning. In Kim, W., Kohavi, R., Gehrke, J., and DuMouchel, W., editors, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 109–117. ACM.

Fung, G. and Mangasarian, O. L. (2001). Proximal support vector machine classifiers. In Lee, D., Schkolnick, M., Provost, F. J., and Srikant, R., editors, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, pages 77–86. ACM.

Gong, P., Ye, J., and Zhang, C. (2012a). Multi-stage multi-task feature learning. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1997–2005.

Gong, P., Ye, J., and Zhang, C. (2012b). Robust multi-task feature learning. In Yang, Q., Agarwal, D., and Pei, J., editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 895–903. ACM.

Han, L. and Zhang, Y. (2015). Learning tree structure in multi-task learning. In Cao, L., Zhang, C., Joachims, T., Webb, G. I., Margineantu, D. D., and Williams, G., editors, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 397–406. ACM.

Han, L. and Zhang, Y. (2016). Multi-stage multi-task learning with reduced rank. In Schuurmans, D. and Wellman, M. P., editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1638–1644. AAAI Press.

Hein, M., Bousquet, O., Hein, M., and Bousquet, O. (2004). Kernels, associated structures and generalizations.

Hernández-Lobato, D. and Hernández-Lobato, J. M. (2013). Learning feature selection dependencies in multi-task learning. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 746–754.

Hernández-Lobato, D., Hernández-Lobato, J. M., and Ghahramani, Z. (2015). A probabilistic model for dirty multi-task feature selection. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1073–1082. JMLR.org.

III, H. D. (2009). Bayesian multitask learning with latent hierarchies. In Bilmes, J. A. and Ng, A. Y., editors, *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pages 135–142. AUAI Press.

Jacob, L., Bach, F. R., and Vert, J. (2008). Clustered multi-task learning: A convex formulation. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 745–752. Curran Associates, Inc.

Jalali, A., Ravikumar, P., Sanghavi, S., and Ruan, C. (2010). A dirty model for multi-task learning. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 964–972. Curran Associates, Inc.

Jawanpuria, P. and Nath, J. S. (2012). A convex feature learning formulation for latent task structure discovery. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

Jebara, T. (2004). Multi-task feature and kernel selection for svms. In Brodley, C. E., editor, *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM.

Jebara, T. (2011). Multitask sparsity via maximum entropy discrimination. *J. Mach. Learn. Res.*, 12:75–110.

Jeong, J. and Jun, C. (2018). Variable selection and task grouping for multi-task learning. In Guo, Y. and Farooq, F., editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1589–1598. ACM.

Kang, Z., Grauman, K., and Sha, F. (2011). Learning with whom to share in multi-task feature learning. In Getoor, L. and Scheffer, T., editors, *Proceedings of the 28th*

International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, pages 521–528. Omnipress.

Kumar, A. and III, H. D. (2012). Learning task grouping and overlap in multi-task learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

Lawrence, N. D. and Platt, J. C. (2004). Learning to learn with the informative vector machine. In Brodley, C. E., editor, *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM.

Li, Y., Tian, X., Song, M., and Tao, D. (2015). Multi-task proximal support vector machine. *Pattern Recognit.*, 48(10):3249–3257.

Liang, L. and Cherkassky, V. (2008). Connection between SVM+ and multi-task learning. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*, pages 2048–2054. IEEE.

Liu, H., Palatucci, M., and Zhang, J. (2009). Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In Danyluk, A. P., Bottou, L., and Littman, M. L., editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 649–656. ACM.

Lozano, A. C. and Swirszcz, G. (2012). Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

Maurer, A. (2006). Bounds for linear multi-task learning. *J. Mach. Learn. Res.*, 7:117–139.

Maurer, A. (2009). Transfer bounds for linear feature learning. *Mach. Learn.*, 75(3):327–350.

Maurer, A. and Pontil, M. (2010). K -dimensional coding schemes in hilbert spaces. *IEEE Trans. Inf. Theory*, 56(11):5839–5846.

Maurer, A., Pontil, M., and Romera-Paredes, B. (2013). Sparse coding for multitask and transfer learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 343–351. JMLR.org.

Maurer, A., Pontil, M., and Romera-Paredes, B. (2016). The benefit of multitask representation learning. *J. Mach. Learn. Res.*, 17:81:1–81:32.

Micchelli, C. A. and Pontil, M. (2005). On learning vector-valued functions. *Neural Comput.*, 17(1):177–204.

Obozinski, G., Taskar, B., and Jordan, M. (2006). Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep*, 2(2.2):2.

Parameswaran, S. and Weinberger, K. Q. (2010). Large margin multi-task metric learning. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 1867–1875. Curran Associates, Inc.

Pong, T. K., Tseng, P., Ji, S., and Ye, J. (2010). Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM J. Optim.*, 20(6):3465–3489.

Pontil, M. and Maurer, A. (2013). Excess risk bounds for multitask learning with trace norm regularization. In Shalev-Shwartz, S. and Steinwart, I., editors, *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 55–76. JMLR.org.

Suykens, J. A. K. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Process. Lett.*, 9(3):293–300.

Thrun, S. and O'Sullivan, J. (1996). Discovering structure in multiple learning tasks: The TC algorithm. In Saitta, L., editor, *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3-6, 1996*, pages 489–497. Morgan Kaufmann.

Vapnik, V. (1982). *Estimation of dependences based on empirical data.* Springer Science & Business Media.

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

Vapnik, V. and Izmailov, R. (2015). Learning using privileged information: similarity control and knowledge transfer. *J. Mach. Learn. Res.*, 16:2023–2049.

Vapnik, V. and Vashist, A. (2009). A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557.

Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244.

Xu, S., An, X., Qiao, X., and Zhu, L. (2014). Multi-task least-squares support vector machines. *Multim. Tools Appl.*, 71(2):699–715.

Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. (2007). Multi-task learning for classification with dirichlet process priors. *J. Mach. Learn. Res.*, 8:35–63.

Yu, K., Tresp, V., and Schwaighofer, A. (2005). Learning gaussian processes from multiple tasks. In Raedt, L. D. and Wrobel, S., editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 1012–1019. ACM.

Zhang, Y. and Yeung, D. (2010). A convex formulation for learning task relationships in multi-task learning. In Grünwald, P. and Spirtes, P., editors, *UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, July 8-11, 2010*, pages 733–442. AUAI Press.

Zhang, Y. and Yeung, D. (2013). A regularization approach to learning task relationships in multitask learning. *ACM Trans. Knowl. Discov. Data*, 8(3):12:1–12:31.

Zhang, Y., Yeung, D., and Xu, Q. (2010). Probabilistic multi-task feature selection. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 2559–2567. Curran Associates, Inc.

Zhou, J., Chen, J., and Ye, J. (2011). Clustered multi-task learning via alternating structure optimization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F. C. N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 702–710.

Zweig, A. and Weinshall, D. (2013). Hierarchical regularization cascade for joint learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 37–45. JMLR.org.