

# Advanced Kernel Methods for Multi-Task Learning

Tesis dirigida por José Dorronsoro y Carlos Alaíz

**Carlos Ruiz Pastor** 

January 3, 2023





Acknowledgements 1

Acknowledgements 2



## **Outline**

0

- ► Introduction

  Multi-Task Learning

  Support Vector Machines
- ➤ A Convex Formulation for Multi-Task Learning Convex Multi-Task Learning with Kernel Methods Convex Multi-Task Learning with Neural Networks
- ► Adaptive Graph Laplacian for Multi-Task Learning Graph Laplacian Multi-Task Learning with Kernels Adaptive Graph Laplacian Algorithm
- ► Experiments
- Summary



1 Introduction

- ► Introduction Multi-Task Learning Support Vector Machines
- A Convex Formulation for Multi-Task Learning
   Convex Multi-Task Learning with Kernel Methods
   Convex Multi-Task Learning with Neural Networks
- Adaptive Graph Laplacian for Multi-Task Learning Graph Laplacian Multi-Task Learning with Kernels Adaptive Graph Laplacian Algorithm
- Experiments
- Summary



# **Introduction to Machine Learning**

1 Introduction

- Machine Learning tries to automatize the learning process
- In the supervised setting we have:
  - An input space  $\mathcal{X}$ ,
  - an output space  $\mathcal{Y}$ ,
  - and the (unknown) probability P(x, y) over  $\mathcal{X} \times \mathcal{Y}$
- Given a function  $f: \mathcal{X} \to \mathcal{Y}$ , we define the loss function:

$$\ell: \mathcal{Y} imes \mathcal{Y} 
ightarrow [0, \infty) \ (\gamma, f(x)) 
ightarrow \ell(\gamma, f(x)),$$

such that  $\ell(y,y) = 0$  for all  $y \in \mathcal{Y}$ 



#### **Loss Functions**

1 Introduction

• In classification, with the class labels  $y_i \in \{-1, 1\}$ , we can use:

$$\ell(y, f(x)) = [1 - yf(x)]_{+} = \begin{cases} 0, & yf(x) \ge 1, \\ 1 - yf(x), & yf(x) < 1. \end{cases} \tag{1}$$



# **Expected Risk**

1 Introduction

- Given a space of hypothesis  $\mathcal{H} = \{h(\cdot, \alpha), \alpha \in A\}$
- Definition: Expected Risk

$$R_{P}(\alpha) = \int_{\mathcal{X} \times \mathcal{Y}} \ell(\mathbf{y}, h(\mathbf{x}, \alpha)) dP(\mathbf{x}, \mathbf{y})$$

Our goal is to find

$$lpha^* = \operatorname*{arg\,min}_{lpha \in A} \left\{ R_P(lpha) = \int_{\mathcal{X} imes \mathcal{Y}} \ell(\mathbf{y}, \mathbf{h}\,(\mathbf{x}, lpha)) dP(\mathbf{x}, \mathbf{y}) 
ight\},$$

however the distribution P(x, y) is unknown



# **Empirical Risk**

#### 1 Introduction

• Instead, we have a set of n instances sampled from P(x, y):

$$D_n = \{(x_i, y_i), i = 1, \ldots, n\},\$$

• Definition: Empirical Risk

$$\hat{R}_{D_n}(\alpha) = \frac{1}{n} \sum_{i=1}^n \ell(\gamma_i, h(x_i, \alpha))$$

Instead of the Expected Risk, we minimize this empirical risk:

$$\underset{\alpha \in A}{\operatorname{arg\,min}} \left\{ \hat{R}_{D}(\alpha) = \frac{1}{n} \sum_{i=1}^{n} \ell(\gamma_{i}, h(\mathbf{x}_{i}, \alpha)) \right\}$$



# **Multi-Task Learning**

1 Introduction



2 A Convex Formulation for Multi-Task Learning

- IntroductionMulti-Task LearningSupport Vector Machines
- ► A Convex Formulation for Multi-Task Learning Convex Multi-Task Learning with Kernel Methods Convex Multi-Task Learning with Neural Networks
- Adaptive Graph Laplacian for Multi-Task Learning Graph Laplacian Multi-Task Learning with Kernels Adaptive Graph Laplacian Algorithm
- Experiments
- Summary



3 Adaptive Graph Laplacian for Multi-Task Learning

- ➤ Introduction

  Multi-Task Learning

  Support Vector Machines
- A Convex Formulation for Multi-Task Learning
   Convex Multi-Task Learning with Kernel Methods
   Convex Multi-Task Learning with Neural Networks
- ► Adaptive Graph Laplacian for Multi-Task Learning Graph Laplacian Multi-Task Learning with Kernels Adaptive Graph Laplacian Algorithm
- Experiments
- Summary



#### 4 Experiments

- ► Introduction

  Multi-Task Learning

  Support Vector Machine
- A Convex Formulation for Multi-Task Learning
   Convex Multi-Task Learning with Kernel Methods
   Convex Multi-Task Learning with Neural Networks
- Adaptive Graph Laplacian for Multi-Task Learning Graph Laplacian Multi-Task Learning with Kernels Adaptive Graph Laplacian Algorithm

#### **►** Experiments

Summary



5 Summary

- ► Introduction

  Multi-Task Learning

  Support Vector Machine
- A Convex Formulation for Multi-Task Learning
   Convex Multi-Task Learning with Kernel Methods
   Convex Multi-Task Learning with Neural Network
- Adaptive Graph Laplacian for Multi-Task Learning Graph Laplacian Multi-Task Learning with Kernels Adaptive Graph Laplacian Algorithm
- Experiments
- ▶ Summary



# **Good Luck!**

5 Summary

• Enough for an introduction! You should know enough by now



# Advanced Kernel Methods for Multi-Task Learning Thank you for listening!

Any questions?