2021

# Balancing Fit And Complexity In Learned Representations

Maria Peifer
*University of Pennsylvania*

# Balancing Fit And Complexity In Learned Representations


## Abstract

This dissertation is about learning representations of functions while restricting complexity. In machine learning, maximizing the fit and minimizing the complexity are two conflicting objectives. Common approaches to this problem involve solving a regularized empirical minimization problem, with a complexity measure regularizer and a regularizing parameter that controls the trade-off between the two objectives. The regularizing parameter has to be tuned by repeatedly solving the problem and does not have a straightforward interpretation. This work formulates the problem as a minimization of the complexity measure subject to the fit constraints.The issue of complexity is tackled in reproducing kernel Hilbert spaces (RKHSs) by introducing a novel integral representation of a family of RKHSs that allows arbitrarily placed kernels of different widths. The functional estimation problem is then written as a sparse functional problem, which despite being non-convex and infinite-dimensional can be solved in the dual domain. This problem achieves representations of lower complexity than traditional methods because it searches over a family of RKHS rather than a subspace of a single RKHS. The integral representation is used in a federated classification setting, in which a global model is trained from a federation of agents. This is possible because the dual optimal variables give information about the samples that are fundamental to the classification. Each agent, therefore, learns a local model and sends only the fundamental samples over the network. This creates a federated learning method that requires only one network communication. Its solution is proven to asymptotically converges to that of traditional classification. Next, a theory for constraint specification is established. An optimization problem with a constraint for each sample point can easily become infeasible if the constraints are too tight. In contrast, relaxing all constraints can cause the solution to not fit the data well. The constrained specification method relaxes the constraints until the marginal cost of changing a constraint is equal to the marginal complexity measure. This problem is proven to be feasible and solvable and shown empirically to be resilient to outliers and corrupted training data.



## Degree Type

Dissertation

## Degree Name

Doctor of Philosophy (PhD)

## Graduate Group

Electrical & Systems Engineering

## First Advisor

Alejandro Ribeiro

## Keywords

Corrupted data, Resilient learning, RKHS, Sparsity, Statistical learning

## Subject Categories

Applied Mathematics | Artificial Intelligence and Robotics | Electrical and Electronics

# BALANCING FIT AND COMPLEXITY IN LEARNED REPRESENTATIONS

Maria Peifer

A DISSERTATION

in

Electrical and System Engineering

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2021

Supervisor of Dissertation

---

Alejandro Ribeiro, Professor of Electrical and System Engineering

Graduate Group Chairperson

---

Victor Preciado, Associate Professor of Electrical and Systems Engineering

Dissertation Committee

Hamed Hassani, Assistant Professor of Electrical and Systems Engineering

Konstantinos Daniilidis, Ruth Yalom Stone Professor of Computer and Information Science

Brian Sadler, Senior Research Scientist for Intelligent Systems at Army Research Labs

BALANCING FIT AND COMPLEXITY IN LEARNED REPRESENTATIONS

# ACKNOWLEDGEMENT

I would like to take this opportunity to thank my advisor, my committee, my professors, my peers and my family who have made my research possible. First, I would like to thank my advisor, Alejandro Ribeiro, for giving me a chance when I struggled the most. I'm especially grateful for being supportive and always challenging me to be better and not overthink the little things.

I am fortunate to say that everything relating to my dissertation committee, from scheduling meetings to getting responses to questions to getting useful advice, was the easiest part of my journey. I would like to thank Hamed Hassani, Kostas Daniilidis, and Brian Sadler for being part of the committee.

I would also like to thank my colleagues who have made the time working on my doctorate an enjoyable experience: Luiz Chamon, Santiago Paternain, Mark Eisen, Fernando Gama, Ling Phuong, Markos Epitropou, Ling Phuong, Kate Tolstaya, Luana Ruiz, Muigel Calvo-Fullana, Mahyar Fazylab, Mohammad Fereydounian, Harshat Kumar, Vinicius Lima, Zhiyang Wang, Juan Cerviño, Ximing Chen, Weyiu Huang. With some of you I have collaborated, with others I made it through the qualifiers, had awesome lunches and enjoyed great conversations. I am incredibly thankful to have met all of you.

I am also grateful from my friends outside of academia that have been supportive and understanding. These people have enriched my life outside of research and have helped me keep a necessary work-life balance.

Perhaps most importantly, I am grateful for the love and support of my family. I want to thank my sister Luciana for teaching me how to read while reading me stories. I want to thank my parents, who have instilled in me from an early age the love of learning. I want to thank my in-laws for being supportive of my studies. I want to thank naṣa and naṣu for being there with wise advice. Most of all, I want to thank my husband Michael for being loving ad supportive through this entire process.

# ABSTRACT

## BALANCING FIT AND COMPLEXITY IN LEARNED REPRESENTATIONS

Maria Peifer

Alejandro Ribeiro

This dissertation is about learning representations of functions while restricting complexity. In machine learning, maximizing the fit and minimizing the complexity are two conflicting objectives. Common approaches to this problem involve solving a regularized empirical minimization problem, with a complexity measure regularizer and a regularizing parameter that controls the trade-off between the two objectives. The regularizing parameter has to be tuned by repeatedly solving the problem and does not have a straightforward interpretation. This work formulates the problem as a minimization of the complexity measure subject to the fit constraints. The issue of complexity is tackled in reproducing kernel Hilbert spaces (RKHSs) by introducing a novel integral representation of a family of RKHSs that allows arbitrarily placed kernels of different widths. The functional estimation problem is then written as a sparse functional problem, which despite being non-convex and infinite-dimensional can be solved in the dual domain. This problem achieves representations of lower complexity than traditional methods because it searches over a family of RKHS rather than a subspace of a single RKHS. The integral representation is used in a federated classification setting, in which a global model is trained from a federation of agents. This is possible because the dual optimal variables give information about the samples that are fundamental to the classification. Each agent, therefore, learns a local model and sends only the fundamental samples over the network. This creates a federated learning method that requires only one network communication. Its solution is proven to asymptotically converges to that of traditional classification. Next, a theory for constraint specification is established. An optimization problem with a constraint for each sample point can easily become infeasible if the constraints are too tight. In contrast, relaxing all constraints can cause the solution to not fit the data

well. The constrained specification method relaxes the constraints until the marginal cost of changing a constraint is equal to the marginal complexity measure. This problem is proven to be feasible and solvable, and shown empirically to be resilient to outliers and corrupted training data.

TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF ILLUSTRATIONS

# CHAPTER 1

# Introduction

## 1.1 Motivation

Machine learning has been an integral part of modern day life. Advances in image recognition He et al. (2016), speech processing Benesty et al. (2008), language translation Hermann and Blunsom (2013); Bahdanau et al. (2014), and language interpretation have led to the automation of job advertisement Poch et al. (2014), candidates selection Erel et al. (2018), medical data analysis Erickson et al. (2017), and "smart" appliances. These methods find a representation with the best fit over a distribution by solving an empirical risk minimization problem to approximate the statistical risk minimization problem, when the underlying distribution is unknown Vapnik (2013); Shalev-Shwartz and Ben-David (2014). The empirical risk is a good approximation when large amounts of data are available, and can find a representation which fits the data well.

Oftentimes, in addition to finding representations that fit the data well, we are interested in our representations having a particular property. Limiting the complexity of the representation ensures that the solution is unique Boyd and Vandenberghe (2004); Schölkopf and Smola (2001). The property promoted can be smoothness in order to improve generalization and avoid overfitting Schölkopf et al. (2001); Boyd and Vandenberghe (2004). Improving the sparsity of a representation can reduce computational complexity Zhang et al. (2015); Tibshirani (1996); Bickel et al. (2009) or make it easier to be shared. For example, in the case of federated learning, system limitations restrict the amount of information to be sent over the network Konečnỳ et al. (2016b); Smith et al. (2017); Bonawitz et al. (2019).

Optimizing the fit and the complexity of the representation, leads to two competing objectives that need to be accomplished. A popular solution is to include the measure of fit and the

measure of complexity in the objective of the problem and use regularization to control the importance of each measure Zhang et al. (2015); Tibshirani (1996); Bickel et al. (2009); Goodfellow et al. (2014); Berk et al. (2017); Xu et al. (2018). The choice of the regularization parameter is not straightforward, since the two objectives often times have a different scale. Furthermore, the regularization parameter does not have a clear interpretation, therefore, it cannot be learned and is tuned by repeatedly training the model with different values for the parameter and selecting the value that performs best on a separate evaluation data set Hsu et al. (2003). Tuning parameters can be costly, especially with algorithms that require a lot of resources for training.

Minimizing the empirical risk assigns the same importance to each collected sample. This assumes that the distribution of the training data set is the same as the distribution of the predictive data, however, the sampling of the data may not reflect the true distribution Vapnik (2013). There can be biases (ex. gender bias, racial bias) in the training data Kodiyan (2019); Datta et al. (2015); Kay et al. (2015) or adversarial examples Lowd and Meek (2005); Gu et al. (2019); Shen and Sanghavi (2019), which can affect the model. Therefore, there is a need for a method of optimizing the fit which takes into account imbalances in the data.As the sample size grows, it becomes impractical to find the best weighting for the sample losses via grid search. This work proposes to learn the best representation by encoding the losses in constraints. The constraints not only have a straightforward interpretation, but also allow the model to fit some samples better than others. This is particularly advantageous when there are corrupted samples or outliers in the training set.

## 1.2 Statistical Learning

Statistical learning theory provides a framework for learning predictive models based on data Hastie et al. (2009). Let $\mathbf{x} \in \mathbb{R}^p$ and $y \in \mathbb{R}$ be random variables with a joint probability distribution, $\mathcal{D}$, and let $\phi : \mathbb{R}^p \to \mathbb{R}$ be a function from a functional space $\mathcal{F}$. The random variable $\mathbf{x}$ can be thought of as an independent variable, while $y$ can be thought of as the target variable. Given a loss function $\ell : \mathbb{R} \to \mathbb{R}$, which measures the fit of the function

$\phi$, the statistical risk of the function is measured by the expected loss over the distribution $\mathcal{D}$ Vapnik (2013):

$$R(\phi) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\ell(\phi(\mathbf{x}), y)\Big]. \tag{1.1}$$

The function $\phi$ that is best at predicting $y$ from the independent variable $\mathbf{x}$ is the function with the lowest risk Vapnik (2013). From this observation, the statistical risk minimization problem is formulated

$$\phi^* = \underset{\phi\in\mathcal{F}}{\operatorname{argmin}} \quad \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\ell(\phi(\mathbf{x}), y)\Big]. \tag{P-SRM}$$

The statistical risk minimization is problem finds the function $\phi$ that can best generalize to samples $\mathbf{x}$ taken from distribution $\mathcal{D}$. However, it has some major drawbacks. The distribution $\mathcal{D}$ is generally not known Vapnik (1992). Instead, a set of $N$ sample pairs $(\mathbf{x}_n, y_n)$ is available, that is assumed to be taken at random from the distribution $\mathcal{D}$. Then the statistical risk is replaced with the empirical risk,

$$R_{emp}(\phi) = \frac{1}{N}\sum_{i=1}^{N} \ell(\phi(\mathbf{x}_n), y_n). \tag{1.2}$$

The empirical risk approximates the statistical risk by averaging over the training samples. From the law of large numbers, it is known that the empirical risk converges asymptotically converges to the statistical risk as the number of samples grows with $O(\sqrt{(N)})$ Hsu and Robbins (1947); Koltchinskii et al. (2002). More than that, even stricter bounds have been established for the solution of the empirical risk minimization problem Koltchinskii et al. (2002); Lugosi et al. (2004)

$$\phi^*_{emp} = \underset{\phi\in\mathcal{F}}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} \ell(\phi(\mathbf{x}_n), y_n) \tag{P-ERM}$$

The problem (P-ERM) works well in practice when two assumptions hold (Vapnik, 2013, Chapter 2): the sample pairs $(\mathbf{x}, y_n)$ are derived from the distribution $\mathcal{D}$ and the sample size is sufficiently large in order for the empirical risk to be a good approximation of the statistical risk. In order to evaluate the representation, $\phi^*_{emp}$ a separate set $S = \{(\mathbf{x}_1, y_1), \ldots (\mathbf{x}_T, y_T)\}$

of samples is needed, for which the samples are taken from the distribution $\mathcal{D}$. This is called an *evaluation set*. The empirical risk is calculated on the evaluation set. If it is observed that the evaluation empirical risk is close to the training empirical risk, then the function $\phi^*_{emp}$ is considered to generalize well to unseen samples Vapnik (2013).

Although, these conditions stated above are sufficient for the problem to generalize well, finding its solution is not straightforward. In fact, searching over very rich functional spaces involves solving infinite dimensional problems and is therefore intractable, e.g. ($\phi \in L_2$). To remediate this, the functional spaces are typically restricted to spaces that can be characterized by a parameter such as the linear space, space spanned by a group of kernels, etc. (Schwartz, 1969, Chapter 1) (Engl et al., 1996, Chapter 9). Although, parametrized functional spaces make the problem easier to optimize, there are examples of functional risk minimization problems for which a solution can be found Chamon et al. (2018); Peifer et al. (2020)

Although the empirical risk minimization problem can be solved, the solution is often not unique. In fact, there are in general an infinite number of functions that can interpolate a finite number of points. Of course, depending on the richness of the functional space $\mathcal{F}$ chosen, none of these interpolations could be included in $\mathcal{F}$. Nonetheless, in many cases, the solution is not unique. In order to pick a solution, a regularizer is added to problem (P-ERM). Specifically, given the set of observation-target pairs $(\mathbf{x}_n, y_n)$, with $\mathbf{x}_n \in \mathbb{R}^p$ and $y_n \in \mathbb{R}$, the regularized empirical risk minimization problem is typically formulated

$$\phi^*_{remp} = \underset{\phi \in \mathcal{F}}{\operatorname{argmin}} \quad \frac{1}{N} \sum_{n=1}^{N} \ell(\phi(x_n), y_n) + \gamma \rho(\phi), \tag{P-rERM}$$

for which $\ell$ represents the loss function, that measures the fit, and $\rho$ is a function that measures the complexity of the representation; $\gamma$ is called the regularizing parameter and is used to control the trade-off between optimizing the fit and reducing complexity of the representation. The measure of complexity, $\rho$, can be a measure of smoothness, e.g. $\ell_2$-norm, which can be used to enhance robustness to noise, and can improve the numerical

properties of the optimization problem Peifer et al. (2020). It can also be used to control the computational complexity of the optimal representation, e.g. $\ell_0$-norm, $\ell_1$-norm Zhang et al. (2015); Tibshirani (1996); Bickel et al. (2009).

Problem (P-rERM) attempts to optimize two conflicting objectives by minimizing a weighted sum of the two. However, the two objectives are differently scaled and require a careful choice of $\gamma$. The regularizing parameter is typically tuned vie grid search, which involves repeatedly optimizing the problem using different values of $\gamma$, using a separate evaluation set to test the performance of the resulting representation and only then selecting the best value for $\gamma$ based on the performance on the evaluation set Hsu et al. (2003). This method can become costly, especially when optimizing a single instance of problem (P-rERM) is computationally expensive.

The problem presented in (P-rERM) minimizes the average loss under the assumption that the training samples reflect the distribution $\mathcal{D}$. This might not always be the case and the data could be corrupted by outliers or miss-classified observations or wrong measurements which can skew the model. Therefore, a formulation which poses the (P-rERM) as a feasibility problem which optimizes the complexity measure $\rho$ is proposed

$$\begin{aligned} \phi = \underset{\phi \in \mathcal{F}}{\text{minimize}} \quad & \rho(\phi) \\ \text{subject to} \quad & \ell(\phi(x_n), y_n) \leq \epsilon_n, \quad n = 1, \ldots, N. \end{aligned} \tag{P-f}$$

In this formulation $\epsilon_n$ represents the slack variable, i.e. the amount of training error allowed. The separate slack variables for each sample allows for the algorithm to set individual specifications for each sample and potentially ignore outliers and wrong measurements. The issue with this formulation is that although we have the ability to set individual specifications, setting the constraints is not a straightforward task. As the number of samples grows, it becomes increasingly more difficult to set the constraints, such that the problem is feasible. Furthermore, when the problem is infeasible it is unclear which constraints should be relaxed, since constraints interact with each other.

### 1.2.1 Learning Parsimonious Representations in Reproducing Kernel Hilbert Spaces

Reproducing kernel Hilbert Spaces (RKHSs) are complete, linear functional spaces that are endowed with a unique kernel and a unique inner product. These are rich functional spaces that have a parametrized representation. Indeed, functions in RKHSs can be represented as a possibly infinite weighted sum of the kernels. Moreover, kernels are often defined by a parameter that dictates the smoothness of the function. Examples include the scale or variance of Gaussian kernels or the bandwidth of sinc functions. Formally, given an RKHS $\mathcal{H}$, a function $\phi \in \mathcal{H}_0$ can be represented as

$$\phi(\mathbf{x}) = \lim_{n\to\infty} \sum_{j=1}^{n} a_j k(\mathbf{x}, \mathbf{z}_j \,;\, w_0), \tag{1.3}$$

where $w_0$ is the kernel parameter, $k(\cdot, \cdot; w_0)$ is the kernel and $a_j$ is the weight. The variable $\mathbf{z}_j$ is called the kernel center. Although, the definition of $\phi$ spans the entire functional space, it is defined by the parameter vector $\mathbf{a} = [a_1, a_2, \dots]$. Moreover, for complexity measures $\rho(g_m(\|\phi\|_{\mathcal{H}_0}))$, where $\|\cdot\|_{\mathcal{H}_0}$ is the functional norm, and $g_m(\cdot)$ is a monotonically non-decreasing function, the solution (P-rERM) and implicitly to (P-f) for appropriately chosen $\epsilon_n$ has a finite representation of the form

$$\phi^\star(\cdot) = \sum_{n=1}^{N} a_n^\star k(\cdot, \mathbf{x}_n \,;\, w_0). \tag{1.4}$$

The solution is not only finite, but it also admits kernels centered at the sample points $\mathbf{x}_n$. Unfortunately, This important result, called the representer theorem, does not hold for complexity measures which promote sparsity (see Remark 1). Furthermore, for functions with varying degrees of smoothness, i.e. functions that vary fast in certain parts of the domain but are smooth in others it is a well known result that these problems require many samples Donoho and Johnstone (1998).

### 1.2.2 Federated Classification using Sparse Representations

Federated learning involves learning a global model from data collected by a federation of agents Konečnỳ et al. (2016a,b); Li et al. (2020); McMahan et al. (2016). These types of problems naturally arise from large amounts of data being collected over distributed networks of devices such as mobile phones, wearable devices, or autonomous vehicles Smith et al. (2017); Anguita et al. (2013). Unlike traditional learning, in which data is collected and pooled together for a model to be learned by a central server, in federated learning, each agent is associated with a user and therefore, the user privacy needs to be taken into account before sharing data Konečnỳ et al. (2015, 2016b); Smith et al. (2017); Zhao et al. (2018); Bonawitz et al. (2019). Moreover, even if privacy is of no concern, there are system limitations which need to be taken into account. As the number of agent grows, it becomes increasingly more difficult to share the data over the network Konečnỳ et al. (2015, 2016b); Smith et al. (2017); Zhao et al. (2018); Bonawitz et al. (2019).

The system challenges, make the centralized learner an impractical solution to federated learning, and requires agents to perform some computations and transmit only data that is necessary for forming a global model to the central server. The communication limitations of the system lead to additional statistical challenges Konečnỳ et al. (2015, 2016b); Smith et al. (2017); Zhao et al. (2018). Because the central server no longer has access to the entire dataset, forming a global model based on the global distribution is no longer a straightforward task.

Federated learning has been approached in a distributed fashion, by sharing the gradient over the network rather than training data McMahan and Ramage (2017); Hard et al. (2018); McMahan et al. (2016). While these methods preserve privacy, they are communication intensive and require all agents to send data over the network at each iteration. More than that, these methods do not guarantee that their solution to generalize well over the global distribution. In fact, as federated learning methods get more communication efficient Wang et al. (2019); Yu et al. (2019); Shokri and Shmatikov (2015), the global method is less

guaranteed to tackle the statistical challenge.

### 1.2.3 Resilient Learning for Balancing Fit and Complexity

In the previous sections, a problem of the form (P-f) was presented as the solution under the assumption that $\epsilon_n$ was chosen such that there exists at least a $\phi$ such that the constraints are met. When this is the case, the problem is solved in the dual domain. Solving the dual problem is akin to solving a regularized problem for which the dual optimizers are the regularizing parameters. Since the optimal dual maximizers are a measure of the difficulty of fitting a constraint Chamon et al. (2020), problem (P-f) solves a regularized problem for which the regularizing parameters are proportional to the difficulty of meeting each constraint. This is advantageous for fitting unusual samples that can be overlooked by the empirical formulation, therefore, (P-f) can be considered a *robust* formulation.

The drawback to the *robust* formulation is that it is susceptible to overfit to outliers and corrupted data. Moreover, different from the constraint learning problem (P-rERM) for which a solution always exists, this might not be the case for the constrained learning problem. In order to ensure the fit of the solution $\epsilon_n$ needs to have a small value. Otherwise, the problem will pick the function with the lowest complexity. When $\epsilon_n$ is small, however, feasibility is no longer guaranteed. In practice, it is likely that some constraints need to be relaxed in order for the problem to become feasible

$$\begin{aligned} \phi_{\text{RES}} \;=\; & \underset{\phi}{\text{argmin}} \quad \rho(\phi), \\ & \text{s.t.} \qquad \ell(\phi(\mathbf{x}_n), y_n) - \epsilon_n \leq w(\mathbf{x}_n), \quad n = 1 \dots N, \end{aligned} \tag{P-RES}$$

where the function $w(\mathbf{x}_n)$ represents the relaxation function. When $w(\mathbf{x}_n) = 0$ the original problem (P-f) is recovered. In order to balance the fit and the complexity of the representation, the function $w$ is chosen for each $\mathbf{x}_n$ such that the marginal cost of relaxing the constraint is equal to the marginal change in the complexity measure.

## 1.3 Objectives and Contributions

The goal of this work is to provide a framework for balancing the fit and complexity objectives when learning representations. The framework will formulate the problem as a constrained learning problem which minimizes the complexity objective subject to a set of fit constraints. Using constraints to find a representation with a good fit has two major advantages: the dual gives provides information on how difficult it is to fit the representation to each sample, it allows for constraints to be relaxed for each sample individually.

The first part of the dissertation focuses on finding representations of low complexity in reproducing kernel Hilbert spaces (RKHS). RKHSs are non-parametric techniques popular in machine learning, signal processing, and statistics Schölkopf and Smola (2001); Bishop (2006); Hofmann et al. (2008); Yuan et al. (2010); Berlinet and Thomas-Agnan (2011); Arenas-Garcia et al. (2013); Koppel et al. (January 2019). They are of interest because of their capabilities to fit many functions while also admitting a straightforward representation. Indeed, functions in RKHS can be represented as a possibly infinite linear combination of basis functions Berlinet and Thomas-Agnan (2011); Bishop (2006). For problems that optimize for the most smooth solution, i.e. the solution that minimizes the functional norm of the representation, it has been shown that the optimal function has a finite representation Kimeldorf and Wahba (1971); Schölkopf et al. (2001). However, no equivalent theorem has been proposed for problems that use different measures of complexity Peifer et al. (2020). The objective of this work is to find a representation of functions in RKHS, which can be used for any complexity measure, particularly that of sparsity. Furthermore, the potential applications that arise from obtaining such a sparse representation are of interest as well.

The second part of this work is concerned with the existence of the solution. Particularly, when encoding the fit requirements into the constraints, it is possible to formulate a problem that is either infeasible or admits representations that do not fit the data well. This is particularly of concern, as the training set grows, since the number of constraints grows as well. Therefore, it is imperative to have a good method of setting the constraint specifications.

This part of the work aims to provide a method for setting constraints by relaxing the ones that have the least marginal impact on the complexity measure.

The main contributions of this work are summarized here and are discussed in greater detail next:

1. Learning sparse models in reproducing kernel Hilbert spaces.
    (a) A continuous representation of functions in RKHS which is shown to be able to represent any function in that RKHS as well as functions of families of RKHS
    (b) A problem based on sparse functional problems (SFP) Chamon et al. (2018) which finds both the optimal reproducing kernel Hilbert space and the optimal kernel centers
    (c) A new integral representer theorem which holds for sparsity promoting complexity objectives
    (d) A method that has been shown empirically to produce representations of lower complexity while maintaining the same fit.
2. Federated Classification using Sparse Representations
    (a) A federated learning method which requires only one share across the network
    (b) Theoretical guarantees that the federated framework asymptotically achieves the same performance as a centralized framework
    (c) A method for federated learning which is both communication efficient and results in a low complexity representation
3. Resilient Learning for fit and complexity trade-off
    (a) A framework for individually constrained losses for learning in the presence of corrupted data

(b) A theory for constraint specification based on difficulty of meeting each constraint

(c) Resilient learning bounds for parametrization and empirical formulation of functional problem.

### 1.3.1 Learning Parsimonious Representations in Reproducing Kernel Hilbert Spaces

Reproducing kernel Hilbert spaces (RKHSs) are rich functional spaces, popular for their versatility and their parametric representations. Indeed, despite their ability to fit various patterns, functions in RKHSs can be represented by a possibly infinite linear combination of kernel functions Berlinet and Thomas-Agnan (2011); Bishop (2006). Moreover, a variational result, called the *representer theorem*, states that the optimal solution to the problem of minimizing the empirical loss using a class of smooth regularizers admits a finite representation Kimeldorf and Wahba (1971); Schölkopf et al. (2001). Furthermore, the representation requires only kernels centered at the sample points. Therefore, the *representer theorem* transforms an infinite functional problem into a parametrized finite problem. These methods however have some major drawbacks. Firstly, the RKHS has to be fixed beforehand and there is no straightforward way to pick the best functional space. While it is possible to search for the best space by trying different spaces, this can be computationally prohibitive as it requires the optimization problem to be solved repeatedly. Secondly, RKHSs have been shown to not be sample efficient for functions with heterogeneous degrees of smoothness Donoho and Johnstone (1998) and lastly, the complexity of the representation is directly linked to the sample size, which makes the evaluation of the learned functions computationally expensive as the sample size grows.

To address the challenges of traditional RKHS methods, an integral representation of a family of RKHS functions was introduced. This was done by leveraging the observation that some RKHS families have a kernel parameter that determines the smoothness of the function, e.g. bandwidth of sinc kernels, scale of Gaussian kernels. The integral representation replaces

the weight parameter from the linear combination of kernels by a function $\alpha$ over the space of kernel centers and kernel parameters, that define the family of functions. This is a more general representation which does not rely on the representer theorem to determine the kernel centers, and therefore can be used with any regularizer. Additionally, it can represent functions with varying degrees of smoothness efficiently, since it can use kernels from a family of RKHSs.

The integral representation was used to formulate a sparse functional problem, that finds parsimonious representations. This was done by adapting the discrete measure of sparsity to functions as the measure of support of the function. The optimization problem was then formulated using an elastic net type of objective function subject to the fit constraints. The optimal $\alpha$ becomes a superposition of bump functions. The integral representation can then be approximated by a discrete representation with only few kernels with kernel centers and parameters at the center of the bumps. In this manner, the optimization problem can be used to find the RKHS and kernel centers needed for a parsimonious representation.

A novel integral representer theorem arises from the solution of the optimization problem. Different from the traditional representer theorem, this theorem holds for the sparsity promoting regularizer and provides a method for selecting kernel centers and kernel parameters for a sparse representation.

Traditional methods for obtaining sparse representation use greedy heuristics Smola and Schölkopf (2000); Vincent and Bengio (2002) and $\ell_1$-norm relaxations of sparsity metrics Tibshirani (1996); Fung et al. (2002); Jud et al. (2016); Gao et al. (2013); Wright et al. (2008). These methods, however, rely on the representer theorem, although it no longer holds for sparsity promoting regularizers. Therefore, these methods no longer search for functions over the complete RKHS, but rather a smaller functional space which is spanned by the kernels centered at the sample points. The functional optimization problem searches over the entire RKHS and therefore can find more efficient representations than traditional methods if these exist. This is shown empirically in Section 3.5.

### 1.3.2 Federated Classification using Sparse Representations

Federated learning is a setting in which a central unit forms a global model from a federation of agents Konečnỳ et al. (2016a,b); Li et al. (2020); McMahan et al. (2016). This type of learning naturally arises in distributed networks of devices that each collect vast amounts of data, such as mobile phones, wearable devices, or autonomous vehicles Smith et al. (2017); Anguita et al. (2013). These systems often have limited communication capabilities Konečnỳ et al. (2015, 2016b); Smith et al. (2017); Zhao et al. (2018); Bonawitz et al. (2019). To account for these challenges and to preserve privacy, traditional federated learning methods take a distributed approach and share only the gradient over the network McMahan and Ramage (2017); Hard et al. (2018); McMahan et al. (2016). Although, these methods require less communication than traditional learning, they still require data to be transmitted at every iteration. Furthermore, as the communication load is reduced Wang et al. (2019); Yu et al. (2019); Shokri and Shmatikov (2015), there are no statistical guarantees of convergence.

In this work introduces a method for federated learning that requires a single data share over the network. This is done by only sharing a subset of the training data, that is fundamental to the classification problem. In order to find the fundamental samples, the agents solve the optimization problem of finding parsimonious representations in RKHS, each on their respective training set. The fundamental samples are then determined by the optimal dual maximizers. Optimal dual maximizers are known to be a measure of the difficulty of meeting a constraint, which makes them a great measure for determining the fundamental samples. Once the fundamental samples are sent, the central server can form the global model and send it back to the agents. Hence, a method for federated learning is established which requires a single share over the network.

The method for federated learning, presented in this work, is communication efficient. Moreover, the global model was proven to asymptotically converge to the traditionally trained model. Intuitively, as the sample size grows, the agents are more likely to agree on the fundamental samples, both within agents and with a traditional centralized unit that has access

to the entire data. Therefore, the representation of the federated solution asymptotically admits the same kernels as that of the centralized solution.

Each agent trains a local model in order to find the fundamental samples. By controlling the sparsity of the model, the agent can control the number of fundamental samples. More parsimonious representations require more detailed functions, $\alpha$ which leads to more fundamental samples. This motivates the agents to trade some level of sparsity for communication efficiency. The central unit, however, does not have the same limitations, it can optimize for sparsity regardless of the number of fundamental samples it finds. Therefore, the federated classification method is communication efficient and has global solution with a low complexity representation.

### 1.3.3 Resilient Learning for Balancing Fit and Complexity

The constrained optimization problem separates the complexity objective and the fit objective, which are differently scaled and has specifications that are interpretable. Indeed, the constraint specifications define the acceptable training loss for each sample. However, setting the specifications for problem (P-f) is not a straightforward task. If the values for $\epsilon_n$ are too large, then the optimal representation will not fit the data well. This motivates the choice of a small $\epsilon_n$, however, as the constraints are tightened the problem could become infeasible. In the case that the problem is feasible, it is solved in the dual domain, and is equivalent to the minimization of a regularized problem, for which the optimal dual variables are the regularizing parameters. Because the dual optimal values are a measure of the difficulty of meeting a constraint, the samples which are more difficult to fit are given more influence over the solution. This can lead to the model overfitting to outliers or corrupted samples.

In this work, a theory for constraint specification was established based on the difficulty of meeting a constraint. There is an inherent trad-off between the complexity of the representation and its ability to fit the data well. Solutions that have a lesser complexity are thought to generalize better, because they capture the underlying signal, but don't overfit to the training data. As the complexity of the representation decreases even further, however, the

signal representation is lost. The equilibrium is achieved when the marginal cost of relaxing a constraint is equal to the change in the measure of complexity. Moreover, although the change in the measure of complexity with respect to the relaxation would require the optimization problem to be solved multiple twice, the dual optimal variable provides a Fréchet subdifferentiable of the complexity measure.

The theory of constraint specification was established for the statistical learning problem. However, the statistical learning problem is often infinite dimensional and the true distribution is unknown. Section 4.3 establish a parametrized resilient learning problem and shows that the solution of the two problems are closed. In Section and 4.4 the issue of the unknown distribution is addressed by replacing the statistical loss with the empirical loss. The solution to the resilient empirical loss minimization exists and is close to that of the statistical loss under mild assumptions.

Relaxing constraints that are difficult to meet allows the model to ignore outliers and corrupted data points. At the same time, the model still is able to fit to most samples. Therefore, the model is resilient to corrupted data. This is illustrated in Section 4.6 through examples of label switching and image blurring.

# CHAPTER 2

# Learning Sparse Representations in Repropducing Kernel Hilbert Spaces

This chapter presents the work published in Peifer et al. (2020) which tackles the topic of finding sparse representations in reproducing kernel Hilbert spaces (RKHS). RKHS are non-parametric techniques used in signal processing, statistics and machine learning Schölkopf and Smola (2001); Bishop (2006); Hofmann et al. (2008); Yuan et al. (2010); Berlinet and Thomas-Agnan (2011); Arenas-Garcia et al. (2013); Koppel et al. (January 2019). For regularized empirical risk minimization problems, using a Tikhonov regularizer, there exists a variational result called the *representer theorem* which states that the optimal solution admits a finite kernel representation Kimeldorf and Wahba (1971); Schölkopf et al. (2001). Indeed, the optimal function can be expressed as a linear combination of kernel functions centered at the training sample points. Therefore, solving a regularized functional problem with a Tikhonov regularizer can be replaced by solving a parametrized finite problem.

As the sample size grows, however, so does the representation, which can make evaluating the function computationally prohibitive. Therefore, there is a need to find a lower complexity representation. A popular solution is imposing a sparsity penalty on the coefficients to reduce the number of kernel evaluations. Greedy heuristics Smola and Schölkopf (2000); Vincent and Bengio (2002) and $\ell_1$-norm relaxations Tibshirani (1996); Fung et al. (2002); Jud et al. (2016); Gao et al. (2013); Wright et al. (2008) are then used as a substitute to the combinatorial problem, which is known to be NP-hard in general Natarajan (1995); Amaldi and Kann (1998). These methods, however, often implicitly rely on the classical representer theorem Kimeldorf and Wahba (1971); Schölkopf et al. (2001), despite the fact that they no longer hold in the presence of sparsity penalties (see Remark 1). In this chapter, a method of obtaining a sparse representation by adapting kernels locally and allows kernels to be

centered arbitrarily, instead of restricting them to the sample points.

## 2.1 Learning in RKHS

In order to define a reproducing kernel Hilbert space, it is important to first define a kernel and an inner product.

**Definition 1.** Given a functional space $\mathcal{F}$, an inner product is defined as any function $\langle \cdot, \cdot \rangle : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$, which for any two functions $\phi, \psi \in \mathcal{F}$ and scalars $a_1, a_2 \in \mathbb{R}$ the following properties hold:

1. Symmetry: $\langle \phi, \psi \rangle = \langle \psi, \phi \rangle$
2. Linearity: $\langle a_1\phi_1 + a_2\phi_2, g \rangle = a_1\langle \phi_1, \psi \rangle + a_2\langle \phi_2, \psi \rangle$
3. Positive-definite: $\langle \phi, \phi \rangle \geq 0$ for all $\phi \in \mathcal{F}$ and $\langle \phi, \phi \rangle = 0$ if and only if $\phi = 0$.

**Definition 2.** Let $\mathcal{X}$ be a nonempty set. The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel if for for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ there exists a map $\phi : \mathcal{X} \to \mathcal{H}_0$ and an $\mathbb{R}$-Hilbert space $\mathcal{H}_0$ such that,

$$k(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle_{\mathcal{H}_0} \tag{2.1}$$

**Definition 3.** A Reproducing Kernel Hilbert Space is a complete, linear function space that is endowed with a unique kernel and an inner product.

RKHS in addition to being defined by a unique kernel also have the reproducing property

$$\langle \phi(\cdot), k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}_0} = f(\mathbf{x}) \tag{2.2}$$

for any function $\phi \in \mathcal{H}_0$ and point $\mathbf{x} \in \mathbb{R}^p$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ denotes the inner product of the Hilbert space $\mathcal{H}_0$ Berlinet and Thomas-Agnan (2011). From the reproducing property, it follows that functions in RKHS can be represented as the pointwise limit of a linear

combination of kernels, i.e.,

$$\phi(\mathbf{x}) = \lim_{n\to\infty} \sum_{j=1}^{n} a_j k(\mathbf{x}, \mathbf{z}_j \,;\, w_0), \tag{2.3}$$

where $w_0$ denotes the kernel parameter and the $\mathbf{z}_j \in \mathcal{X}$ are called the kernel *centers*. The kernel parameter $w_0$ characterizes the smoothness/richness of the RKHS, for example bandwidth of sincs, the order of polynomial kernels, or the scale/variance of Gaussian kernels (radial basis functions, RBF) Schölkopf and Smola (2001); Bishop (2006).

Given a training set of data pairs $(\mathbf{x}_n, y_n)$, $n = 1, \ldots, N$, where $\mathbf{x}_n \in \mathcal{X}$ are the observations or independent variables, with $\mathcal{X} \subset \mathbb{R}^p$ compact, and $y_n \in \mathbb{R}$ is the label or dependent variable, the goal is to find a function $\phi : \mathcal{X} \to \mathbb{R}$ in the RKHS $\mathcal{H}_0$ that fits these data, i.e., such that $c(\phi(\mathbf{x}_n), y_n)$ is small for some convex figure of merit $c$, e.g., quadratic loss, hinge loss, or logistic log-likelihood. Since there are infinitely many representations of the form (2.3) that can interpolate a finite set of points, a complexity reducing measure is added to the problem. A popular option which is also known to avoid overfitting is the RKHS norm of the solution Schölkopf and Smola (2001); Bishop (2006) as in

$$\begin{aligned} \underset{\phi\in\mathcal{H}_0}{\text{minimize}} \quad & \|\phi\|_{\mathcal{H}_0} \\ \text{subject to} \quad & c(\phi(\mathbf{x}_n), y_n) \le 0, \quad n = 1, \ldots, N. \end{aligned} \tag{PI}$$

Although the optimization problem above is infinite dimensional, its solution can be represented as a finite linear combination of kernels centered at the sample points, i.e., there exists a solution $\phi^\star$ of (PI) of the form Schölkopf et al. (2001); Kimeldorf and Wahba (1971)

$$\phi^\star(\cdot) = \sum_{n=1}^{N} a_n^\star k(\cdot, \mathbf{x}_n \,;\, w_0). \tag{2.4}$$

This result is called the representer theorem, which holds for any monotonically increasing real-valued function of the functional norm. The representer theorem reduces the functional

problem (PI) to the finite dimensional parametric problem

$$\begin{aligned} \underset{\{a_n\}\in\mathbb{R}}{\text{minimize}} \quad & \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m k(\mathbf{x}_n, \mathbf{x}_m\,;\, w_0) \\ \text{subject to} \quad & c(\hat{y}_n, y_n) \leq 0, \quad n = 1, \ldots, N, \\ & \hat{y}_n = \phi(\mathbf{x}_n) = \sum_{m=1}^{N} a_m k(\mathbf{x}_n, \mathbf{x}_m\,;\, w_0). \end{aligned} \tag{PI$'$}$$

Although finding a representation in an RKHS can be reduced to a finite parametric problem, there are still challenges that need to be addressed. The type of kernel and its parameter $w_0$ must be chosen *a priori*. The choice affects the types of functions that the model can represent. More than that, when trying to fit functions with heterogeneous degrees of smoothness, i.e. functions that are changing rapidly in some parts of the domain, and are smooth in others, these functions require a large sample size and have limited capabilities Donoho and Johnstone (1998). In this case, although the representer theorem guarantees a finite dimensional solution, the computational complexity of this solution will be very large such that the evaluation of the representation at a single point requires many kernel evaluations. To address this issue, methods that fit a combination of kernels from a predefined set Lanckriet et al. (2004); Micchelli and Pontil (2005); Gönen and Alpaydın (2011) or use spectral representations of positive-definite functions Ong et al. (2005); Wilson and Adams (2013); Yang et al. (2015) have been proposed. The choice of kernel parameter is then selected using grid search with cross-validation Bergstra and Bengio (2012); Kuhn and Johnson (2016) or application-specific heuristic such as maximizing the margin of the support vector machine (SVM) Li et al. (2012); Kuo et al. (2014). In order to better model functions with heterogeneous degrees of smoothness, methods have been developed that choose different RKHSs for different regions of the domain by means of plug-in rules Brockmann et al. (1993), binary optimization Liu et al. (1993), hypothesis testing Ghosh (2008), or gradient descent and alternating minimization Yuan et al. (2009); Chen et al. (2016), these solutions have no optimality guarantees due to the non-convexity of these locally adapted smoothness formulations. Due to the

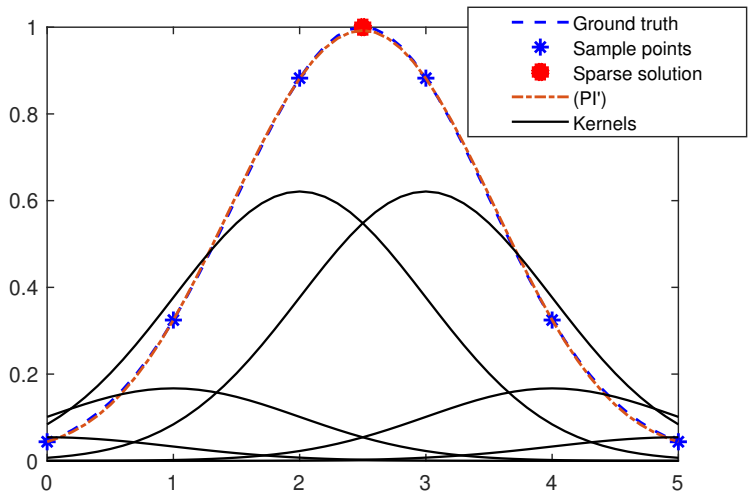



Figure 1: Illustration of Remark 1 on the importance of kernel centers for model complexity.

complexity of the solution when fitting functions with heterogeneous degrees of smoothness reducing the representation to a sparse set of coefficients $a_j$ in (PI′) is not tractable Natarajan (1995); Amaldi and Kann (1998). Moreover, the representer theorem no longer holds in the presence of a sparsity regularization. In fact, among the solutions which fit the data well, the most parsimonious solution does not always admit kernels centered at the sample points (see Remark 1).

The next section presents an integral representation of functions in RKHS that can be used to find parsimonious functions which locally adapt the kernel centers and kernel parameter to best fit the function. Moreover, it allows for regularization beyond smoothness, most notably sparsity.

**Remark 1.** *Among the representations in RKHS that fit the data the most parsimonious solution is not of the form* (2.4)*, therefore, classical represnter theorems do not apply. To see this is the case, consider the example illustrated in Figure 1. Let the sample points be taken from an underlying function composed of a single Gaussian kernel, namely*

$$y_n = \exp\left[-\frac{(x_i - 2.5)^2}{2}\right], \quad i = 1, \ldots, N, \tag{2.5}$$

*where $x_i \neq 2.5$ for all $i$. What is more, assume that the correct RKHS $\mathcal{H}_0$ is known, i.e., that the kernel function in* (2.3) *and* (2.4) *is $k(x,z) = \exp\left[-(x-z)^2/2\right]$. Then, it is clear that the most parsimonious function in $\mathcal{H}_0$ that fits the data is $\phi'(\cdot) = \exp\left[-\frac{(\cdot-2.5)^2}{2}\right]$. However, $\phi'$ is not in the feasible set of* (PI′). *Hence, though it can find functions with the same approximation error, they will not be the simplest representation, as illustrated in Figure 1. Peifer et al. (2020)*

## 2.2 RKHS: an Integral Representation

This section describes an integral representation and the optimization problem for learning that representation which addresses the three main challenges of classical RKHS-based methods: (i) the RKHS must be chosen *a priori*, (ii) they require a lot of training data for functions with heterogeneous degrees of smoothness, and (iii) the computational complexity of evaluating solutions grows with the sample size. In order to address the first two issues, the representation needs to admit kernels from multiple RKHSs; and in order to have a sparsity promoting regularizer, the representation needs to admit kernels at different centers. Let $\{\mathcal{H}_i \mid k(\cdot,\cdot;w_i) \in \mathcal{H}_i\}$ be a family of RKHSs, for which $w_i$ is the kernel parameter which can adapt the smoothness of the kernel. The representation of the function then becomes

$$\phi(\mathbf{x}) = \lim_{m\to\infty} \sum_{j=1}^{m} a_j k(\mathbf{x}, \mathbf{z}_j\,;\, w_j). \tag{2.6}$$

Although there have been similar formulations proposed to address the challenges of classical RKHS-based methods, optimally selecting $w_j$ and $\mathbf{z}_j$ remains an open problem Smola and Schölkopf (2000); Vincent and Bengio (2002); Tibshirani (1996); Fung et al. (2002); Jud et al. (2016); Gao et al. (2013). In order to optimally select the kernel parameters and centers an integral counterpart to (2.6) is introduced

$$f(\cdot) = \int_{\mathcal{X}\times\mathcal{W}} \alpha(\mathbf{z}, w) k(\cdot, \mathbf{z}\,;\, w) d\mathbf{z} dw, \tag{2.7}$$

where $\mathcal{W}$ is a compact subset of $\mathbb{R}$ and $\alpha : \mathcal{X}\times\mathcal{W} \to \mathbb{R}$ is in $L_2(\mathcal{X}\times\mathcal{W})$. Different from (2.6),

the representation in (2.7) does not explicitly require a set of kernel centers and parameters to be predetermined. Instead, it defines a continuous function over all possible centers and parameters. The representations in (2.6) and (2.7) both can represent all functions in $\mathcal{H}$as it is outlined in the following proposition

**Proposition 1.** *Let $k$ be a continuous reproducing kernel, i.e., $k(\cdot, \mathbf{z}; \cdot)$ is continuous over the compact set $\mathcal{X} \times \mathcal{W}$ for each $\mathbf{z} \in \mathcal{X}$. Then, for each $\phi \in \mathcal{H}$ there exists a sequence $\{f_m\}$ of functions as in* (2.7) *such that $f_m \to \phi$ pointwise.*

*Proof.* Consider the approximation of the identity $r_m(x) = m\,\mathbb{I}\,[|x| < 1/m]$ and note that $r_m(x) \to \delta(x)$ weakly in the vague topology, i.e., $\int_{\mathcal{D}} r_m(x)\varphi(x) \to \varphi(0)$ for all $\varphi$ continuous and $\mathcal{D}$ compact Rudin (1991). Now let $\phi \in \mathcal{H}$ be written as $\phi(\cdot) = \sum_{j=1}^{n} a_j k(\cdot, \mathbf{z}_j\,;\, w_j)$ and take $f_m(\cdot) = \int_{\mathcal{X}\times\mathcal{W}} \alpha_m(\mathbf{z}, w) k(\cdot, \mathbf{z}\,;\, w) d\mathbf{z} dw$ with

$$\alpha_m(\mathbf{z}, w) = \sum_{j=1}^{n} a_j r_m(w - w_j) \prod_{k=1}^{p} r_m([\mathbf{z}]_k - [\mathbf{z}_j]_k), \tag{2.8}$$

where $[\mathbf{z}]_k$ indicates the $k$-th element of the vector $\mathbf{z}$. Note that $\alpha_m \in L_2$, so that $f_m$ is indeed of the form (2.7). Since the reproducing kernel is continuous, it readily holds from (2.8) that $f_m \to \phi$ pointwise for all $\mathbf{x} \in \mathcal{X}$. ■

Proposition 1 shows that any function that admits a representation of the form (2.6) can be represented by a function of the form (2.7). This result is straightforward when the inner product of $\mathcal{H}$ is defined by (2.7). This is only the case if the kernel family is that of sinc functions.The result is also straightforward if $\alpha$ were a sum of Dirac delta functions, however, $\alpha$ is restricted to the space of $L_2$ functions.

### 2.2.1 A Sparse Functional Formulation

The integral representation (2.7) replaced a set of discrete coefficients with a function over the possible kernel centers and parameters. While $\alpha$ is continuous, by introducing a sparsity measure it can be reduced to a superposition of bump functions, i.e. a function that vanishes

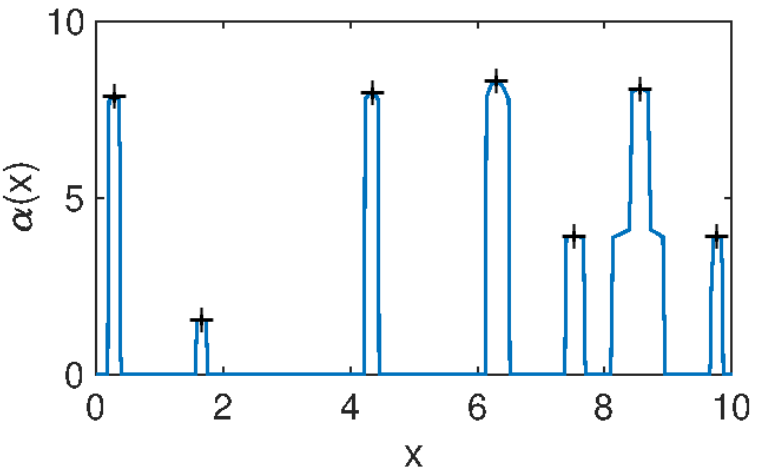



Figure 2: Sample $\alpha$ with bump centers.

over most of the domain except for a finite set of intervals. The integral representation can then be approximated by a discrete representation using only kernels centers and parameters at the peaks of the bump functions (Figure 2). The discrete approximation becomes closer to the integral representation as the sparsity is increased and the intervals over which the bump functions are defined get smaller. In the finite and countably infinite case, sparsity is defined as the number of non-zero coefficients Zhang et al. (2015); Tropp (2006); Rudelson and Vershynin (2008). In the continuous case, sparsity was defined as the Lebesgue measure of the support of the function. A counterpart to the discrete "$\ell_0$-norm" is established as the "$L_0$-norm"

$$\|\alpha\|_{L_0} = \int_{\mathcal{X}\times\mathcal{W}} \mathbb{I}\left[\alpha(\mathbf{z}, w) \neq 0\right] d\mathbf{z}dw, \tag{2.9}$$

where $\mathbb{I}[x \neq 0] = 1$ if $x \neq 0$ and zero otherwise. Using this measure, sparse functions are considered to be functions of limited support as constructed by the proof of Proposition 1 and illustrated in Figure 2. This observation motivates estimating $f$, equivalently $\alpha$, using

the following SFP:

$$\begin{aligned} \underset{\alpha \in L_2(\mathcal{X}\times\mathcal{W})}{\text{minimize}} \quad & \frac{1}{2}\|\alpha\|_{L_2}^2 + \gamma\|\alpha\|_{L_0} \\ \text{subject to} \quad & c(\hat{y}_n, y_n) \leq 0, \quad n = 1, \ldots, N, \\ & \hat{y}_n = f(\mathbf{x}_n) = \int_{\mathcal{X}\times\mathcal{W}} \alpha(\mathbf{z}, w) k(\mathbf{x}_n, \mathbf{z}\,;\, w) d\mathbf{z} dw, \end{aligned} \tag{PII}$$

where $\gamma \geq 0$ is a regularization factor that trades-off smoothness and sparsity; $\|\cdot\|_{L_2}$ denotes the $L_2$-norm, which induces smoothness, enhances robustness to noise, and improves the numerical properties of the optimization problem. It is worth noting that the solution which minimizes the unconstrained objective function is $\alpha = 0$ or any $\alpha \in L_2$ with support on any countable set. This trivial solution, while optimizing the complexity, does not meet the fit constraints and is not a feasible solution to the problem.

The problem presented in (PII) locally adapts the kernel parameter based in order to fit functions of varying degrees of smoothness. Additionally, because it allows for multiple RKHSs, it does not need to preselect the kernel. Moreover, it adapts the kernel centers, in order to obtain parsimonious representations that still fit the data well. Discrete representations can be approximated from the sparse continuous representations by using the bumps in the solution $\alpha$ to determine pairs $(\mathbf{z}_j, w_j)$.

Although the remainder of this chapter studies the general problem (PII), there are two special cases of the problem that are of interest. In the first case, the functional space is application specific and is known from expert knowledge. In this case, the problem only searches over kernel centers while keeping the reproducing kernel $k$ and its parameter $w_0$ fixed. The problem (PII) reduces to

$$\begin{aligned} \underset{\alpha \in L_2(\mathcal{X})}{\text{minimize}} \quad & \frac{1}{2}\|\alpha\|_{L_2}^2 + \gamma\|\alpha\|_{L_0} \\ \text{subject to} \quad & c(\hat{y}_n, y_n) \leq 0, \quad n = 1, \ldots, N, \\ & \hat{y}_n = f(\mathbf{x}_n) = \int_{\mathcal{X}} \alpha(\mathbf{z}) k(\mathbf{x}_n, \mathbf{z}\,;\, w_0) d\mathbf{z}. \end{aligned} \tag{PII$'$}$$

Problem (PII′) optimizes for sparsity for a fixed RKHS. This type of problem has been attempted using greedy methods such as KOMP Vincent and Bengio (2002). These methods rely on the representer theorem theorem Koppel et al. (January 2019) to get a dense solution before reducing the number of kernels in the representation. However, this sparse representation is not guaranteed to be optimal (See Remark 1). In contrast, the solution of (PII′) is guaranteed to provide the sparsest integral representation because it searches over all possible representations and not just the subset of representations which have kernels centered at the sample points. A discrete solution can then be approximated by a peak finding method. Problem (PII′) can therefore obtain solutions with similar performance to greedy methods but lower complexity, as illustrated in Section 2.4.

The second special case of interest occurs when a set of candidate kernels is available, either obtained from domain experts or unsupervised techniques such as clustering. The problem (PII) can be reduced to only search over the kernel parameters and choose a subset of kernel centers

$$\begin{aligned} \underset{\alpha_j \in L_2(\mathcal{W})}{\text{minimize}} \quad & \sum_{j=1}^{M} \left[ \frac{1}{2} \|\alpha_j\|_{L_2}^2 + \gamma \|\alpha_j\|_{L_0} \right] \\ \text{subject to} \quad & c(\hat{y}_n, y_n) \leq 0, \quad n = 1, \ldots, N, \\ & \hat{y}_n = f(\mathbf{x}_n) = \sum_{j=1}^{M} \int_{\mathcal{W}} \alpha_j(w) k(\mathbf{x}_n, \mathbf{z}_j \,;\, w) dw, \end{aligned} \tag{PII″}$$

where $\mathbf{z}_j$, for $j = 1, \ldots, M$, are the predefined candidate centers. Problem (PII″) optimizes the sparsity of each $\alpha_j$, such that the kernel centers that do not have an effect on the fit of the representation will vanish. Hence, the solution of (PII″) effectively selects the smallest subset of candidate centers. Moreover, by locally adapting the kernel parameter, the number of candidate kernels can be further reduced by placing kernels that more naturally fit the data. The problem (PII″) is advantageous for high dimensional problems for which the computation of the integral might become intractable.

Problem (PII) balances the fit and complexity objectives by optimizing over the sparsest integral representation that fits the data. While the problem is nonconvex and infinite

dimensional, it will be shown in the next section that a solution can be found exactly and efficiently

## 2.3 Learning in the Dual Domain

Problem (PII) [or (PII′)–(PII″)] is a non-convex, infinite dimensional optimization program. However, it can be efficiently solved by using duality, which is an established approach to solve semi-infinite convex programs Shapiro (2006); Tang et al. (2013); Candès and Fernandez-Granda (2014). Dual problems are concave even when the primal function is not convex and are finite dimensional. Indeed, the dimension of the dual problem is equal to the number of constraints. In addition to being solvable, the solution to the primal problem can be obtained from that of the dual problem when the primal is convex and mild conditions hold Boyd and Vandenberghe (2004). Although, (PII) is nonconvex its solution can be obtained from solving the dual problem.

### 2.3.1 The Dual Problem of (PII)

In order to derive the dual problem of (PII), the Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^N$, associated with its equality constraints and $\boldsymbol{\mu} \in \mathbb{R}^N_+$, associated with its inequality constraints, are introduced. The Lagrangian is then defined as

$$\begin{aligned}\mathcal{L}(\alpha, \hat{\boldsymbol{y}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\|\alpha\|^2_{L_2} + \gamma\|\alpha\|_{L_0} - \sum_{n=1}^{N} \boldsymbol{\lambda}_n \int \alpha(\mathbf{z}, w) k(\mathbf{x}_n, \mathbf{z}\,;\, w) d\mathbf{z} dw \\ + \sum_{n=1}^{N} \boldsymbol{\lambda}_n \hat{y}_n + \sum_{n=1}^{N} \boldsymbol{\mu}_n c(\hat{y}_n, y_n).\end{aligned} \tag{2.10}$$

The set $\mathcal{X} \times \mathcal{W}$ over which the integrals are computed was omitted for a clearer notation. The dual function is obtained by minimizing the Lagrangian over the primal variables

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\substack{\alpha \in L_2 \\ \hat{y}_n \in \mathbb{R}}} \mathcal{L}(\alpha, \hat{\boldsymbol{y}}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{2.11}$$

The dual problem maximizes the dual function over the feasible dual variables

$$\underset{\boldsymbol{\lambda}\in\mathbb{R}^N,\,\boldsymbol{\mu}\in\mathbb{R}^N_+}{\text{maximize}} \quad g(\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{DII}$$

The dual function is defined as the minimum over a set of affine functions in $(\boldsymbol{\lambda},\boldsymbol{\mu})$ and is therefore always concave Boyd and Vandenberghe (2004), regardless of the convexity of the primal function. Furthermore, the dimensionality of the dual problem is always equal to the number of constraints. Therefore, the dual problem is a concave finite dimensional problem which can be solved efficiently if it can be computed. However, computing the dual function $g$ requires solving a functional, non-convex problem. Nonetheless, (2.11) can be separated as

$$g(\boldsymbol{\lambda},\boldsymbol{\mu}) = \min_{\hat{y}_n\in\mathbb{R}} \mathcal{L}_{\hat{y}}(\hat{\boldsymbol{y}},\boldsymbol{\lambda},\boldsymbol{\mu}) + \min_{\alpha\in L_2} \mathcal{L}_{\alpha}(\alpha,\boldsymbol{\lambda}), \tag{2.12}$$

where

$$\mathcal{L}_{\hat{y}}(\hat{\boldsymbol{y}},\boldsymbol{\lambda},\boldsymbol{\mu}) = \sum_{n=1}^{N} \boldsymbol{\mu}_n c(\hat{y}_n, y_n) + \sum_{n=1}^{N} \boldsymbol{\lambda}_n \hat{y}_n \tag{2.13}$$

is a convex function, since $c$ is convex and $\boldsymbol{\mu}_n \geq 0$, and

$$\mathcal{L}_{\alpha}(\alpha,\boldsymbol{\lambda}) \;=\; \int \left[\frac{1}{2}\alpha^2(\mathbf{z},w) + \gamma\,\mathbb{I}\left[\alpha(\mathbf{z},w)\neq 0\right] \;-\; \sum_{n=1}^{N} \boldsymbol{\lambda}_n \alpha(\mathbf{z},w) k(\mathbf{x}_n,\mathbf{z}\,;\,w)\right] d\mathbf{z}dw. \tag{2.14}$$

From (2.14) it follows that computing $g$ requires solving a non-convex functional optimization problem similar to the original (PII). Different from (PII), this is an unconstrained minimization problem, and it can be solved by exploiting separability to obtain a closed form thresholding solution.

**Proposition 2.** *A minimizer $\alpha_d$ of* (2.14) *is given by*

$$\alpha_d(\mathbf{z},w;\boldsymbol{\lambda}) = \begin{cases} \bar{\alpha}_d(\mathbf{z},w;\boldsymbol{\lambda}), & |\bar{\alpha}_d(\mathbf{z},w;\boldsymbol{\lambda})| > \sqrt{2\gamma} \\ 0, & \textit{otherwise} \end{cases} \tag{2.15}$$

*where* $\bar{\alpha}_d(\mathbf{z},w;\boldsymbol{\lambda}) = \sum_{n=1}^{N} \boldsymbol{\lambda}_n k(\mathbf{x}_n,\mathbf{z};w)$.

*Proof.* To obtain (2.15), the objective of (2.14) is separated across $\mathbf{z}$ and $w$, by leveraging the following lemma:

**Lemma 1.** *Let $F(\alpha, x)$ be a normal integrand as defined in (Rockafellar and Wets, 2009, Def. 14.27). Then,*

$$\inf_{\alpha \in L_2} \int F(\alpha(x), x) dx = \int \inf_{\bar{\alpha} \in \mathbb{R}} F(\bar{\alpha}, x) dx. \tag{2.16}$$

*Proof.* See (Rockafellar and Wets, 2009, Thm. 14.60) or (Shapiro et al., 2014, Thm. 7.92). Note that since $(\mathcal{X} \times \mathcal{W})$ is Borel and $F$ is lower semicontinous in $\alpha$ (its only point of discontinuity is $\alpha = 0$) , $F$ is normal (Rockafellar and Wets, 2009, Ex 14.31). ■

Defining the normal integrand as

$$F(\bar{\alpha}, \mathbf{z}, w) = \frac{\bar{\alpha}^2}{2} + \gamma \, \mathbb{I}\left[\bar{\alpha} \neq 0\right] - \sum_{n=1}^{N} \boldsymbol{\lambda}_n k(\mathbf{x}_n, \mathbf{z}\,;\, w)\bar{\alpha}, \tag{2.17}$$

in Lemma 1, yields that minimizing (2.14) is equivalent to minimizing $F$ individually for each $(\mathbf{z}, w)$. The indicator function in (2.17) can only take two values depending on $\bar{\alpha}$, therefore, its optimal value is the minimum of two cases: (i) if $\bar{\alpha} = 0$, then $F(0, \mathbf{z}, w) = 0$ for all $(\mathbf{z}, w)$; alternatively, (ii) if $\bar{\alpha} \neq 0$, then (2.17) becomes

$$F'(\bar{\alpha}, \mathbf{z}, w) = \frac{\bar{\alpha}^2}{2} - \sum_{n=1}^{N} \boldsymbol{\lambda}_n k(\mathbf{x}_n, \mathbf{z}\,;\, w)\bar{\alpha} + \gamma, \tag{2.18}$$

whose minimization is a quadratic problem with closed-form solution

$$\bar{\alpha}^\star(\mathbf{z}, w) = \operatorname*{argmin}_{\bar{\alpha} \in \mathbb{R}} F'(\bar{\alpha}, \mathbf{z}, w) = \sum_{n=1}^{N} \boldsymbol{\lambda}_n k(\mathbf{x}_n, \mathbf{z}\,;\, w), \tag{2.19}$$

so that $\min_{\bar{\alpha} \neq 0} F(\bar{\alpha}^\star, \mathbf{z}, w) = \gamma - \bar{\alpha}^\star(\mathbf{z}, w)^2/2$. Immediately, $\alpha_d(\mathbf{z}, w) = \bar{\alpha}^\star(\mathbf{z}, w)$ if $\gamma - \bar{\alpha}_d(\mathbf{z}, w)^2/2 < 0$ or $\alpha(\mathbf{z}, w)$ vanishes, which yields (2.15). ■

Proposition 2 shows that the solution to the non-convex infinite dimensional problem (2.14) is

simply the solution of a quadratic problem that is thresholded by the regularizing parameter $\gamma$. This leads to a closed form solution for (DII). As a result, classical convex optimization methods can be used to solve the dual problem. In Section 2.3.3 (stochastic) (super)gradient ascent is used to find the optimal solution. The optimal dual is known to be a lower bound to the primal problem Boyd and Vandenberghe (2004). In the next section, it is shown that solving the dual leads to the solution of the primal problem because of strong duality.

### 2.3.2 Strong Duality and the Integral Representer Theorem

In general, for non-convex problems, the optimal value of the dual problem provides a lower bound for the primal optimal value, however, in some cases strong duality holds for non-convex problems and the optimal dual value and optimal primal value are equal Chamon et al. (2018). The central technical result in this section shows that (PII) has null duality gap (Theorem 1). This result implies that the solution of (PII) is readily obtained from the solution of the dual problem (DII) (Corollary 1). Moreover, it leads to a new representer theorem, which holds for the sparsity promoting regularizer.This integral representer theorem shows that the optimal primal value has a finite dimensional representation. The following theorem describes the conditions for which strong duality holds:

**Theorem 1.** *Strong duality holds for* (PII) *if the kernel* $k(\cdot, \mathbf{z}\,;\, \omega)$ *has no point masses and Slater's condition is met. In other words, if* $P^*$ *is the optimal solution of* (PII) *and* $D^*$ *is the optimal solution of* (DII)*, then* $P - D$*.*

*Proof.* The proof is provided in Appendix A.1 and is analogous to the result presented in Chamon et al. (2018). ■

Theorem 1 states that the optimal value of the dual problem (4.9) is equal to that of the primal problem (PII). The way to obtain the optimal primal variables is described in the following corollary:

**Corollary 1.** *Let* $(\boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star)$ *be a solution of* (DII) *and assume* $k \in L_2$ *and analytic. Then,* $\alpha_d^\star(\cdot, \cdot) = \alpha_d(\cdot, \cdot; \boldsymbol{\lambda}^\star)$ *is a solution of* (PII) *for* $\alpha_d$ *as in* (2.15)*.*

*Proof.* See Appendix A.2. ■

Theorem 1 and corollary 1 provide a path for solving problem (PII) despite it being non-convex and infinite dimensional by leveraging duality. The hypothesis made for Corollary 1 are mild and hold for most reproducing kernels including Gaussian and sinc kernels. More importantly, the solution does not rely on formulating a discrete representation of the function, since discrete problems are often NP-hard, large dimensional, and potentially ill-conditioned problems. Furthermore, it offers a solution to the problem of finding sparse representations without relying on convex relaxations of sparsity promoting regularizers or representations that are sub-optimal.

Another fundamental implication of Theorem 1 is the following integral representer theorem.

**Corollary 2** (Integral Representer Theorem)**.** *A solution $\alpha^\star$ of* (PII) *can be obtained by thresholding a parametrized family of functions $\bar{\alpha}_w^\star \in \mathcal{H}_w$, where $\mathcal{H}_w$ is the RKHS induced by the kernel $k(\cdot,\cdot;w)$. In fact, $\bar{\alpha}_w^\star$ lives in a finite dimensional subspace of $\mathcal{H}_w$ spanned by the kernels evaluated at the data points. Explicitly, there exist $a_n \in \mathbb{R}$ such that*

$$\bar{\alpha}_w^\star(\cdot) = \sum_{n=1}^{N} a_n k(\mathbf{x}_n, \cdot; w). \tag{2.20}$$

*Proof.* From Corollary 1, $\alpha^\star = \alpha_d^\star$ almost everywhere with $\alpha_d^\star(\mathbf{z}, w) = \alpha_d(\mathbf{z}, w; \boldsymbol{\lambda}_n^\star)$. Thus, the corollary stems from (2.15) for $\alpha_w(\cdot) = \sum_{n=1}^{N} \boldsymbol{\lambda}_n^\star k(\mathbf{x}_n, \cdot; w)$, so that $\alpha_w \in \mathcal{H}_w$ by definition and $\alpha^\star(\mathbf{z}, w) = \alpha_w(\mathbf{z})\, \mathbb{I}(|\alpha_w(\mathbf{z})| > \sqrt{2\gamma})$. ■

The classical representer theorem Kimeldorf and Wahba (1971); Schölkopf et al. (2001) shows that the optimal representation admits only kernels centered at the sample points and therefore reduces the functional optimization problem (PI) to the finite dimensional problem (PI$'$). Similarly, The integral representer theorems offers a finite closed form solution for the optimal variable $\alpha$ (2.20). Although classical representer theorem only provides a finite representation for regularizers that are monotonically non-decreasing functions of the functional norm

which does not include any sparsity promoting functions, Corollary 2 holds in the presence of the "$L_0$-norm" regularizer.

From Corollary 2 it would appear that the computational complexity has not been reduced since the evaluation of a single the function $\alpha$ at a single point $(\mathbf{z}, w)$ requires a kernel function to be computed for each training point. Indeed, the complexity of evaluating $\alpha$ is $O(N)$. However, $\alpha$ can be approximated by a discrete vector of length $K$, by only allowing one kernel per bump. Due to the limited support of $\alpha$, the number of kernels $K \ll N$ and is determined by the underlying signal and can be controlled by the choice of $\gamma$. The discrete approximation provides a solution of computational complexity $O(K) \ll O(N)$.

An important consequence of Corollary 2 is that although problem (PII) allows for any function $\alpha \in L_2$, the solution is itself a function that belongs to the family of kernels to which $f$ is restricted over limited support. Explicitly, the solution $\alpha^\star(\cdot, w)$ of (PII) is a thresholded version of a function in the RKHS $\mathcal{H}_w$. In the spacial case of $\gamma = 0$, i.e. no sparsity, we have that $\alpha^\star(\cdot, w) \in \mathcal{H}_w$. Moreover, for the problem (PII$'$), this further simplifies to $\alpha^\star \in \mathcal{H}_0$.

Equation (2.7) can therefore be interpreted as building the function $f^\star$ point-by-point by integrating the value of partial $L_2$-inner products between the reproducing kernel of $\mathcal{H}_w$ and a function $\bar{\alpha}_w \in \mathcal{H}_w$. Explicitly, (2.7) can be written as

$$f^\star(\cdot) = \int_{\overline{\mathcal{W}}} \left[ \int_{\overline{\mathcal{X}}} \bar{\alpha}_w(\mathbf{z}) k(\cdot, \mathbf{z}; w) d\mathbf{z} \right] dw, \tag{2.21}$$

where $\overline{\mathcal{X}} \subseteq \mathcal{X}$, $\overline{\mathcal{W}} \subseteq \mathcal{W}$, and $\overline{\mathcal{X}} \times \overline{\mathcal{W}}$ is the set induced by the support of $\alpha^\star$, i.e., $\{(\mathbf{z}, w) \in \mathcal{X} \times \mathcal{W} \mid |\alpha(\mathbf{z}, w)| > \sqrt{2\gamma}\}$. The innermost integral in (2.21) can be interpreted as an inner product in $L_2$ between $\bar{\alpha}_w$ and $k(\cdot, \mathbf{z}; w)$ computed only where the magnitude of $\bar{\alpha}_w$ is large enough, defined by the regularization parameter $\gamma$. This sort of trimmed inner product is linked to robust projections found in different statistical methods Chen et al. (2013); Feng et al. (2014). The outer integral then accumulates the projections of $\bar{\alpha}_w$ over the relevant

**Algorithm 1** Stochastic optimization Peifer et al. (2020) for (PII)

1: Initialize $\boldsymbol{\lambda}_n(0)$ and $\boldsymbol{\mu}_n(0) > 0$
2: **for** $t = 0, 1, \ldots, T$
3: Evaluate the supergradient $d_{\boldsymbol{\mu}_n}(t) = c\left(\hat{y}_{d,i}(t), y_n\right)$ for

$$\hat{y}_{d,n}(t) = \underset{\hat{y}_n}{\operatorname{argmin}} \sum_{i=1}^{N} \boldsymbol{\mu}_n(t) c(\hat{y}_n, y_n) - \sum_{n=1}^{N} \boldsymbol{\lambda}_n(t) \hat{y}_n$$

4: Draw $\{(\mathbf{z}_k, w_k)\}$, $k = 1, \ldots, B$, uniformly at random and compute the stochastic supergradient

$$\hat{d}_{\boldsymbol{\lambda}_n}(t) = \hat{y}_{d,i}(t) - \frac{1}{B} \sum_{k=1}^{B} \alpha_d(\mathbf{z}_k, w_k; \boldsymbol{\lambda}_n(t)) k(\mathbf{x}_n, \mathbf{z}_k; w_k)$$

5: Update the dual variables:

$$\begin{aligned} \boldsymbol{\lambda}_n(t+1) &= \boldsymbol{\lambda}_n(t) + \eta_\lambda \hat{d}_{\boldsymbol{\lambda}_n}(t) \\ \boldsymbol{\mu}_n(t+1) &= \left[\boldsymbol{\mu}_n(t) + \eta_\mu d_{\boldsymbol{\mu}_n}(t)\right]_+ \end{aligned}$$

6: **end**
7: Evaluate the primal solution as

$$\alpha^\star(\mathbf{z}, w) = \begin{cases} \bar{\alpha}^\star(\mathbf{z}, w), & |\bar{\alpha}^\star(\mathbf{z}, w)| > \sqrt{2\gamma} \\ 0, & \text{otherwise} \end{cases}$$

for $\bar{\alpha}^\star(\mathbf{z}, w) = \sum_{i=1}^{N} \boldsymbol{\lambda}_n(T) k(\mathbf{x}_n, \mathbf{z}; w)$

subset $\overline{\mathcal{W}}$ of RKHSs considered to form the functional solution.

Theorem 1 holds under very mild conditions as it only requires the kernels to not have point masses. This is the case for most reproducing kernels used in applications, such as polynomial or Gaussian kernels. In fact, by restricting $\alpha$ to $L_2$, functions with point masses are automatically not considered since these are not square integrable. Moreover, due to the infinite dimensionality of $\alpha$ it is always possible to find a representation that fits the data. Therefore, finding a strictly feasible solution which meets Slater's condition Boyd and Vandenberghe (2004) is possible for most cost functions $c$.

### 2.3.3 Dual Gradient Ascent

Corollary 1 shows that the optimal primal variables of (PII) are precisely the dual minimizer from Proposition 2. These have a closed form solution (2.15) that depends on the optimal dual

variables $(\boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star)$. In order to find the optimal dual variable a projected supergradient ascent method is proposed (Algorithm 1). This method was used to obtain the results presented in Section 2.4, however, any convex optimization algorithms can be used, for example exploiting structure in the problem to solve large-scale instances Bertsekas (2015).

The supergradient of a function $f : \mathcal{D} \to \mathbb{R}$ at $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^n$ is any vector $\mathbf{d}$ such that $f(\mathbf{y}) \leq f(\mathbf{x}) + \mathbf{d}^T(\mathbf{y} - \mathbf{x})$ for all $\mathbf{y} \in \mathcal{D}$. Although supergradients are not guaranteed to be the steepest ascent direction at $\mathbf{x}$, taking small steps in their direction decreases the distance to any maximizer of a convex function $f$ Boyd and Vandenberghe (2004). Moreover, when the function is differentiable at $\mathbf{x}$, the supergradient is unique and represents the gradient. Thus, supergradients can be used to optimize concave functions, which are not guaranteed to be differentiable everywhere. The dual problem (DII) is solved by repeating, for $t = 0, 1, \ldots,$

$$\boldsymbol{\lambda}_n(t+1) = \boldsymbol{\lambda}_n(t) + \eta_\lambda d_{\boldsymbol{\lambda}_n}(\boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)), \tag{2.22a}$$

$$\boldsymbol{\mu}_n(t+1) = \left[\boldsymbol{\mu}_n(t) + \eta_\mu d_{\boldsymbol{\mu}_n}(\boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\right]_+, \tag{2.22b}$$

where $\mathbf{d}_{\boldsymbol{\lambda}}, \mathbf{d}_{\boldsymbol{\mu}}$ are the supergradients of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, respectively, $\eta_\lambda, \eta_\mu > 0$ are step sizes, and $[x]_+ = \max(0, x)$. The dual variables $\boldsymbol{\mu}_n$ are projected onto the non-negative numbers in order to guarantee that the optimal $\boldsymbol{\mu}_n$ is feasible. The supergradient in (2.22) are readily obtained from the constraint violation of the dual minimizers Boyd and Vandenberghe (2004). Given $\hat{y}_{d,n}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and $\alpha_d(\mathbf{z}, w; \boldsymbol{\lambda})$, the minimizers of (2.13) and (2.14) respectively, the $n^{th}$ element of the supergradient vectors is expressed as

$$d_{\boldsymbol{\lambda}_n}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \hat{y}_{d,n}(\boldsymbol{\lambda}, \boldsymbol{\mu}) - \int \alpha_d(\mathbf{z}, w; \boldsymbol{\lambda}) k(\mathbf{x}_n, \mathbf{z}; w) d\mathbf{z} dw, \tag{2.23a}$$

$$d_{\boldsymbol{\mu}_n}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = c\left(\hat{y}_{d,n}(\boldsymbol{\lambda}, \boldsymbol{\mu}), y_n\right). \tag{2.23b}$$

Since $\hat{y}_{d,n}$ is the solution of the convex optimization problem (2.13), the update for the dual

variables $\boldsymbol{\mu}_n$ in (2.22b) can be efficiently evaluated using (2.23b). The update expression for $\boldsymbol{\lambda}_n$ in (2.22a), however, requires that the integral in (2.23a) be evaluated.This can be done by either using numerical integration methods, since $\alpha_d$ has a closed-form representation from (2.15), or using Monte Carlo methods to approximate the integral. The latter approach was integrated with the optimization iterations in (2.22) to obtain a stochastic supergradient ascent algorithm summarized in Algorithm 1. The Monte Carlo approach gives an unbiased estimate of $d_{\boldsymbol{\lambda}_n}$, and has been shown convergence in Ruszczyński and Syski (1986); Ribeiro (2010); Bottou et al. (2016).

## 2.4 Applications

The following results have been presented in Peifer et al. (2020) ©[2020]IEEE. In the previous sections we have claimed that our algorithm can estimate (i) kernel widths (PII$''$), (ii) kernel centers (PII$'$), and (iii) kernels of varying centers and widths (PII). In this section we show, through a sample signal, how we can achieve claim (i). Then we show how moving from (i) to (iii) reduces complexity. In our discussion about the complexity of the representation in section 2.4.1, we show how we can achieve (ii) on random signals of fixed width. In section 2.4.2, we solve (PII) for a signal of varying degrees of smoothness and show how we can reduce complexity regardless of sample size. Lastly, in sections 2.4.3 and 2.4.4 we apply our algorithm to solve (PII) and (PII$''$) on two examples of real applications: a user localization problem and a digit classification problem.

For the estimation, we search over functions in the family of RKHSs, which have Gaussian functions as kernels

$$k(\mathbf{x}, \mathbf{x}') = \exp\left\{\frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{2w^2}\right\}, \tag{2.24}$$

where the width of the kernel is directly proportional to the hyper-parameter $w$. We select a quadratic cost function

$$c(\mathbf{z}_i, \mathbf{y}_i) = (\mathbf{y}_i - \mathbf{z}_i)^2 - \epsilon, \tag{2.25}$$

where $\epsilon$ represents the maximum squared error of the solution on the training data. Notice that using a quadratic cost function leads to the following solution for $\hat{y}_{d,i}$

$$\hat{y}_{d,i} = y_i - \frac{\lambda_i}{2\mu_i} \tag{2.26}$$

To start, the effect of the choice of RKHS on the performance of a learning algorithm is examined. To this end, a signal, which lies in the RKHS with a Gaussian kernel of width $w_0 = 0.453$ is constructed. The classical problem in (PI$'$) is compared to the problem presented in (PII$''$). A grid search is used to examine the performance of (PI$'$) for different values of $w$. Recall that (PI$'$) is the finite dimensional problem equivalent to (PI) obtained by leveraging the representer theorem. Therefore, the representation of the solution admits kernels centered at the sample points.

The value of $w_0$ was chosen such that it would not be directly on the grid, since in practice it is unlikely to include the value of the width of the originating signal. We generate $S$ signals of the form

$$f_j(x) = \sum_{i=1}^{m} a_i \times \exp\left[-\frac{\|\mathbf{x} - \tilde{\mathbf{x}}_i\|^2}{2 * w_0^2}\right] + \xi_j \tag{2.27}$$

with $j = 1 \dots S$. For each $f_j$ a training set of $N = 50$ samples was generated with $m = 10$. The amplitude $a_i$ of each function is selected at random from a uniform distribution $\mathcal{U}(1, 2)$. The $\tilde{\mathbf{x}}_i$ are i.i.d random variables drawn from the uniform distribution $\mathcal{U}(1, 2)$ and the $\xi_j$ are i.i.d. random variables drawn from $\mathcal{N}(0, 10^{-3})$, which represent the noise.

It should be noted that, given sufficient iterations, well chosen step sizes, and a large $\gamma$, our method can approximate point masses. However, smoother approximations of the point masses can be obtained by using only few iterations. Additionally, these smooth approximations are more robust to the choice of the tuning parameters. Kernel centers and widths can subsequently be obtained by selecting the extreme points of the function $\alpha(\mathbf{z}, w)$, since the optimal $\alpha(\mathbf{z}, w)$ is a function of $w$, the kernel width, and $\mathbf{z}$ the kernel centers. Kernels

using the widths and centers approximated from the extreme points of the $\alpha(\mathbf{z}, w)$ are used to train a least squared estimator.

A grid search is performed for problem (PI$'$) by uniformly sampling $w$ over the interval [0,1] at 0.1 increments. The problem (PII$''$) is solved using $\gamma = 4000$, $\eta_\lambda = 0.001$, $\eta_\mu = 0.1$ and $T = 5000$. The performance of the two algorithms is compared over 1000 realizations of the sampled signal, each with a training set of size $N = 100$ and a test set of size $N_{test} = 1000$. The MSE of (PI$'$) decreases as the value of $w$ increases—see Figure 3. Due to the non-uniform sampling of the signal, smoother kernels on average have a better performance. In areas, in which the sampling is sparse, the thinner kernels cannot represent the signal between the samples. Additionally, the thinner kernels are more likely to overfit to the noise than the smoother kernels. However, the smoother kernels cannot model the faster variation in the signal well. In contrast, (PII$''$) finds a sparse solution, which uses 14 kernels on average, of varying smoothness, with an average MSE of 0.0457, which can both take advantage of the ability of smoother kernels to avoid overfitting and thinner kernels to model fast variation. Indeed, we observe in Figure 4 that our algorithm chooses a mixture of kernels of width around 0.453 and kernels of width 1.

Smoother kernels perform better because of the random sampling combined with the restriction of only using kernels centered at the sample points. Therefore, we investigate the effect of solving problem (PII) which finds both kernel centers and kernel widths. Problem (PII) is solved using $\gamma = 1000$, $\eta_\lambda = 0.01$, $\eta_\mu = 1$ and $T = 1000$ over 1000 randomly sampled training sets, and results in an MSE of 0.0588. Although the MSE of (PII) is similar to that of (PII$''$), it is important to note that by placing kernels arbitrarily we are able to better estimate the width of the kernel: by comparing Figure 5 to Figure 4 it can be seen that (PII) uses only 1 to 2 kernels per representation of width 1 whereas (PII$''$) uses on average 6 kernels of width 1. Moreover, we consistently obtain representations of lower complexity when solving (PII)—see Figure 6.

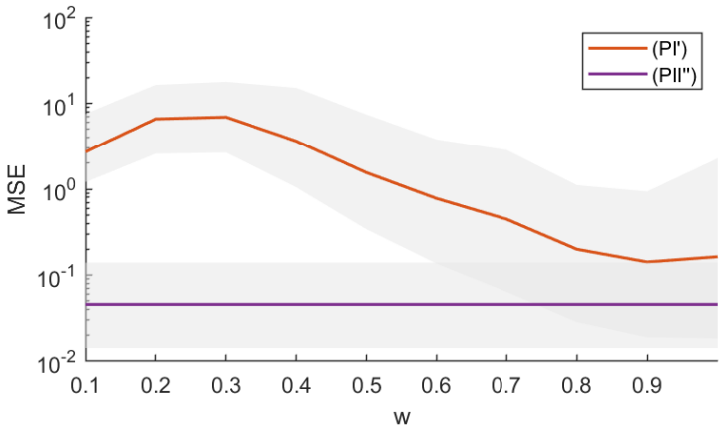



Figure 3: MSE obtained by (PII″) and (PI′) over 1000 realizations of random sampling of the signal in (2.27). (PI′) is solved over different values of $w$ over a grid on the interval $[0.1, 1]$. (PII″) finds the width as part of the algorithm and is presented for comparison with (PI′). The standard deviation around each mean is plotted in gray for both (PII″) and (PI′). The figure shows that the selection of the width within the algorithm gives the advantage of a lower mean generalization error.

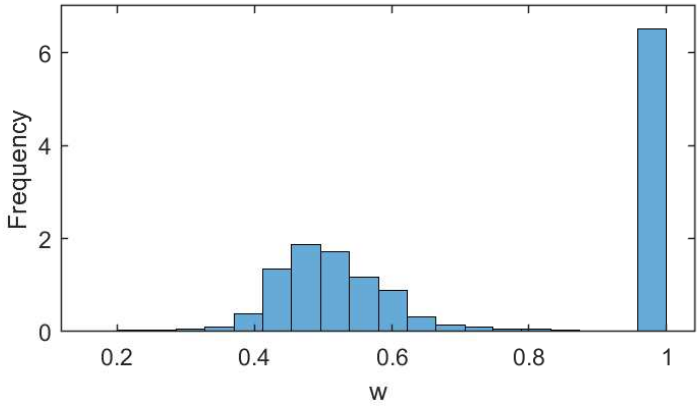



Figure 4: Histogram of the widths found using (PII″) over 1000 realizations of random sampling of the signal in (2.27). On average, 14 kernels were selected for the representation of the function out of which an average of 6 kernels have a width of 1.

### 2.4.1 Examining the Complexity of the Solution

So far we have shown that the complexity of the formulation can be reduced by moving centers in addition to moving the width. To further explore the effect of kernel centers on the complexity of the solution, we compare the performance of (PII′) to that of kernel orthogonal matching pursuit (KOMP) with pre-fitting (see Vincent and Bengio (2002); Koppel et al.

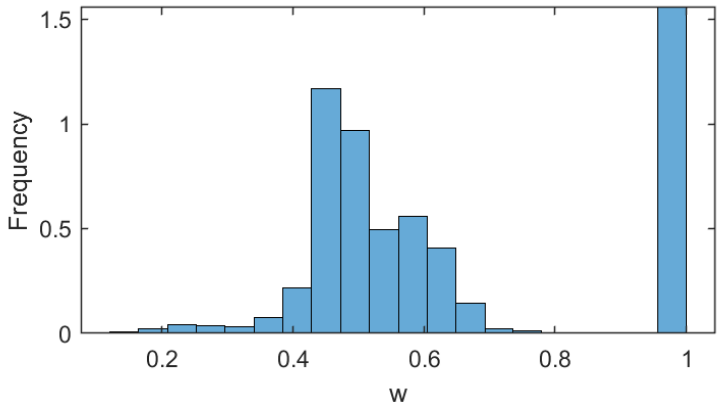



Figure 5: Histogram of the widths found using (PII) over 1000 realizations of random sampling of the signal in (2.27). On average, a representation had 6 kernels out of which between 1 and 2 kernels had a width of $w = 1$ and 4 kernels had a width in the interval $[0.384, 0.648]$.

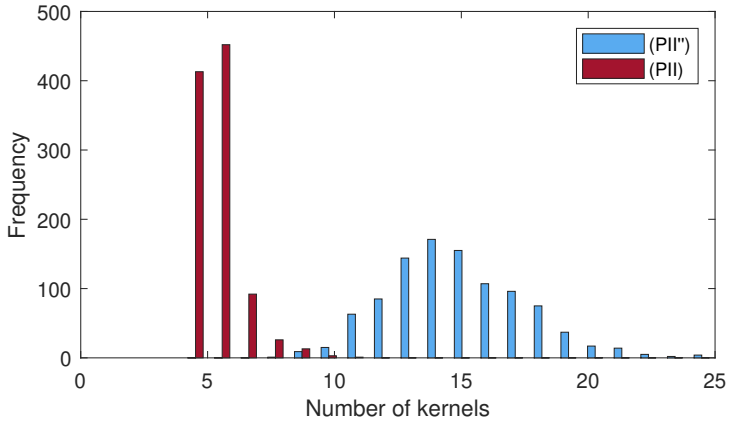



Figure 6: Histogram of the number of kernels in the representation of the estimated functions by solving problems (PII″) and (PII). (PII) achieves a lower complexity representation by moving the centers in addition to the widths.

(January 2019)), for a simulated signal as in (2.27). KOMP takes an initial function and a set of sample points and tries to estimate it by a parsimonious function of a lower complexity. As a backwards feature selection method, the algorithm starts by including all samples and then reduces the complexity of the function by reducing one feature at a time. The KOMP algorithm in Vincent and Bengio (2002); Koppel et al. (January 2019) was modified by changing the stopping criteria to be the estimation error, rather than the distance to the original function. This stopping criteria allows us to compare the sparsity needed to obtain

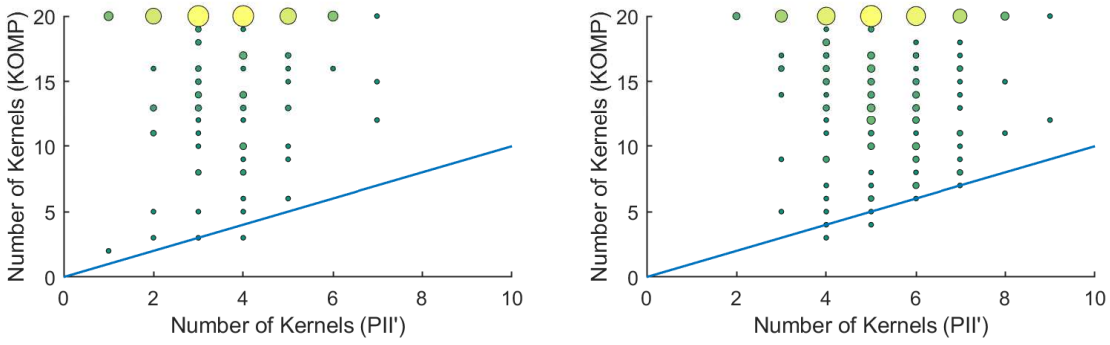



Figure 7: Comparison of the complexity of the representation of (PII′) and KOMP for a similar MSE over 1000 realizations. In Figure (a) 5 Gaussian functions were used to simulate the signal. In Figure (b) 10 Gaussian functions were used to simulate the signal. In both cases, (PII′) achieves a lower complexity for 99% of the realizations.

similar estimation error.

The signal was sampled from the function in (2.27) using $w_0 = 0.5$ and $m = [5, 10, 20]$ by generating $N = [2m, 4m, 6m]$ samples for each function, thus creating 9 different sample size and signal pairs. The problem in (PII′) was solved using $\gamma = 30$, $\eta_\lambda = 0.05$, $\eta_\mu = 0.1$ and $T = 1000$. Subsequently, a least squares algorithm was trained using kernels at the location found by our algorithm. Both our method and KOMP used $w = 0.5$ as the kernel hyper-parameter.

The number of kernels needed to obtain the same MSE is compared over 1000 realizations of each signal between (PII′) and KOMP. When the number of samples is at least 30, our method is able to find a sparser representation 100% of the times. In the cases with fewer samples, the problem is likely undersampled, such that the estimation of the function is more difficult. Figure 7 shows two cases in which 20 samples are simulated, where $m = 5$ and $m = 10$. In both cases, our method finds sparser representations in 99% of the realizations. When 10 kernels and 5 kernels are superimposed, our method finds a representation which is less sparse in only 0.4% and 0.3% of realization respectively. Lastly, when the signal is a weighted sum of 5 functions and only 10 samples are generated, our method cannot find a sparser solution for 3.7% of the realizations. The generalization MSE was compared between

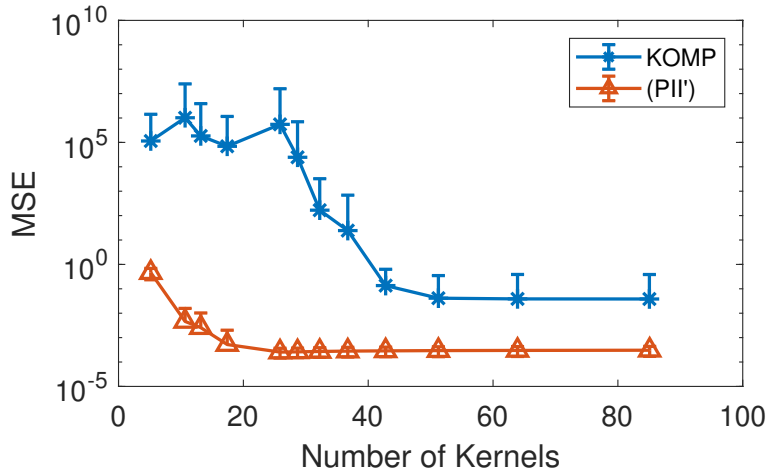



Figure 8: Generalization MSE as a function of number of kernels for KOMP and (PII′) over 1000 realizations of the signal in (2.27).

the two methods for different levels of sparsity. Figure 8 shows the changes in generalization MSE as the number of kernels used in the representation increases. 1000 realizations of a signal with $m = 10$ and a training set of size $N = 100$ were used. The ability of our method to place kernels at any location, beyond the training set, allows it to achieve significantly lower errors compared to KOMP at any sparsity level. As the number of kernels used increases, the difference in performance between the two methods decreases. At approximately 25 kernels, the performance of our method plateaus. Comparatively, KOMP achieves a plateau when the representation holds 50 kernels.

### 2.4.2 Varying Degrees of Smoothness

In the previous sections, we have only considered signals from functions belonging to one RKHS in the family of RKHSs with Gaussian kernels. In this section, we explore the effect of sample size on the complexity of the representation and the MSE on a signal of varying degrees of smoothness. To this end, a signal of varying smoothness is simulated using the following equation:

$$y_i = \sin(0.5\pi x_i^2) + \xi_i \tag{2.28}$$

where $\xi_i \in \mathcal{N}(0, 10^{-3})$ represents the noise.

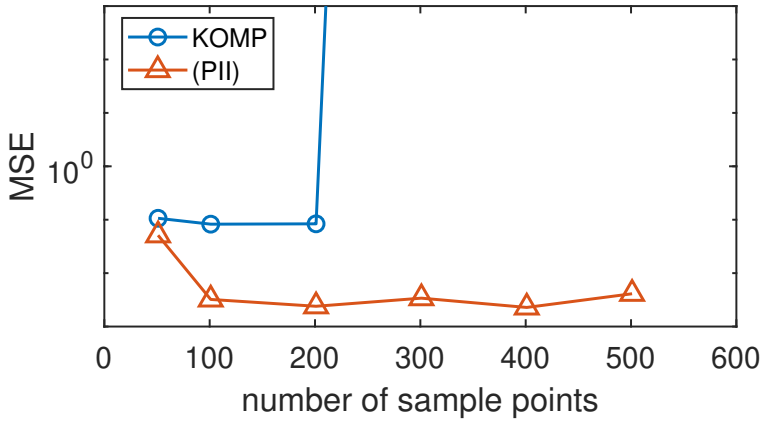



Figure 9: MSE for varying sample sizes using (PII) and KOMP with 26 kernels over 100 realizations of the signal in (2.28).

The solution of problem (PII) was compared to destructive KOMP, with the stopping criteria set to be the desired number of kernels rather than the distance from the original function. This stopping criteria allows us to have a fair comparison between our method and KOMP by using equally sparse functions. The problem in (PII) was solved using $\gamma = 2$, $\eta_\lambda = 0.001$, $\eta_\mu = 30$ and $T = 1000$. Sample sizes of 51, 101, 201, 301, 401, and 501 were created by uniformly sampling in the interval $[-5, 5]$. Test sets of 1000 samples randomly selected on the interval $[-5, 5]$ were created. Using the method of selecting kernel centers and widths by selecting the peaks of the function $\alpha(\mathbf{z}, w)$, our method finds a representation with 26 kernels regardless of the sample size. It can be seen in Figure 9 that in addition to the number of kernels being consistent across all sample sizes, the MSE is also consistent for our method. The MSE of the estimation using KOMP, however, increases as the sample size grows.

The problem of reducing features is a combinatorial problem which grows exponentially with the sample size. The backwards approach used by KOMP is a greedy approach which removes only one kernel at a time. As the sample size increases, there are more misleading paths of removal it can take. Additionally, it is only using kernels placed at the sample points, which means it will need more kernels when the true kernel is centered between two sample points. Figure (10) examines the complexity of the solution as measured by the number of kernels evaluations, required by (PII), KOMP, and (PI). Note that the complexity of the solution

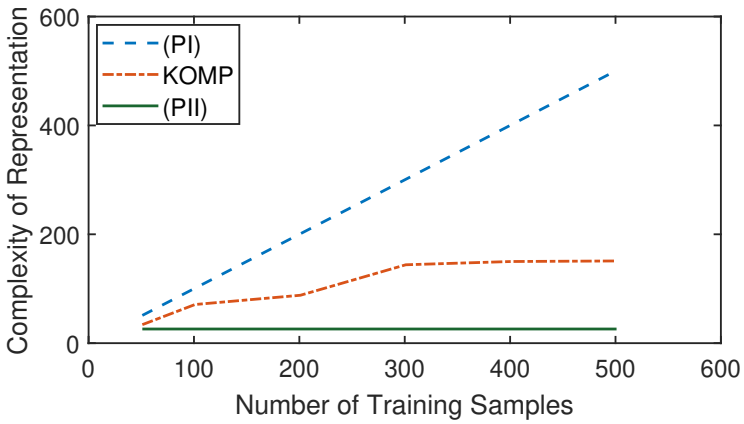



Figure 10: The complexity of the resulting RKHSs function as measured by the number of kernel evaluations needed.

of (PI) grows linearly with the sample size. In contrast, KOMP achieves a lower complexity representation, while maintaining the same MSE as (PII). However, since it uses a greedy heuristic and does not allow kernels to be placed at any other points than the sample points, it requires more kernels in its representation than the solution of (PII). In fact, the integral representation, by allowing both kernels of varying width and centers, is able to represent the signal using the same complexity regardless of the sample size.

### 2.4.3 User Localization Problem

In the remainder of this section, we will apply our method to real world application for which the class of functions the signal belongs to is unknown. We consider the problem of using RF signals to identify the location of a receiver. Specifically, given the Wi-Fi signal strength from seven routers, we wish to identify the room in which our receiver is located Narayanan et al. (2016). The signal strength varies depending on the location of the router. The signal was received from 7 routers spread throughout an office building. The data was collected using an Android device. At each location, the signal strength from each router was observed at $1s$ intervals. The data was then categorized into 4 groups, each representing the room in which the signal strength was observed. All the rooms are on the same floor, with the rooms representing the conference room, the kitchen, the indoor sports room, and work areas Narayanan et al. (2016). The goal is to be able to accurately detect the location

of the android device given the measured signal strength.

We use 10-fold cross-validation in order to estimate the generalization error of our algorithm as was used in Narayanan et al. (2016). The dataset was split into 10 sets of equal size with equal distribution of each label. At each turn one of the sets was used for testing while the others were concatenated and used to train the algorithm. This multiclass classification problem was solved using the one-vs-one strategy, which required 6 comparisons. The final class assignment is made through voting. Each comparison makes a prediction on the class of a sample, and thus casts a vote for a particular class. The class with the majority of votes is assigned to the sample. The cost function for this classification problem is

$$c(\mathbf{z}, \mathbf{y}) = \sum_i \max\{0, 1 - \mathbf{y}_i \mathbf{z}_i\} - \epsilon, \tag{2.29}$$

where $\epsilon$ controls the number of misclassifications allowed in the problem. In this test, $\epsilon$ was assigned a value of 2. Solving problem (PII) we obtain an average accuracy of 98%, similar to the performance observed in Narayanan et al. (2016), in which a fuzzy decision tree algorithm with 50 rules was used to obtain an accuracy of 96.65%. This result has been observed to be consistent over increasing values of the sparsity parameter $\gamma$.

### 2.4.4 MNIST Digits Classification

We use data of handwritten digits from the MNIST data set LeCun (1998), which consists of a training set of $60,000$ sample-label pairs and a testing set of $10,000$ images and labels. Each sample is a 28-pixel by 28-pixel grayscale image, which was vectorized to form 784 dimensional features. The labels are between 0 and 9 and correspond to the digit written. There are a total of 10 classes.

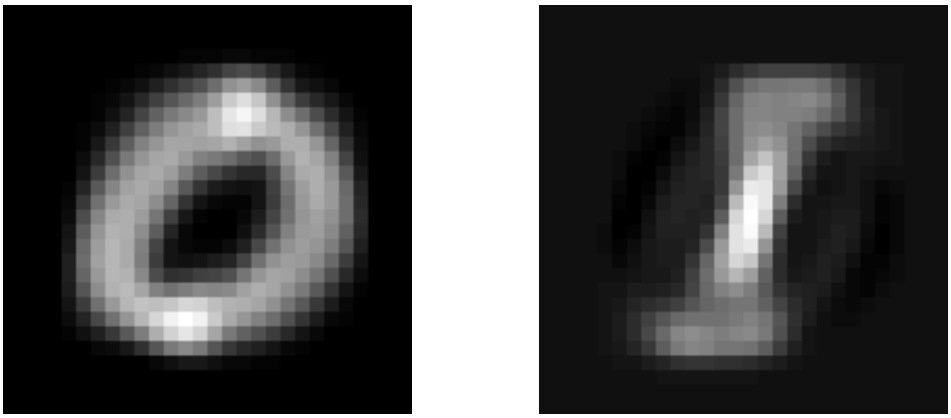
The number of features is too large to estimate the value of $\alpha(\mathbf{z}, w)$ at every $\mathbf{z} \in \mathbb{R}^{784}$. In order to find a set $\mathcal{X}$ over which $\alpha(\mathbf{z}, w)$ is defined, we use k-means with 400 clusters for each digit. Then $\mathcal{X}'$ in (PII$''$) is defined as the set of all cluster centers and the cost function in (2.29) is used. We then run our algorithm using a one-to-one strategy for multi-class classification

and achieve an accuracy of 98.12% for an average of 788 features per classification, which is comparable to the accuracy found using (PI′) using the training set as kernel centers and (PI′) using the centers found through k-means. The complexity of the representation can be further be reduced, however it comes at the cost of the classification accuracy.

Table 1: Classification results for (PI′) using the training samples as kernel centers and using centers selected from k-means and (PII″) using the centers selected from k-means

| Method | Number of Kernels per Classifier | Accuracy |
|---|---|---|
| (PI′) | 12000 | 98.83 % |
| (PI′) with k-means | 800 | 98.16 % |
| (PII″) | 788 | 98.12 % |
| (PII″) | 731 | 96.71 % |
| (PII″) | 53 | 85.66 % |

Although the dimensionality of the features in the original data makes the use of (PII) impractical, we can solve that problem, by projecting the data into a lower dimensional space by using principal component analysis (PCA). The formulation in (PII) has the advantage that the found kernel centers can give some intuition about the distribution of the signal. Particularly, in the case of digits, they can describe digits which are representative of written digits. To illustrate that we have performed the classification of the digits '0' and '1' using the first 3 principal components. The low dimensional feature set allows us to find the $\mathbf{x}$ which result in the highest value for $\alpha(w, \mathbf{x})$. From these points we can reconstruct the corresponding digits. Figure 11 shows the resulting images, which are not part of the initial written digit data set but rather represent an image that is closest to all written digit. The accuracy of the classification is 99.62%.



(a) (b)

Figure 11: Kernel centers obtained by solving (PII) with the highest value for $\alpha(\mathbf{z}, w)$ for each digit. These centers are representative of the digits, however, are distinct from any of the samples in the training set.

# CHAPTER 3

# Federated Classification using Parsimonious Representations of RKHS

The previous chapter presented a method for obtaining parsimonious representations in reproducing kernel Hilbert spaces. These have applications in federated learning by providing representations that are both communication efficient and computationally efficient.

In federated learning, a global model is trained by a central server using data gathered by a federation of agents Konečnỳ et al. (2016a,b); Li et al. (2020); McMahan et al. (2016). The problem is motivated by distributed networks generating vast amounts of data, such as those that arise when data is pooled together from mobile phones, wearable devices, or autonomous vehicles Smith et al. (2017); Anguita et al. (2013). There are several unique challenges to federated learning, such as respecting privacy of data, accounting for heterogeneous computational capabilities, or dealing with limited communication resources Konečnỳ et al. (2015, 2016b); Smith et al. (2017); Zhao et al. (2018); Bonawitz et al. (2019).

When dealing with limited communication resources, the systems Bonawitz et al. (2019); Konečnỳ et al. (2016b); Smith et al. (2017) challenges lead to additional statistical Konečnỳ et al. (2015, 2016b); Smith et al. (2017); Zhao et al. (2018) challenges. Because of the system challenges, it is impractical to transmit large data over the network. Consequently, a traditional learning approach with a central unit learning the global model is often impossible, and it is imperative for agents to transmit information about the problem without sending their entire data. Therefore, forming a global modal based on the global distribution is not a straightforward task, since each agent collects data over its own distribution.

Existing work on federated learning takes a distributed optimization approach. These attempt to form a global modal by sharing the gradient McMahan and Ramage (2017); Hard

et al. (2018); McMahan et al. (2016). These methods have the advantage that they preserve privacy by not sharing any collected data. However, they do not tackle the statistical challenge and require extensive communication due to sharing data over the network at each iteration. There have been efforts to tackle the statistical challenge as well, however these problems do end up sharing a small subset of data Zhao et al. (2018) or only work under strict conditions Li et al. (2019). In an effort to reduce communication load, methods have been developed which only share the gradient every few iterations Wang et al. (2019); Yu et al. (2019) or only share a subset of the gradient Shokri and Shmatikov (2015). Although these methods require less communication than traditional distributed methods, they are still iterative and have a communication load proportional to the number of iterations. The method, presented in this work, is fundamentally different from distributed learning. Although it does not guarantee privacy because it shares a subset of the data collected, it requires only a one time communication over the network from the agents.

The mechanism, presented in this work, reduces data sharing to a minimum while still allows the central server to learn a classifier that would be as good as the one that it would learn if all agents shared all of their data. This is achieved by sharing only a subset of the collected data, which is critical to the classification problem. The central unit having access to the critical samples is comparable to having access to all samples, because the samples which are not critical do not contribute to the global model. This is achieved by having each agent learn a local mode which detects the critical samples. The central unit receives the critical samples from each agent and trains a global model.

## 3.1 The centralized Learner

This section reviews the method in the previous chapter and presents the centralized learner problem. The classification problem is considered, when a training set is available, made up of $N$ feature-class pairs of the form $(\mathbf{x}_n, y_n) \in \mathcal{T}$. Features $\mathbf{x}_n \in \mathcal{X} \subset \mathbb{R}^p$ are real valued $p$-dimensional vectors and classes $y_n \in \{-1, +1\}$ are binary. To this end, we seek a method for finding a function approximation $f(x)$ such that $f(x_n)$ coincides with $y_n$ to the extent

possible. This is formally stated by introducing the loss function $\ell(f(x), y) = 1 - \epsilon - yf(x)$, a class function $\mathcal{C}$ and a function complexity measure $\rho(f)$ to define the optimization problem

$$\begin{aligned} P = \min_{f \in \mathcal{C}} \quad & \rho(f) \\ \text{s.t.} \quad & \frac{1}{N} \ell\Big(f(\mathbf{x}_n), y_n\Big) \leq 0, \quad (\mathbf{x}_n, y_n) \in \mathcal{T}. \end{aligned} \tag{PIII}$$

The constraints in (PIII) force the function $f$ to satisfy $f(\mathbf{x}_n) \geq 1 - \epsilon$ when $y_n = +1$ and $f(\mathbf{x}_n) \leq -(1 - \epsilon)$ when $y_n = -1$. The class function $\mathcal{C}$ and the complexity measure $\rho(f)$ constrain the variability of $f$ and dictate how it generalizes to unobserved samples $\mathbf{x}$. The problem (PII) presented in the previous chapter is well suited for these requirements.

To recall the low complexity RKHS representations, let $k(\mathbf{x}, \mathbf{s}; w)$ be a *family* of kernel functions in which $\mathbf{x} \in \mathbb{R}^p$ is a variable, $\mathbf{s} \in \mathbb{R}^p$ is a kernel center and $w \in \mathbb{R}$ is a kernel parameter. Further, consider a compact set of possible kernel parameters $\mathcal{W} \subset \mathbb{R}$, and a compact set of possible kernel centers $\mathcal{S} \subseteq \mathbb{R}^p$. The function class $\mathcal{C}$ is defined as

$$\mathcal{C} = \left\{ f \,:\, f(\mathbf{x}) = \int_{\mathcal{S} \times \mathcal{W}} \alpha(\mathbf{s}, w) k(\mathbf{x}, \mathbf{s}\,;\, w)\, d\mathbf{s} dw \right\}, \tag{3.1}$$

Then, recall the elastic net measure of complexity from Section 2.2, which supports both smoothness and sparsity,

$$\begin{aligned} \rho(f) &= \frac{1}{2} \|\alpha\|_{L_2} + \gamma \|\alpha\|_{L_0} \\ &= \int_{\mathcal{S} \times \mathcal{W}} \frac{1}{2} \alpha^2(\mathbf{s}, w) + \gamma\, \mathbb{I}\left[\alpha(\mathbf{s}, w) \neq 0\right]\, d\mathbf{s} dw, \end{aligned} \tag{3.2}$$

The low complexity RKHS classification problem is defined as (PIII) with the class function $\mathcal{C}$ given by (3.1) and the function complexity measure given by (3.2). This is a problem that

can be rewritten as an optimization over the coefficient function,

$$\begin{aligned} P = \min_{\alpha} \quad & \int_{\mathcal{S}\times\mathcal{W}} \frac{1}{2}\alpha^2(\mathbf{s}, w) + \gamma\, \mathbb{I}\left[\alpha(\mathbf{s}, w) \neq 0\right] \, d\mathbf{s}dw \\ \text{s.t.} \quad & \frac{1}{N}\ell\Big(f(\mathbf{x}_n), y_n\Big) \leq 0, \quad (\mathbf{x}_n, y_n) \in \mathcal{T} \\ & f(\mathbf{x}) = \int_{\mathcal{S}\times\mathcal{W}} \alpha(\mathbf{s}, w) k(\mathbf{x}, \mathbf{s}\,;\, w)\, d\mathbf{s}dw, \end{aligned} \tag{PC}$$

The problem (PC) differs from (PIII) in that it replaces the search for the function $f$ by a search for the function $\alpha$. The problems are otherwise equivalent – with function class $\mathcal{C}$ as per (3.1) and complexity measure $\rho$ as per (3.2) – in the sense that both attain the same optimal objective $P$ and we can recover the optimal function $f^*$ from the optimal coefficient $\alpha^*$ by evaluating $f(\mathbf{x})$ according to (3.1).

The constraints in (PC) specify the form of the function $f$ in terms of the coefficient function $\alpha$ and force the constraints $\ell(f(\mathbf{x}_n), y_n) \leq 0$ to be satisfied for all entries of the training set. Out of all the coefficient functions that satisfy these constraints, we search for the $\alpha^*$ with the lowest elastic net cost. This is expected to be a sparse coefficient function. We are therefore searching for a function $f$ that can be specified by as few kernels as possible while still passing within $\epsilon$ of all the elements of the training set. When we search for kernels to add to the representation, the search is over kernel centers $\mathbf{s} \in \mathcal{S}$ and kernel parameters $\mathbf{w} \in \mathcal{W}$. The latter allows, e.g., a search over kernel widths – see Peifer et al. (2020) for details.

**Remark 2.** Problem (PC) fits the classification function by using constraints, as opposed to a regularized minimization problem. The advantage of using a constraint problem is two-fold: it allows for the problem to be solved in the dual domain, and the solution of the dual gives us information about the critical samples to our learning problem. This concept is explained in more detail in sections 3.4 and 3.3.

**Remark 3.** The function class $\mathcal{C}$ is not an RKHS but is closely related. For a fixed kernel parameter $w$ in (3.1) the expression $f(\mathbf{x}) = \int_{\mathcal{S}} \alpha(\mathbf{s}, w) k(\mathbf{x}, \mathbf{s}; w) d\mathbf{s}$ is an integral representa-

tion of the RKHS generated by the kernel $k(\mathbf{x}, \mathbf{s}; w)$ Peifer et al. (2018, 2019, 2020). The use of this integral representation as opposed to the more traditional series representation $f(\mathbf{x}) = \sum_{j=1}^{J} a_j k(\mathbf{x}, \mathbf{s}_j; w)$ may seem an unnecessary complication, but it is actually a crucial simplification. The search for a sparse set of kernels is intractable with a series representation. But the problem in (PC) is tractable in the dual domain. As a byproduct of this more tractable formulation, we can also incorporate the kernel parameter $w$ to the search space and still guarantee tractability. This results in a problem that not only optimizes kernel placement but also kernel width and can even accommodate representations in unions of RKHS, i.e., representations having a mix of kernels of different widths Peifer et al. (2020).

## 3.2 Federated Learning

In the previous section, the centralized learning setting was reviewed. In this section, learning in a federated setting is considered, in which a centralized method for obtaining low complexity reproducing kernel Hilbert space data is collected by a group of agents over the space $\mathcal{X}$. The federation of agents must work together to find a global model over $\mathcal{X}$. To this end, the federation adopts the strategy of each agent learning a local model using the data it collects. From that model, the agent detects the critical samples to the classification problem. The agent sends only the critical samples to the central server, which learns the global model. Particularly, given a set of $N_i$ feature-class pairs of the form $(\mathbf{x}_n, y_n) \in \mathcal{T}_i$, agent $A_i$ solves the following problem

$$
\begin{aligned}
P_i = \min_{\alpha \in L_2} \quad & \int_{\mathcal{S}\times\mathcal{W}} \frac{1}{2}\alpha^2(\mathbf{s}, w) + \gamma\, \mathbb{I}\left[\alpha(\mathbf{s}, w) \neq 0\right] \, d\mathbf{s}dw \\
\text{s.t.} \quad & \frac{1}{N_i}\ell(f(\mathbf{x}_n), y_n) \leq 0, \quad (\mathbf{x}_n, y_n) \in \mathcal{T}_i \\
& f(\mathbf{x}) = \int_{\mathcal{S}\times\mathcal{W}} \alpha(\mathbf{s}, w) k(\mathbf{x}, \mathbf{s}\,;\, w)\, d\mathbf{s}dw.
\end{aligned}
\tag{Pi}
$$

In order to find the set of critical feature-class pairs $\tilde{\mathcal{T}}_i \subset \mathcal{T}_i$ and a model parameter to send to the central server. The central unit learns the problem using $\tilde{\mathcal{T}} = \cup_i \tilde{\mathcal{T}}_i$ such that $|\mathcal{T}| \gg |\tilde{\mathcal{T}}|$, where $\mathcal{T} = \cup_i \mathcal{T}_i$. Typically, each agent $A_i$ is not able to sample $\mathcal{X}$ entirely, but rather observes a subspace $\mathcal{X}_i$, however, the subspaces, observed by the agents, cover the

space $\mathcal{X}$, such that $\cup_i \mathcal{X}_i = \mathcal{X}$.

Notice, problems (PC) and (Pi) are minimizing the same objective function, however, (PC) has additional constraints due to a larger sample set. Problem (Pi) is limited to only samples from a specific subspace. Although this might seem like an initial disadvantage, solving a smaller problem can improve computational speed, whereas, solving (PC) requires a lot of information sharing from each agent which can become impractical. Moreover, by solving the dual problem of (Pi), we can obtain the critical samples of the classification problem. The central server uses the critical samples from the agents to find the global model. Formally, the central server solves the problem

$$\begin{aligned} PF = \min_{\alpha} \quad & \int_{\mathcal{S}\times\mathcal{W}} \frac{1}{2}\alpha^2(\mathbf{s}, w) + \gamma\, \mathbb{I}\left[\alpha(\mathbf{s}, w) \neq 0\right] \, d\mathbf{s}dw \\ \text{s.t.} \quad & \frac{1}{N_F}\ell(f(\mathbf{x}_n), y_n) \leq 0, \quad (\mathbf{x}_n, y_n) \in \tilde{\mathcal{T}} \\ & f(\mathbf{x}) = \int_{\mathcal{S}\times\mathcal{W}} \alpha(\mathbf{s}, w) k(\mathbf{x}, \mathbf{s}\,;\, w)\, d\mathbf{s}dw, \end{aligned} \tag{PF}$$

where $N_F$ is the number of critical samples in $\tilde{\mathcal{T}}$. Notice, problems (PC) and (PF) solve the same problem, however (PF) solves it for a restricted data set. The goal is to find a subset $\tilde{\mathcal{X}}$ such that the solution of (PF) is close to that of (PC). The simplest solution is to make $\tilde{\mathcal{T}} = \mathcal{T}$, by pooling the data collected from all the agents and have the central server compute the global model. However, this solution involves a large amount of data to be sent, which could surpass the capabilities of the network.

## 3.3 Convergence of federated problem

In the previous section, we have presented a federated learning problem and proposed a method for each agent to solve a local problem and transmit a set of critical samples to a central server, which in turn produces a global model. In this section, we argue that solving (PF) becomes equivalent to solving (PC) as the training sample size grows. First, let's examine the solution to the centralized problem (PC).

### 3.3.1 Learning the Centralized Problem

Similarly to the agent problem (Pi) and the server problem (PF), the centralized problem (PC) is solved in the dual domain. In order to derive the dual problem, we first start by introducing the Lagrange multiplier $\boldsymbol{\lambda} \in \mathbb{R}_{+}^{N}$, associated with the inequality constraints. Formally, we introduce the Lagrangian

$$\begin{aligned} \mathcal{L}(\alpha, \boldsymbol{\lambda}) = & \frac{1}{2}\|\alpha\|_{L_2}^2 + \gamma\|\alpha\|_{L_0} \\ & + \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\lambda}_n \ell(f(\mathbf{x}_n), y_n). \end{aligned} \tag{3.3}$$

Similarly to the federated problem, the central learner obtains the dual function and the dual problem. The dual function is concave, and therefore the dual problem is solved using gradient descent. The gradients are computed by evaluating the constraints at the variable $\alpha_d$, which minimizes the Lagrangian $\alpha_d(\mathbf{s}, w) = \underset{\alpha \in L_2}{\operatorname{argmin}}\ \mathcal{L}(\alpha, \boldsymbol{\lambda})$. The variable $\alpha_d$ which minimizes the Lagrangian (3.3) has the following expression

$$\alpha_d(\mathbf{s}, w; \boldsymbol{\lambda}) = \begin{cases} \bar{\alpha}_d(\mathbf{s}, w; \boldsymbol{\lambda}) & (\bar{\alpha}_d(\mathbf{s}, w; \boldsymbol{\lambda}))^2 > 2\gamma \\ 0 & otherwise, \end{cases} \tag{3.4}$$

for which,

$$\bar{\alpha}_d(\mathbf{s}, w; \boldsymbol{\lambda}) = \frac{1}{N}\sum_{n} \boldsymbol{\lambda}_{i,n} y_n k(\mathbf{s}, \mathbf{x}_n, w). \tag{3.5}$$

Using (3.4), we can form a closed form expression for the dual function as the quadratic function, given the measure $m(\mathcal{X}, \mathcal{W}) = \int \mathbb{I}[\alpha_l(\mathbf{s}, w) \neq 0] d\mathbf{s}\, dw$

$$g(\boldsymbol{\lambda}) = -0.5\boldsymbol{\lambda}^{\top}\mathbf{Q}\boldsymbol{\lambda} + \frac{1}{N}\boldsymbol{\lambda}^{\top}(\mathbf{1} - \boldsymbol{\epsilon}) + m(\mathcal{X}, \mathcal{W}), \tag{3.6}$$

with $\mathbf{Q}$ being a positive definite matrix for which

$$\mathbf{Q}_{nm} = \int_{\mathcal{C}} \frac{1}{N^2} y_n y_m k(\mathbf{x}_n, \mathbf{s}; w) k(\mathbf{x}_m, \mathbf{s}; w) d\mathbf{s} dw, \tag{3.7}$$

where $\mathcal{C} = \{(\mathbf{s}, w) \mid \alpha_d(\mathbf{s}, w) \neq 0\}$.

### 3.3.2 Critical Samples

In section 3.4, we have claimed that the critical samples are determined by the values of the optimal dual variable. Particularly, given a set of samples, only the sample points which contribute to the classification model are considered critical. The following proposition shows that these critical points are not just particular to this training set, but to the classification problem in general.

**Proposition 3.** *Let $\alpha^*$ be the optimal variable of* (PC) *trained on data set $\mathbf{X}$, $\alpha'^*$ be the optimal variable of* (PC) *trained on data set $\mathbf{X}' = \mathbf{X} \setminus \{\mathbf{x}_n\}$ and $\hat{y}_n = \int \alpha'^*(\mathbf{s}, w) k(\mathbf{x}_n, \mathbf{s}, w) d\mathbf{s} dw$. The dual optimal variable associated with the nth sample, $\lambda_n^* = 0$ if and only if $1 - \epsilon - y_n \hat{y}_n < 0$ and the solutions to the data $\mathbf{X}$ and the data $\mathbf{X}'$ are equal.*

*Proof.* See Appendix A.3 ■

This proposition implies that if the federated learner (PF) and the centralized learner (PC) agree on the critical samples, then solving the two problems is equivalent. Furthermore, it is sufficient for the agent learner (Pi) to agree with the centralized learner despite only sampling from a subspace of $\mathcal{X}$. Next we will argue that this is in fact the case as the sample size grows.

We consider the case in which the subspaces sampled by the agents are not distinct, i.e., there exists at least one pair $i, j$ such that $\mathcal{X}_i \cap \mathcal{X}_j \neq \emptyset$. If all subspaces are disjoint, the problem becomes trivial. In this case, there is no need to form a global model because the agents do not gain useful information from other agents. Given a new sample, its classification can be done by simply finding the space to which it belongs and using the model of the respective agent. It should be noted that the problem of identifying the subspace is not trivial, yet in a federated learning setting, a new sample generally, belongs to the subspace of the agent that has collected it. Similarly, in the case in which agents sample over the same space, i.e., $\mathcal{X}_i = \mathcal{X}_j$, for all $i, j$ the need for sharing data across agents disappears. As the agents collect

more data their models will converge. We are, therefore, interested in the case for which there exist at least one pair of agents $A_i, A_j$ such that $\mathcal{X}_i \cap \mathcal{X}_j \neq \emptyset$ and $\mathcal{X}_i \neq \mathcal{X}_j$. We make the following hypothesis about the kernels centered at points belonging exclusively to one subspace

**Hypothesis 1.** *The overlap between any two partitions is large enough such that there exists a small $\xi > 0$ such that for all $\mathbf{x}_i \in \{\mathcal{X}_i \setminus \mathcal{X}_j\}$ and $\mathbf{s} \in \{\mathcal{X}_j \setminus \mathcal{X}_i\}$ and $w \in \mathcal{W}$*

$$k(\mathbf{x}_i, \mathbf{s}; w) \leq \xi. \tag{3.8}$$

This hypothesis implies that samples which uniquely belong to a subspace do not affect the models of other subspaces in the non-overlapping regions. Indeed, recall that the function $\alpha(\mathbf{s}, w)$ is a weighted sum of the kernels centered at the sample points, therefore a point located in a non overlapping part of a subspace the weighted sum of the kernels outside that partition will be small and not contribute significantly to the value of $\alpha$. Additionally, we make the following assumption about the choice of $\gamma$

**Hypothesis 2.** *Let $\mathcal{C} = \{(\mathbf{s}, w) \mid \alpha^*(\mathbf{s}, w) \neq 0\}$ be the support of the optimal value $\alpha^*(\mathbf{s}, w)$ of* (Pi). *We choose the variable $\gamma$ that leads to $\mathcal{C}$ being rich enough such that there exists a $\mu > 0$ for which*

$$\mathbf{Q} = \int_{\mathcal{C}} \mathbf{q}(\mathbf{s}, w)\mathbf{q}^\top(\mathbf{s}, w) ds\, dw \succeq \mu \mathbf{I} \tag{3.9}$$

*where the variable $\mu$ represents the smallest eigenvalue of the matrix $\mathbf{Q}$, and $\mathbf{q}_n(\mathbf{s}, w) = (1/N) y_n k(\mathbf{x}_n, \mathbf{s}; w)$.*

Notice that this hypothesis relies on the choice of $\gamma$. In theory, the choice of $\gamma$ does not cause the measure of $\mathcal{C}$ to go to zero. In practice, however, the choice of $\gamma$ becomes more important (see Remark 4). Furthermore, the hypothesis suggests that the matrix $Q$ is positive definite. As a consequence, the dual function is strongly concave near the optimal $\boldsymbol{\lambda}^*$ and we can formulate the following lemma.

**Lemma 2.** *The dual function $g(\boldsymbol{\lambda})$ for the problem in* (Pi) *is strongly concave near the optimal value, $\boldsymbol{\lambda}^*$. The strong concavity parameter $\mu$ as defined in Hypothesis 2 such that*

$$-g(\boldsymbol{\lambda}) \geq -g(\boldsymbol{\lambda}^*) - \nabla g(\boldsymbol{\lambda}^*)^T(\boldsymbol{\lambda} - \boldsymbol{\lambda}^*) + \frac{\mu}{2}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^*\|^2 \tag{3.10}$$

*and $\mu$ corresponds to the smallest eigenvalue of $\mathbf{Q}$.*

*Proof.* First recall the definition of the dual function:

$$g(\boldsymbol{\lambda}) = -\frac{1}{2}\boldsymbol{\lambda}^\top \mathbf{Q}\boldsymbol{\lambda} + \boldsymbol{\lambda}^\top(\mathbf{1} - \boldsymbol{\epsilon}) + m(\mathcal{X}, \mathcal{W}) \tag{3.11}$$

There must exist a variable $\delta > 0$ such that

$$g_i(\lambda^* + \delta) < g_i(\lambda^*) \tag{3.12}$$

with $\delta$ close to zero. We will show that there exists a $\mu > 0$ such that

$$g(\lambda^*) - g(\lambda^* + \delta) + \nabla g(\lambda^*)^\top(\delta) \geq \frac{\mu}{2}\|\delta\|^2 \tag{3.13}$$

We can calculate the value on the right side of the inequality, we assume that the support of the matrix $\mathbf{Q}$ is approximately equal for $\boldsymbol{\lambda}^*$ and $\boldsymbol{\lambda}^* + \delta$.

$$\begin{aligned}
&g(\lambda^*) - g(\lambda^* + \delta) + g(\lambda^*)^\top(\delta) = \\
&-\frac{1}{2}\sum_i\sum_j \lambda_i^*\lambda_j^*\mathbf{Q}_{ij} + \sum_i \lambda_i^*(1 - \epsilon_i) + m(\mathcal{S}, \mathcal{W}) \\
&+\frac{1}{2}\sum_i\sum_j \lambda_i^*\lambda_j^*\mathbf{Q}_{ij} + \sum_i\sum_j \lambda_i^*\delta_j\mathbf{Q}_{ij} + \frac{1}{2}\sum_i\sum_j \delta_i\delta_j\mathbf{Q}_{ij} \\
&-\sum_i \lambda_i^*(1 - \epsilon_i) - \sum_i \delta_i(1 - \epsilon_i) - m(\mathcal{S}, \mathcal{W}) \\
&-\sum_i\sum_j \lambda_i^*\delta_j\mathbf{Q}_{ij} + \sum_i \delta_i(1 - \epsilon_i) = \\
&\frac{1}{2}\boldsymbol{\delta}^\top \int_{\mathcal{C}} \mathbf{q}(s, w)\mathbf{q}^\top(s, w)dsdw\ \boldsymbol{\delta} \geq \frac{\mu}{2}\|\boldsymbol{\delta}\|^2
\end{aligned} \tag{3.14}$$

This proves that the dual function is strongly concave near the optimal value. ■

Notice that as Hypothesis 2 and Lemma 2 apply not only to problem (Pi), they also apply to problems (PC) and (PF). Given two hypotheses, we can state that the solutions of (PF) and (PC) converge to each other.

**Theorem 2.** *Let $\alpha_C^*$ and $\alpha_F^*$ be the solution to the problem* (PC) *and* (PF) *respectively. Given that hypotheses 1 and 2 hold, the two solutions converge, as the sample size grows*

$$\lim_{N\to\infty} |\alpha_F^*(\mathbf{s}, w) - \alpha_C^*(\mathbf{s}, w)| \to 0. \tag{3.15}$$

*Proof.* See Appendix A.4. ■

In order to understand the proof of this theorem, it is necessary to examine the two cases: the case of agents sampling the same space and the case of agents sampling disjoint spaces. Consider the case in which the two agents are observing completely separate spaces. We assume the two spaces are far apart such that the kernel value for two points in the separate spaces takes on a small value.

**Hypothesis 3.** *Let $\mathcal{X}_i$ and $\mathcal{X}_j$ be two subspaces of $\mathcal{X}$ which do not overlap ($\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$) and $w \in \mathcal{W}$. Then for $\xi$ from Hypothesis 1 the following holds*

$$k(\mathbf{x}_i, \mathbf{x}_j; w) < \xi, \text{ for all } w \in \mathcal{W}, \mathbf{x}_i \in \mathcal{X}_i, \mathbf{x}_j \in \mathcal{X}_j. \tag{3.16}$$

Notice that this hypothesis is similar to Hypothesis 1 and implies that samples from one subspace do not affect the solution of another subspace. From this assumption, we can formulate the following lemma about the global dual function with respect to the local dual functions.

**Lemma 3.** *Given a group of agents which sample separate spaces as dictated by hypothesis*

*3, let $g$ be the global dual function and $g_i$ be the agent dual function for agent $i$. Then*

$$|g(\boldsymbol{\lambda}) - \sum_i g_i(\boldsymbol{\lambda}_i)| \leq \frac{2\xi m L}{N^2}, \tag{3.17}$$

*for any $\boldsymbol{\lambda} = N[\boldsymbol{\lambda}_1^\top/N_1, \ldots, \boldsymbol{\lambda}_K^\top/N_K]^\top$, where $L = \frac{N^2}{N_1,N_2}\sum_i \sum_{j\neq i} \boldsymbol{\lambda}_i^T \mathbf{J}\boldsymbol{\lambda}_j$ and $m$ is the measure of the support of the function $\alpha_d$. $\mathbf{J}$ is an all-ones matrix.*

*Proof.* See Appendix A.5 ■

Notice that the values of the dual variables are weighted by the number of samples which means $\boldsymbol{\lambda}/N = [\boldsymbol{\lambda}_1^\top/N_1, \ldots, \boldsymbol{\lambda}_K^\top/N_K]^\top$. This causes the primal variables to be at the same scale despite being a sum of kernels weighted by the number of samples. Therefore, we can establish the following theorem

**Theorem 3.** *Let $\alpha_C^*$ and $\alpha_i^*$ be the solution to the problem* (PC) *and* (Pi) *respectively. Given that hypotheses 3 and 2 hold, the two solutions converge, as the sample size grows*

$$|\alpha_C^*(\mathbf{s}, w) - \sum_i \alpha_i^*(\mathbf{s}, w)| \leq \frac{2\sqrt{2\xi m L}}{N\sqrt{\mu N}}. \tag{3.18}$$

*Proof.* See Appendix A.6 ■

This theorem establishes the relationship between the primal variables of (Pi) and (PC) over non-overlapping areas. In the case in which agents observe the same space, we formulate the following theorem.

**Theorem 4.** *Let $\alpha_i^*$ and $\alpha_j^*$ be the optimal variables to problem* (Pi) *solved by agent $i$ and $j$ respectively. The agents observe samples independently over the same distribution. Further, let $M \geq \|f\|^2$, $c$ be the Berry-Essen theorem constant, $\rho = \mathbb{E}_{\mathbf{x}}\left[|\lambda(\mathbf{x})y_{\mathbf{x}}k(\mathbf{x}, \mathbf{s}; w)|^3\right]$ and $\sigma^2 = \mathbb{E}_{\mathbf{x}}\left[|\lambda(\mathbf{x})y_{\mathbf{x}}k(\mathbf{x}, \mathbf{s}; w)|^2\right]$. Let $\mu = min(\mu_i, \mu_j)$ for which $\mu_i$ and $\mu_j$ are the minimum eigenvalue of $\mathbf{Q}_i$ and $\mathbf{Q}_j$ respectively. Then the difference between the solutions computed by*

*the two agents has the following bound*

$$|\alpha_i(\mathbf{s}, w) - \alpha_j(\mathbf{s}, w)| \leq 2\left(2\sqrt{\frac{M}{\mu N^{1.5}}} + \frac{c\rho}{\sigma^3\sqrt{N}}\right), \tag{3.19}$$

*where $N = min(N_i, N_j)$ is the minimum of the two samples sizes.*

*Proof.* See Appendix A.7 ■

Because agents sample the same space, as the sample size grows the solutions from two agents converge. In fact, if the solutions of two agents are reciprocally feasible, then they are equal (Lemma 4). The centralized learner can be viewed in this case as an agent which collects more samples, therefore the solution of an agent converges to that of the centralized learner as well.

**Lemma 4.** *Given two problems as in* (Pi) *with different sample sets, let $P_1$ and $P_2$ be the solutions to these problems If the two solutions $P_1$ and $P_2$ are reciprocally feasible, that is if the optimal variable $\alpha_1^*$ is feasible to the second problem and vice versa then the two solutions are equal:*

$$P_1 = P_2. \tag{3.20}$$

*Proof.* Let us first notice that the objective function is independent of sample size and is therefore equal for both problems let us denote it as

$$f_0(\alpha) = \int \frac{1}{2}\alpha^2(s, w) + \gamma\mathbb{I}\left(\alpha(s, w) \neq 0\right) ds\, dw. \tag{3.21}$$

Suppose now that $P_2 > P_1$ this implies that there exists an $\alpha_1$ which is feasible in the second problem such that $f_0(\alpha_1) = P_1 < P_2$, however, since $P_2 = \min_{\alpha} f_0(\alpha)$ by definition, it follows that $P_2 \leq P_1$. This implies that there exists an $\alpha_2$ which is feasible in the first problem, such that $f_0(\alpha_2) = P_2 < P_1$. However, $P_1$ is by definition optimal, and therefore it must be that

**Algorithm 2** Federated classification algorithm

**for** i
  agent $i$ samples the subspace $\mathcal{X}_i$
  agent $i$ solves (Pi) and calculate optimal $\boldsymbol{\lambda}_i^*$
  agent $i$ sends critical samples for which $\boldsymbol{\lambda}_{i,n}^* > 0$
**end**
central unit solves (PF)

$P_1 = P_2$.

■

**Remark 4.** *In Hypothesis 2 we make an assumption about the parameter $\gamma$ being chosen such that the space $\mathcal{C}$ is rich enough to assure strong concavity of the dual problem in Lemma 4. The strong concavity is guaranteed by the optimal $\alpha^*$ having a non-zero support measure. In theory, as long as the zero function, $f(x) = 0$ is not feasible because of the constraints, alpha is guaranteed to have non-zero support. Increasing parameter $\gamma$ shrinks the support of $\alpha$ and consequently reduces the value of the strong concavity parameter $\mu$. Although, the dual problem still has strong concavity, the lower value of $\mu$ makes it more difficult to solve and therefore the dual problem algorithm requires more iterations to converge.*

## 3.4 Learning the Federated Classification Problem

The federated classification problem requires the agents to solve their local problem (Pi) in order to find a local model and detect the critical samples. The critical samples to the classification problem are sent to the server. The server then forms the global model by solving (PF). The federated classification algorithm is summarized in Algorithm 2. The next section describes the algorithm for solving the agent problem.

### 3.4.1 The Agent Problem

The agent solves problem (Pi) in the dual domain. In order to derive the dual problem, agent $i$ defines the Lagrange multiplier $\boldsymbol{\lambda}_i \in \mathbb{R}_+^{N_i}$, associated with the inequality constraints.

Formally, the Lagrangian is characterized as

$$\begin{aligned}\mathcal{L}_i(\alpha, \boldsymbol{\lambda}_i) = &\int_{\mathcal{S}\times\mathcal{W}} \frac{1}{2}\alpha^2(\mathbf{s}, w) + \gamma\, \mathbb{I}\left[\alpha(\mathbf{s}, w) \neq 0\right]\, d\mathbf{s}dw \\ &+ \frac{1}{N_i}\sum_{n}^{N_i} \boldsymbol{\lambda}_{i,n}\ell(f(\mathbf{x}_{i,n}), y_{i,n}).\end{aligned} \tag{3.22}$$

Each element of the Lagrangian multiplier is associated with the loss over a single sample point. The Lagrangian is less than or equal to the primal function for any feasible $\alpha$. Therefore, by minimizing the Lagrangian over $\alpha$ each agent obtains a lower bound for the primal problem. This is called the dual function

$$g_i(\boldsymbol{\lambda}_i) = \min_{\alpha\in L_2} \mathcal{L}_i(\alpha, \boldsymbol{\lambda}_i). \tag{3.23}$$

The dual function is the minimum over a set of affine functions of $\boldsymbol{\lambda}$ and is therefore concave Boyd and Vandenberghe (2004). Additionally, it is a lower bound to the primal problem for any feasible function $\alpha$ which meets the constraints. Indeed, the dual function is the sum of the primal function and the constraints weighted by the Lagrangian multiplier. In order for a function $\alpha$ to be feasible, the constraints must be non-positive and therefore the dual function can be at most equal to the primal function. Maximizing the dual function results in the best lower bound. Moreover, when strong duality holds, the maximum value of the dual function is equal to the solution of the primal problem. This leads to the formulation of the dual problem

$$\underset{\boldsymbol{\lambda}_i\geq 0}{\text{maximize}} \quad g_i(\boldsymbol{\lambda}_i). \tag{Di}$$

Solving the dual problem provides a solution for the primal problem Peifer et al. (2020). Because the dual function is concave, the dual problem can be solved using gradient descent Boyd and Vandenberghe (2004). The gradients can be obtained by evaluating the constraints at $\alpha_d$, which minimizes the Lagrangian

$$\alpha_i^\star(\mathbf{s}, w, \boldsymbol{\lambda}_i) = \underset{\alpha\in L_2}{\text{argmin}}\ \mathcal{L}_i(\alpha, \boldsymbol{\lambda}_i). \tag{3.24}$$

In order to find $\alpha_i^\star$ we must minimize the function $\mathcal{L}_\alpha$, the term of the Lagrangian which depends on $\alpha$

$$\begin{aligned}\mathcal{L}_\alpha(\alpha, \boldsymbol{\lambda}_i) = \int \Bigg[ & \frac{1}{2}\alpha^2(\mathbf{s}, w) + \gamma\, \mathbb{I}[\alpha(\mathbf{s}, w) \neq 0] \\ & - \frac{1}{N_i} \sum_n^{N_i} \boldsymbol{\lambda}_{i,n} y_n \alpha(\mathbf{s}, w) k(\mathbf{x}_n, \mathbf{s}; w) \Bigg] d\mathbf{s} dw.\end{aligned} \tag{3.25}$$

The function in (3.25) can be minimized with respect to $\alpha$ for each variable $\mathbf{s}$ and $w$ separately Peifer et al. (2020). Therefore, the minimization of $\mathcal{L}_\alpha$ reduces to the minimization of a quadratic function with a discontinuity at $\alpha = 0$ and hence, we obtain a closed-form thresholding solution of $\alpha_d(\mathbf{s}, w)$

$$\alpha_i^\star(\mathbf{s}, w; \boldsymbol{\lambda}) = \begin{cases} \bar{\alpha}_i(\mathbf{s}, w; \boldsymbol{\lambda}) & (\bar{\alpha}_i(\mathbf{s}, w; \boldsymbol{\lambda}))^2 > 2\gamma \\ 0 & otherwise, \end{cases} \tag{3.26}$$

for which

$$\bar{\alpha}_i(\mathbf{s}, w; \boldsymbol{\lambda}) = \frac{1}{N_i} \sum_n \boldsymbol{\lambda}_{i,n} y_n k(\mathbf{s}, \mathbf{x}_n, w). \tag{3.27}$$

The gradient has the following expression

$$d_{\boldsymbol{\lambda}_{i,n}} = \nabla_{\boldsymbol{\lambda}_{i,n}} g_i(\boldsymbol{\lambda}_i) = \frac{1}{N_i} \ell(f_d(\mathbf{x}_n), y_n), \tag{3.28}$$

where $f_i^\star$ is given by:

$$f_i^\star = \int_{\mathcal{X}\times\mathcal{W}} \alpha_i^\star(\mathbf{s}, w) k(\mathbf{x}, \mathbf{s}; w)\; d\mathbf{s} dw. \tag{3.29}$$

Each agent starts by initializing the dual variable $\boldsymbol{\lambda}_i \in \mathbb{R}_+^{N_i}$ to a positive random value. The gradient of the dual function provides the direction of descent, however, it does not provide any information on how close we are to the maximum, nor does it provide any information about how long to move along that direction. Therefore, a small step size $\eta$ to move along the gradient such that the direction of descent is evaluated often. The variable is updated in

the direction of the gradient as follows

$$\boldsymbol{\lambda}_i(t+1) = \left[\boldsymbol{\lambda}_i(t) + \eta \; d(\boldsymbol{\lambda}_i)\right]_+ , \tag{3.30}$$

where $[m]_+ = max(0, m)$. The dual problem is constrained to only have non-negative values for $\boldsymbol{\lambda}_i$ and therefore the updates are restricted.

Once the gradient descent algorithm has converged, the critical samples are identified by examining the optimal dual variable $\boldsymbol{\lambda}_i^*$. Notice that the Lagrangian is the primal function to which the constraints are added weighted by the Lagrange multiplier $\boldsymbol{\lambda}_i$. For feasible $\alpha$, the constraints are always non-positive. Moreover, at the optimal dual variable, the constraints multiplied by the optimal dual variable have to be zero for strong duality to hold, which means that either the constraints or the dual variable are equal to zero. This is known as complementary slackness Boyd and Vandenberghe (2004). Hence, the dual variable is an indicator that certain constraints are difficult to satisfy:

$$\begin{cases} 1 - \epsilon - y_n \hat{y}_n = 0, & \boldsymbol{\lambda} > 0 \\ 1 - \epsilon - y_n \hat{y}_n < 0, & \boldsymbol{\lambda} = 0. \end{cases} \tag{3.31}$$

The solution to the primal problem (Pi), $\alpha_i^*(\mathbf{s}, w)$, can be found according to the following proposition

**Proposition 4.** *Let* $\boldsymbol{\lambda}_i^*$ *be the solution of* (Di)*, then the solution to problem* (Pi) *is given by* $\alpha_i^\star(\cdot, \cdot, \boldsymbol{\lambda}_i^*)$ *from* (3.26)*.*

A formal proof can be found in Peifer et al. (2020). Proposition 4 suggests that the solution to problem (Pi), $\alpha_i^*(\mathbf{s}, w)$ is a weighted sum of kernels centered at the sample points. Samples, for which the dual variable $\boldsymbol{\lambda}_{i,n}^* = 0$, do not contribute to the function $\alpha_i^*$ and therefore are not considered critical to the problem.

**Algorithm 3** Agent algorithm

Collects data over subspace $\mathcal{X}_i$
Initialize $\boldsymbol{\lambda}_i(0) > 0$ randomly
**for** $t = 0, \ldots, T$
Compute $\alpha_i(\mathbf{s}, w, \boldsymbol{\lambda}_i)$

$$\alpha_i^\star(\mathbf{s}, w, \boldsymbol{\lambda}_i) = \begin{cases} \bar{\alpha}_i(\mathbf{s}, w, \boldsymbol{\lambda}_i), & |\bar{\alpha}_i(\mathbf{s}, w, \boldsymbol{\lambda}_i)| > \sqrt{2\gamma} \\ 0, & \text{otherwise} \end{cases}$$

for $\bar{\alpha}_i(\mathbf{s}, w; \boldsymbol{\lambda}_i) = \frac{1}{N_i} \sum_n \boldsymbol{\lambda}_{i,n} y_n k(\mathbf{s}, \mathbf{x}_n, w)$
evaluate the gradient

$$d_{\boldsymbol{\lambda}_n}(t) = \frac{1}{N_i} \ell(f_d(x_n), y_n)$$

for which

$$f_d(x_n) = \int_{\mathcal{X} \times \mathcal{W}} \alpha_d(\mathbf{s}, w) k(\mathbf{x}_n, \mathbf{s}; w) d\mathbf{s} dw$$

Update local parameter

$$\boldsymbol{\lambda}_{i,n}(t+1) = [\boldsymbol{\lambda}_{i,n}(t) + \eta \; d_{\boldsymbol{\lambda}_n}(t)]_+$$

**end**
Let the local optimal dual variable be $\boldsymbol{\lambda}_i^\star = \boldsymbol{\lambda}_i(t+1)$
Determine critical sample pairs: $\tilde{\mathcal{T}}_i = \{(\mathbf{x}_n, y_n) \mid \boldsymbol{\lambda}_n^* > 0\}$
Send $\tilde{\mathcal{T}}_i$ and $\tilde{\boldsymbol{\lambda}}_i^* = \{\boldsymbol{\lambda}_n^* \mid \boldsymbol{\lambda}_n^* > 0\}$ to the server

### 3.4.2 The Server Problem

The server receives the critical sample pairs $\tilde{\mathcal{T}}_i$ from each agent along with the optimal dual variables $\tilde{\boldsymbol{\lambda}}_i^*$ and forms the training its set $\tilde{\mathcal{T}} = \cup_i \tilde{\mathcal{T}}_i$ which is used to solve problem (PF) in the dual domain. The Lagrange multiplier $\boldsymbol{\lambda}_F \in \mathbb{R}_+^{N_F}$ is defined in order to formulate the Lagrangian of (PF)

$$\begin{aligned} \mathcal{L}_F(\alpha, \boldsymbol{\lambda}_F) = & \int_{\mathcal{S} \times \mathcal{W}} \frac{1}{2} \alpha^2(\mathbf{s}, w) + \mathbb{I}\left[\alpha(\mathbf{s}, w) \neq 0\right] d\mathbf{s} dw \\ & + \frac{1}{N_F} \sum_n^{N_F} \boldsymbol{\lambda}_{F,n} \ell(f(\mathbf{x}_n), y_n). \end{aligned} \tag{3.32}$$

Similarly to (3.23) and (Di) the dual function $g_F(\boldsymbol{\lambda}_F)$ and the corresponding dual problem are established. The server solves its dual problem using gradient descent. The gradients are

**Algorithm 4** Server algorithm

Receive $\tilde{\mathcal{T}}_i$ from all agents $i = 1, \ldots, K$
Initialize $\boldsymbol{\lambda}_F(0) = [\tilde{\boldsymbol{\lambda}}_1, \ldots, \tilde{\boldsymbol{\lambda}}_K]$
**for** $t = 0, \ldots, T_F$
Compute $\alpha_F^\star(\mathbf{s}, w)$

$$\alpha_F^\star(\mathbf{s}, w, \boldsymbol{\lambda}_F) = \begin{cases} \bar{\alpha}_F(\mathbf{s}, w, \boldsymbol{\lambda}_F), & |\bar{\alpha}_F(\mathbf{s}, w, \boldsymbol{\lambda}_F)| > \sqrt{2\gamma} \\ 0, & \text{otherwise} \end{cases}$$

for $\bar{\alpha}_F(\mathbf{s}, w; \boldsymbol{\lambda}_F) = \frac{1}{N_F} \sum_n \boldsymbol{\lambda}_{F,n} y_n k(\mathbf{s}, \mathbf{x}_n, w)$
evaluate the gradient

$$d_{\boldsymbol{\lambda}_{F,n}}(t) = \frac{1}{N_F} \ell(f^\star(x_n), y_n)$$

for which

$$f^\star(x_n) = \int_{\mathcal{X} \times \mathcal{W}} \alpha_F^\star(\mathbf{s}, w) k(\mathbf{x}_n, \mathbf{s}; w) d\mathbf{s} dw$$

Update local parameter

$$\boldsymbol{\lambda}_{F,n}(t+1) = [\boldsymbol{\lambda}_{F,n}(t) + \eta \; d_{\boldsymbol{\lambda}_F,n}(t)]_+$$

**end**
Compute global $\alpha^*(\mathbf{s}, w) = \alpha^\star(\mathbf{s}, w, \boldsymbol{\lambda}_F(T_F + 1))$
Send global model to the agents

computed by evaluating the constraints of (PF) at $\alpha_F^\star = \text{argmin}_\alpha \, \mathcal{L}_F(\alpha, \boldsymbol{\lambda}_F)$

$$d_{\boldsymbol{\lambda}_F,n} = \nabla_{\boldsymbol{\lambda}_n} g_F(\boldsymbol{\lambda}_F) = \frac{1}{N_F} \ell(f^\star(\mathbf{x}_n), y_n), \tag{3.33}$$

where $f^\star$ is given by:

$$f^\star = \int_{\mathcal{X} \times \mathcal{W}} \alpha_F^\star(\mathbf{s}, w) k(\mathbf{x}, \mathbf{s}; w) \; d\mathbf{s} dw. \tag{3.34}$$

The server initializes the dual variable $\boldsymbol{\lambda}_F(0) = \left[\tilde{\boldsymbol{\lambda}}_1^*, \ldots \tilde{\boldsymbol{\lambda}}_K\right]$. Then for each iteration $t$, $\alpha_F^\star(\mathbf{s}, w, \boldsymbol{\lambda}_F(t))$ is computed and used to find the gradient according to (3.33). A small step size $\eta_F$ is chosen to move along the gradient and update the variables

$$\boldsymbol{\lambda}_F(t+1) = [\boldsymbol{\lambda}_F(t) + \eta_F \; d_{\boldsymbol{\lambda}_F}(t)]_+ , \tag{3.35}$$

where $[m]_+ = max(0, m)$.

## 3.5 Applications

In the previous section, we have proposed a federated learning model which (i) reduces the necessary communication complexity and (ii) converges to the omniscient unit solution as the sample size grows. In this section, we first show through a simulated signal that the solution of our (PF) converges to that of (PC) with as the sample size grows. Then, using an activity identification task, we demonstrate that our algorithm can significantly reduce the communication cost to the central unit without compromising the performance of the classification. For the classification task, we use the family of RKHSs with Gaussian kernels

$$k(\mathbf{x}, \mathbf{x}') = \exp\left\{\frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{2w^2}\right\}. \tag{3.36}$$

The width of the kernel is directly proportional to the hyper-parameter $w$.

To start, the effect of sample size on the generalization accuracy is examined on a simulated data set. To this end, we simulate a uniformly distributed signal and define the class membership for each sample as

$$y = \begin{cases} 1, & (\mathbf{x}' - \mathbf{c}_i)^\top \mathbf{A}(\mathbf{x}' - \mathbf{c}_i) \leq r_i \text{ for any } i \\ -1 & otherwise, \end{cases} \tag{3.37}$$

for which $r_1 = 9$, $r_2 = 30$, $\mathbf{c}_1 = [3, 0]^\top$, $\mathbf{c}_2 = [-10, 6]^\top$ and $\mathbf{A} = [1, 0; 0, 0.25]$. The space $\mathcal{X}$ is divided into 9 overlapping subspaces. Each agent collects data from only one subspace and forms its local model. There were 9 subspaces from which the agents collect the data. The setup of subspaces and class labels can be seen in Figure 12. After samples are assigned to a class, Gaussian noise is added to the samples $\mathbf{x} = \mathbf{x}' + \xi$, where $\xi \ \in \mathcal{N}(0, 0.2)$ in order to create noisy samples. A separate testing set of 1000 samples is created for the evaluation of the learner.

The performance of the federated learner (PF) is compared to that of the centralized learner (PC). Both methods used $\gamma = 25$, $\epsilon = 10^{-2}$ and a learning rate of $\eta = 0.1$. Figure 13 com-

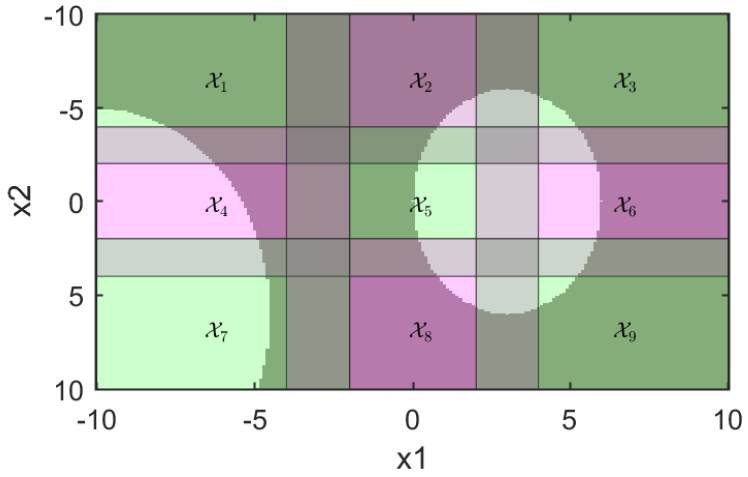



Figure 12: The simulated space $\mathcal{X}$. The subspaces sampled by each agent are colored either purple or green with the gray spaces being sampled by multiple agents. The class membership is determined by the brightness: the bright areas belong to class +1, and the darker areas belong to class -1.

pares the generalization accuracy of the two learners over training sample sizes ranging from 90 to 900. The average generalization accuracy was calculated over 100 repetitions. When the sample size is below 400, the federated classification learner has a better generalization accuracy. This is most likely due to the agents being able to learn a simpler problem despite having a small training set. As the sample size grows, the two solutions converge, which is reflected by the generalization accuracy converging.

### 3.5.1 Task Classification

We further evaluate our methods using biometric data Weiss et al. (2019) containing measurements from the accelerometer sensor from a smartphone. The study contained measurements from participants while performing various tasks, such as jogging, walking, writing, and typing.

The smartphone of each participant is considered an agent collecting data over its distribution. The agents collect data over different spaces since people don't perform the same activity the same way, e.g., some people walk faster, some people type slower, some write cursive, etc. Similarly, since all participants perform the same task, the spaces from which

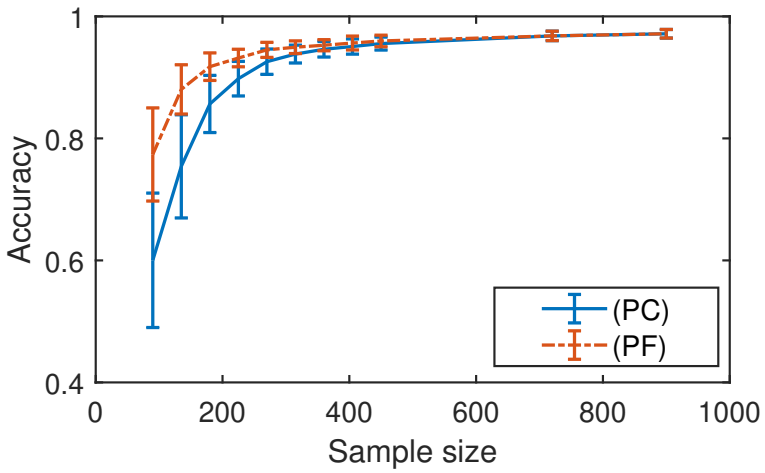



Figure 13: The average accuracy taken over 100 repetitions of randomized sampling of the federated learner (PF) and that of the centralized learner (PC) as a function of sample size.

the agents are collecting the data should not be distinct.

We examine the effect of the number of agents on the performance of our federated learner for the classification of running versus jogging. Agents are selected randomly from our training set and the data from each agent is randomly split into a training and a testing set. The time series from the phone's accelerometer is divided into 5 second intervals, with each interval considered a sample. The average value is taken such that each sample contains three features. Then we train our federated learner and the centralized learner and compare the accuracy on the test set. Both learners use the following parameters: $\gamma = 100$, $\eta = 0.1$, $T = 1000$ and $\epsilon = 0.5$. This procedure was repeated 100 times in order to obtain average performance. The federated learner (PF) and the centralized learner (PC) have comparable average accuracy. When the number of agents is increased to 51 agents the average accuracy of the federated learner is 77.35% and the average accuracy of the centralized learner is 75.29%.

The effect of the number of agents is evaluated on a second task: typing and writing. The learners use the following parameters : $\gamma = 150$, $\eta = 0.1$, $T = 500$ and $\epsilon = 0.5$. In this case, the average accuracy decreases as the number of agents increases for both the federated learner and the centralized learner. The performance of the learners could potentially be

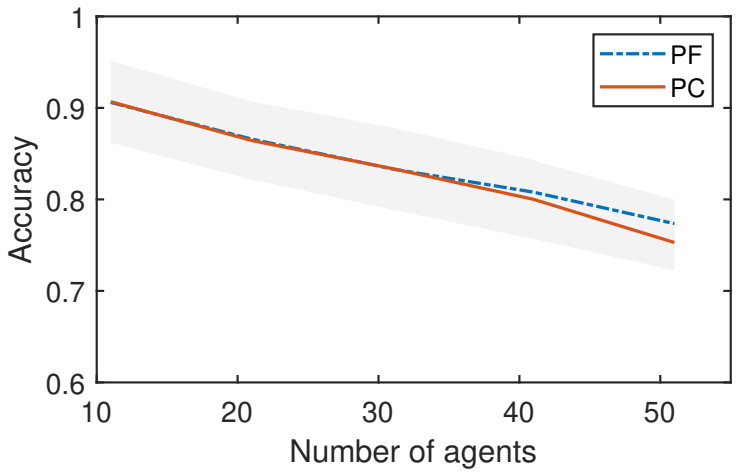



Figure 14: The performance of the federated learner and the centralized learner as a function of the number of agents. The two tasks were running and walking

improved by increasing the parameter $\epsilon$ for a larger number of agents.

Next, we examine the effect of the sparsity parameter $\gamma$ on the performance of the learners. Data from 10 participants is used to distinguish between the activities of walking and jogging. The regularizing parameter $\gamma$ which controls the complexity of the representation was varied to observe the effects on three metrics: accuracy of classification, the cost of communication, and the cost of the representation. The accuracy is measured as the percentage of correctly classified tasks. The communication cost is measured as the average number of samples that need to be transmitted over the channel. The representation cost is determined by the number of kernels used in the resulting global model.

The features are extracted from averaging over 5 second intervals. The data is randomly split to create a training set and a test set. The accuracy is evaluated on the test set. The parameters used by both agents are: $\eta = 0.1$, $T = 500$ and $\epsilon = 0.5$. The federated learner and the centralized learner are trained over 100 random splits and the resulting accuracy, representation cost, and communication cost is averaged over those repetitions. The average accuracy does not change for either learner with respect to the sparsity parameter, and both learners have similar performance, Figure 16 (a). This is expected since the algorithm can produce representations of varying complexity without sacrificing performance. The

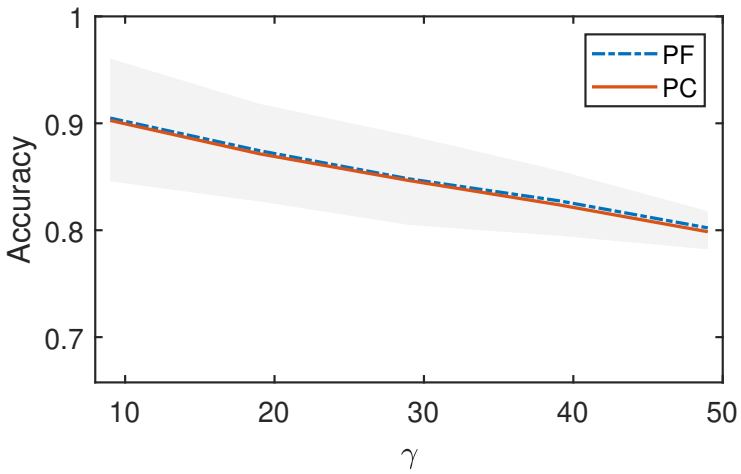
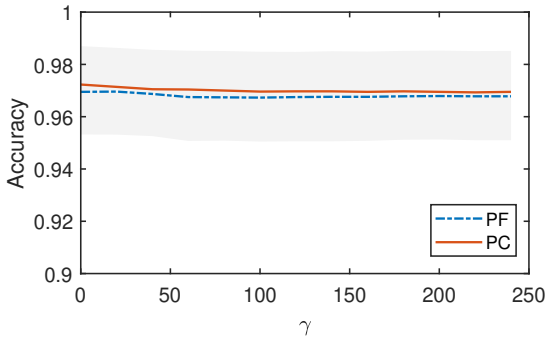



Figure 15: The performance of the federated learner and the centralized learner as a function of the number of agents.

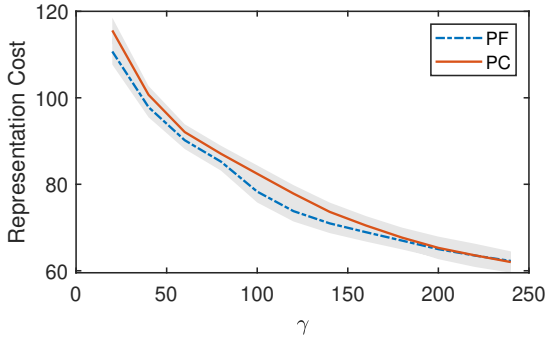
average representation cost of the global model is inversely proportional to the sparsity parameter. Both learners achieve similar representation costs as can be seen in Figure 16 (b). The communication cost of the federated learner is directly proportional to the sparsity parameter. This is expected since low complexity representations for $\alpha(\mathbf{s}, w)$ require more intricate kernel functions and therefore more samples. Therefore, there exists a trade-off between the complexity of the representation of the global model and the communication cost of sending data over the network. If sparsity of the global model is not a concern, the federated learner can achieve a communication cost that is 40% of the communication cost of the centralized learner (Figure 16 (c)).

We further validated our method by examining the problem of classification of writing and typing with data acquired from the phone accelerometer Weiss et al. (2019). The features are obtained by averaging over a 5 second time window. The performance of our federated learner was compared to that of the centralized learner on the three metrics: accuracy, communication cost, and representation cost. The data from the agents was split randomly using 100 repetitions. The parameters used by both agents are: $\eta = 0.1$, $T = 500$ and $\epsilon = 0.5$. The sparsity parameter $\gamma$ was varied between 0 and 240.

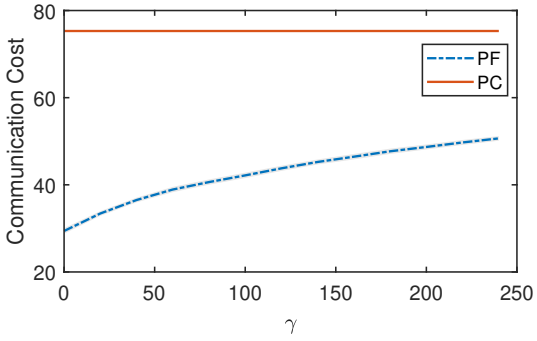
The federated learner (PF) and the centralized learner (PC) have similar accuracy, and their




(a)



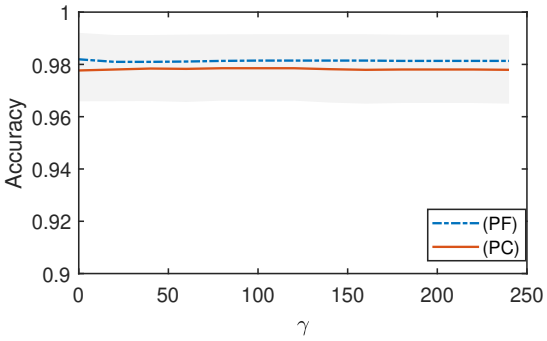

(b)



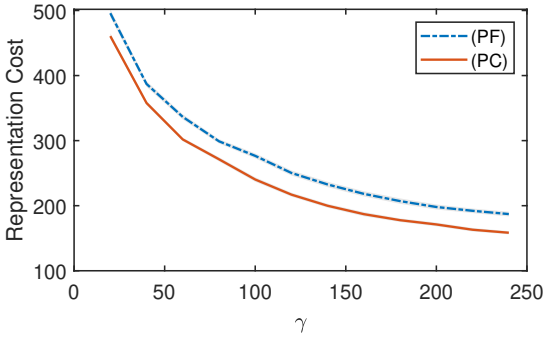

(c)

Figure 16: Classification of walking versus running using the federated learner and the centralized learner. (a) The accuracy of the federated classification learner and the centralized learner as a function of the sparsity parameter. (b) The representation cost of both learners as a function of the sparsity parameter. (c) The communication cost of transmitting data to the central unit for the federated learner and the centralized learner as a function of the sparsity parameter.

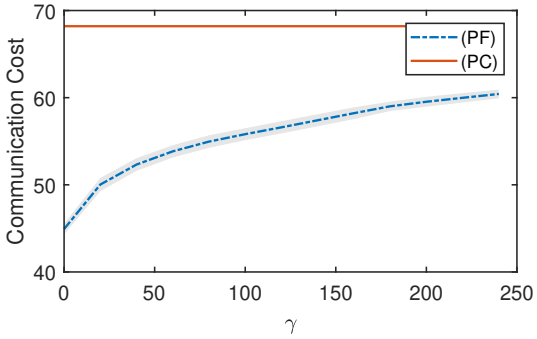
performance is not affected by the sparsity parameter. This implies that the functions needed to represent the class difference are sufficiently sparse. The sparsity parameter controls the complexity of the representation which can be seen in the representation cost (Figure 17,




(a)




(b)




(c)

Figure 17: Classification of writing versus typing using the federated learner and the centralized learner. (a) The accuracy of the federated classification learner and the centralized learner as a function of the sparsity parameter. (b) The representation cost of both learners as a function of the sparsity parameter. (c) The communication cost of transmitting data to the central unit for the federated learner and the centralized learner as a function of the sparsity parameter.

(b)). Both learners achieve similar representation costs. The advantage of the federated learner comes from reducing the communication cost Figure 17 (c). When sparsity is not required, the federated learner achieves a reduction of 64% in communication cost.

# CHAPTER 4

## Resilient Learning for Balancing Fit and Complexity

In the previous chapters, the focus has been on changing the representation in order to minimize complexity without affecting the fit. This chapter explores the problem of improving the fit in a resilient manner. Although there are advantages to encoding the fit objective in the constraints as shown in the previous chapters, this can lead to new challenges. As the number of constraints increases, it becomes increasingly more difficult to find specifications without causing the problem to become infeasible. Because there is interaction between constraints, it is difficult to select which particular constraint to relax.

A simple solution is to consider the empirical risk as a single constraint, however, the empirical risk problem works under the assumption that the samples available for training are taken from the same distribution as the data used for prediction. However, the sampling of the data may not reflect the true distribution, but be affected by biases (ex. gender bias, racial bias) in the training data Kodiyan (2019); Datta et al. (2015); Kay et al. (2015) or adversarial examples Lowd and Meek (2005); Gu et al. (2019); Shen and Sanghavi (2019). For example, a hiring tool only has access to the current employees, but is used to evaluate potential candidates. In a profession with a history of having a higher percentage of employees from a particular group, a hiring tool is likely to favor that particular group, despite the candidate pool being more diverse Datta et al. (2015).

In this work, the issue of constraint setting is addressed through resilient learning. We provide properties of the resilient learning formulation and propose to relax each constraint until the marginal cost of increasing the constraint becomes equal to the marginal complexity measure.

## 4.1 Resilient Statistical Learning

Consider data pairs $(\mathbf{x}, y)$ where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ is an input feature and $y \in \mathcal{Y} \subset \mathbb{R}$ is the corresponding output. Pairs $(\mathbf{x}, y)$ can be sampled according to a set of $m + q + 1$ distributions $\mathcal{D}_i$. Our goal is to learn a function $\phi : \mathcal{X} \to \mathcal{Z} \in \mathbb{R}$ that can be used to produce estimates $\phi(\mathbf{x})$ of outputs $y$. Different from classical (unconstrained) learning, we have a set of $m + q + 1$ convex performance functions $\ell_0 : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$ that evaluate the fitness of estimates produced by $\phi(\mathbf{x})$ relative to outputs $y$. We then define the constrained statistical learning (CSL) problem as the program

$$\begin{aligned} \mathsf{P} \;=\; \min_{\phi} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi(\mathbf{x}), y)\Big], \\ \text{s.t.} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\Big[\ell_i(\phi(\mathbf{x}), y)\Big] \leq 0, \qquad i = 1, \ldots, m, \\ & \ell_i(\phi(\mathbf{x}), y) \leq 0, \quad \mathcal{D}_i - \text{a.e.}, \qquad i = m+1, \ldots, m+q. \end{aligned} \tag{P-CSL}$$

Setting aside the constraints in (P-CSL); namely, if we make $m = q = 0$, we recover the classical unconstrained statistical learning problem in which we seek the function $\phi$ that best fits the outputs $y$ as dictated by the loss function $\ell_0$. The first $m$ constraints $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}[\ell_i(\phi(\mathbf{x}), y)] \leq 0$, with $i = 1, \ldots, m$, represent a set of statistical losses that constrain the function $\phi$. The remaining $q$ constraints, with $i = m+1, \ldots, m+q$, are pointwise as we require $\ell_i(\phi(\mathbf{x}), y) \leq 0$ for almost all pairs $(\mathbf{x}, y)$ drawn according to the probability distribution $\mathcal{D}_i$. We remark that the probability distributions $\mathcal{D}_i$ can be dense, implying that the number of constraints in (P-CSL) can be infinite. Problem (P-RLX) is introduced in Chamon and Ribeiro (2020) and arises in fair Benesty et al. (2008); Agarwal et al. (2018); Donini et al. (2018); Kearns et al. (2018); Zafar et al. (2019); Cotter et al. (2019) robust Madry et al. (2017); Sinha et al. (2017); Zhang et al. (2019) and safe Garcıa and Fernández (2015); Achiam et al. (2017); Paternain et al. (2019) learning among many other examples.

An important challenge in formulating meaningful (P-CSL) problems is the specification of constraints that make the problem feasible. This is an important distinction with respect to unconstrained learning. The optimal function $\phi$ in the latter case always exists. In

(P-CSL) it may be that no function $\phi$ satisfies the given requirements. In practice, landing on satisfiable requirements may necessitate the relaxation of some constraints relative to their initial specification. Then, in lieu of (P-CSL) we settle for the relaxed optimization problem

$$\begin{aligned} \mathsf{P}(\mathbf{u}) \;=\; \min_{\phi} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi(\mathbf{x}),y)\Big], \\ \text{s.t.} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\Big[\ell_i(\phi(\mathbf{x}),y)\Big] \le u_i, \qquad i=1,\dots,m, \\ & \ell_i(\phi(\mathbf{x}),y) \le u_i(\mathbf{x}), \quad \mathcal{D}_i-\text{a.e.}, \qquad i=m+1,\dots,m+q. \end{aligned} \tag{P-RLX}$$

In (P-RLX) the statistical loss constraints are relaxed to $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}[\ell_i(\phi(\mathbf{x}),y)] \le u_i$ for some nonnegative $u_i \ge 0$ and the pointwise constraints are relaxed to $\ell_i(\phi(\mathbf{x}),y) \le u_i(\mathbf{x})$ for some $\mathcal{D}_i$-square-measurable function taking nonnegative values $u_i(\mathbf{x}) \ge 0$. When we do this, the optimal cost $\mathsf{P}$ changes to $\mathsf{P}(\mathbf{u})$, where the variable $\mathbf{u}$ signifies all of the individual constraint relaxations $u_i$ and $u_i(\mathbf{x})$. For future reference, we denote as $\phi^*(\mathbf{x};\mathbf{u})$ the function that solves (P-RLX).

The function $\mathsf{P}(\mathbf{u})$ is known as the perturbation function of (P-CSL) as it describes the effect on the optimal primal value of perturbing the constraints. It is ready to verify that $\mathsf{P}(\mathbf{u})$ is a decreasing function of its argument, in the sense that for componentwise comparable arguments $\mathbf{u}_1 \preceq \mathbf{u}_2$ we have $\mathsf{P}(\mathbf{u}_1) \ge \mathsf{P}(\mathbf{u}_2)$. It is also ready to verify that $\mathsf{P}(\mathbf{u})$ is a convex function of its argument. This latter fact permits definition of a global Fréchet subdifferential as we formulate next.

**Definition 4.** Given the convex perturbation function $\mathsf{P}(\mathbf{u})$, we say that $\mathbf{D}\mathsf{P}(\mathbf{u})$ is a Fréchet subdifferential of $\mathsf{P}(\mathbf{u})$ if for all $v$ it holds

$$\mathsf{P}(\mathbf{v}) \ge \mathsf{P}(\mathbf{u}) + \Big\langle \mathbf{D}\mathsf{P}(\mathbf{u}), \mathbf{v}-\mathbf{u} \Big\rangle, \tag{4.1}$$

where $\langle \mathbf{D}\mathsf{P}(\mathbf{u}), \mathbf{v}-\mathbf{u}\rangle = \sum_{i=1}^{m} \lambda_i(v_i-u_i) + \sum_{i=m+1}^{m+q} \mathbb{E}_{\mathcal{D}_i}[\lambda_i(\mathbf{x})(v_i(\mathbf{x})-u_i(\mathbf{x})]$ denotes the inner product of $\mathbf{D}\mathsf{P}(u)$ and $\mathbf{v}-\mathbf{u}$.

If the perturbation function is Fréchet differentiable at $\mathbf{u}$, then, the Fréchet derivative is a Fréchet subdifferential. In this case, the subdifferential is unique save for a set of zero measure. In general, there may be points $\mathbf{u}$ at which the perturbation function is not Fréchet differentiable. At these points, the Fréchet subdifferential is not unique.

For sufficiently large relaxations $\mathbf{u}$, Problem (P-RLX) is feasible under mild conditions. The larger the relaxation, however, the less impact the constraints have on the optimal function $\phi^*(\mathbf{x}; \mathbf{u})$. Thus, the point at issue is coming up with reasonable definitions of relaxations that are sufficiently large but not too large. The insight we adopt in this work is to equate the marginal cost of the relaxation with the marginal cost of its effect on the optimal cost, as we formally define next.

**Definition 5.** Let $h(\mathbf{u})$ be a Fréchet differentiable convex nondecreasing function. The resilient learning problem is the relaxed problem (P-RLX) in which the constraint relaxation $\mathbf{u} = \mathbf{w}$ satisfies

$$\mathbf{D}h(\mathbf{w}) = -\mathbf{D}\mathsf{P}(\mathbf{w}) \tag{P-RSL}$$

where $\mathbf{D}h(\mathbf{w})$ is the Fréchet derivative of $h(\mathbf{w})$. We say that $\mathbf{w}$ is a resilient relaxation and that $\phi^*(\mathbf{x}; \mathbf{u})$ is a corresponding resilient statistical minimizer.

In Definition 5, the function $h(\mathbf{u})$ measures the cost of relaxation $\mathbf{u}$. The equilibrium defined by (P-RSL) states that we seek a relaxation for which the marginal price of modifying the constraint equates to its marginal benefit as measured by the change it implies in the optimal loss.

For a better understanding, consider the case of a problem with a single statistical constraint and no pointwise constraints – i.e., we make $m = 1$ and $q = 0$ in (P-RLX). Further, assume that we make $h(u_1) = h(u) = u^2/2$. We then have that the resilient relaxation $u = w$ is the one for which

$$w = -\frac{\partial \mathsf{P}(u)}{\partial u}\bigg|_{u=w}. \tag{4.2}$$

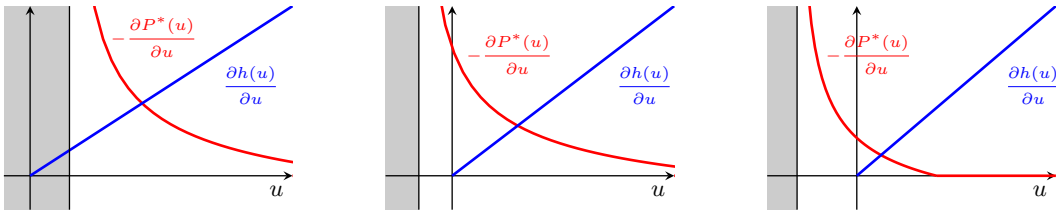



Figure 18: Resilient Equilibrium. We relax constraints to points where the marginal cost of the relaxation equals the marginal cost of its effect on the optimal cost (Definition 5). Constraints that are infeasible are relaxed to make them feasible (a) and constraints that are difficult to satisfy (b) are relaxed more than constraints that are easy to satisfy (c).

Figure 18 considers some prototypical situations that depict the intersection between the marginal cost of relaxation $u$ and the marginal benefit of relaxation $-\partial \mathsf{P}(u)/\partial u$. In all cases, the negative derivative diverges at the constraint value $u$ that makes the problem infeasible and decreases towards $\partial \mathsf{P}(u)/\partial u = 0$ as the constraint is relaxed. Case (a) represents a problem for which the original formulation (P-CSL) is infeasible. The resilient equilibrium (P-RSL) produces a version of (P-RLX) with a large relaxation. Case (b) illustrates a problem where (P-CSL) is feasible but close to infeasible. The resilient equilibrium (P-RSL) produces a version of (P-RLX) with a small relaxation. In case (c) (P-CSL) is feasible, and the constraint is barely active. The resilient equilibrium (P-RSL) produces a version of (P-RLX) that is barely different from the original (P-CSL).

In general, we can think of constraints associated with larger partial derivatives $\partial \mathsf{P}(\mathbf{u})/\partial u_i$ as more difficult to satisfy because their relaxation results in a larger marginal decrease in the optimal value $\mathsf{P}(\mathbf{u})$. The resilient equilibrium in (P-RSL) allows for a larger relaxation of these constraints. Conversely, constraints associated with smaller partial derivatives $\partial \mathsf{P}(\mathbf{u})/\partial u_i$ are more difficult to satisfy because their relaxation results in a larger marginal decrease in the optimal value $\mathsf{P}(\mathbf{u})$. The resilient equilibrium in (P-RSL) allows for a larger relaxation of these constraints.

In Section 4.2 we show that the equilibrium in (P-RSL) exists and introduce equivalent formulations that are important for algorithmic developments. Before that, we present examples to clarify ideas.

### 4.1.1 Resilient Classification

Consider a classification problem where we are given inputs $\mathbf{x}$ with associated categories $y = c \in \mathcal{C}$. We want to train a classifier $\phi(\mathbf{x})$ that outputs a vector with $\#(\mathcal{C})$ probabilities $\phi_c(\mathbf{x})$ associated to each class $c \in \mathcal{C}$. If pairs $(\mathbf{x}, y)$ have distribution $\mathcal{D}$ the standard formulation of this problem calls for minimizing a statistical risk to produce the classifier,

$$\phi_{\text{SRM}} \;=\; \operatorname*{argmin}_{\phi} \mathbb{E}_{\mathcal{D}}\Big[\ell_0(\phi(\mathbf{x}), y)\Big]. \tag{C-SRM}$$

The specific choice of loss is not central to the forthcoming discussion but to fix ideas think of the cross-entropy loss $\ell_0(\phi(\mathbf{x}), y) = -\sum_{c\in\mathcal{C}} \mathbb{I}(y = c) \log(\phi_c(\mathbf{x}))$.

A possible constrained formulation of the classification problem in (C-SRM) is to incorporate pointwise loss constraints as well. For instance, we set a threshold $\Delta$ and require that the losses of all pairs $(\mathbf{x}, y)$ satisfy $\ell_0(\phi(\mathbf{x}), y) \leq \Delta$. This leads to the following instantiation of (P-CSL),

$$\begin{aligned} \phi_{\text{CSL}} \;=\; \operatorname*{argmin}_{\phi} \quad & \mathbb{E}_{\mathcal{D}}\Big[\ell_0(\phi(\mathbf{x}), y)\Big], \\ \text{s.t.} \qquad & \ell_0(\phi(\mathbf{x}), y) - \Delta \leq 0, \quad \mathcal{D} - \text{a.e.} \end{aligned} \tag{C-CSL}$$

The merit of (C-CSL) is that the loss is required to be small for almost all pairs $(\mathbf{x}, y)$. The constrained formulation gives more weight to unusual observations, as it requires the classifier to make a correct classification. This is fundamentally different from (C-SRM) in which all observations are weighted equally. Unusual observations can get washed out in the average loss. We may say that (C-CSL) is a *robust* formulation.

The weakness associated with giving more weight to unusual observations is that (C-CSL) can give too much weight to unusual observations. This is a drawback because some unusual observations may be outliers that are better off discarded. Another weakness of (C-CSL) is that feasibility is not guaranteed. For the constraints to have an effect in the classifier $\Delta$ has to be small. But as we tighten $\Delta$ it is possible – indeed, likely – that there is no classifier $\phi$ that can satisfy the requirement $\ell_0(\phi(\mathbf{x}), y) \leq \Delta$ almost everywhere.

To mitigate these effects we introduce a resilient formulation with $h(\mathbf{u}) = \mathbb{E}_{\mathcal{D}}[u^2(\mathbf{x})/2]$. Since the Fréchet derivative of this relaxation cost function is $\mathbf{D}h(\mathbf{u}) = \mathbf{u}$ we obtain the following instantiation of the resilient learning problem of Definition 5,

$$\begin{aligned} \phi_{\text{RES}} \;=\; \underset{\phi}{\text{argmin}} \quad & \mathbb{E}_{\mathcal{D}}\Big[\ell_0(\phi(\mathbf{x}), y)\Big], \\ \text{s.t.} \qquad & \ell_0(\phi(\mathbf{x}), y) - \Delta \leq w(\mathbf{x}), \quad \mathcal{D} - \text{a.e.}, \\ & \mathbf{w} = -\mathbf{D}\mathsf{P}(\mathbf{w}). \end{aligned} \tag{C-RES}$$

In (C-RES), the pointwise constraints in the loss function are relaxed in proportion to their effect on the optimal cost. The larger the change in the optimal cost $\mathsf{P}(\mathbf{w})$ that is effected by the relaxation of a constraint, the larger the constraint is relaxed. Different from (C-CSL), the resilient problem in (C-RES) is always feasible. We can always find a classifier (C-RES) for some relaxation level. We also expect the resilient formulation in (C-RES) to bend the classifier to adapt to unusual entries but not so much as to (inaccurately) adapt to outliers. The numerical experiments in Section 4.6 corroborate this intuition.

We say that (C-RES) is a *resilient* formulation because it adapts to the data distribution. The unconstrained formulation in (C-SRM) is brittle. It doesn't respond well to unusual entries in the distribution. It's a tree with branches that break easily in a heavy storm. The constrained formulation in (C-CSL) is robust. It responds well to unusual entries but it does so at the cost of reducing performance. It's a tree with stiff branches that resist a heavy storm. The resilient formulation in (C-CSL) responds adaptively to the constraint difficulty. The tolerance $\Delta$ is not a hard constraint but a reference. If changing $\Delta$ yields a large performance payoff, the requirement is relaxed. It's a tree with pliable branches that bend with storms.

### 4.1.2 Assumptions

In forthcoming derivations, we variously make use of the following assumptions.

**A.1** (*Strongly Convex Objective*) There exists a $\mu > 0$ such that for any $\phi, \phi' \in \mathcal{F}$, all $\mathbf{x} \in \mathcal{X}$,

$y \in \mathbb{R}$ and an $\alpha \in [0, 1]$

$$\ell_0(\alpha\phi(\mathbf{x})+(1-\alpha)\phi', y) \leq \alpha\ell_0(\phi(\mathbf{x}), y)+(1-\alpha)\ell_0(\phi'(\mathbf{x}), y)-\frac{\alpha(1-\alpha)\mu}{2}(\phi(\mathbf{x})-\phi'(\mathbf{x}))^2 \quad (4.3)$$

**A.2** (*Constraint qualification*) There exist a finite relaxation $\mathbf{u} \preceq \infty$ and a function $\phi$ for which all constraints are met with some strictly positive margin $c > 0$,

$$\mathbb{E}_{\mathcal{D}_i}\Big[\ell_i(\phi(\mathbf{x}), y)\Big] \leq u_i - c, \qquad i = 1, \ldots, m, \quad (4.4)$$

$$\ell_i(\phi(\mathbf{x}), y) \leq u_i(\mathbf{x}) + c, \quad \mathcal{D}_i - \text{a.e.}, \qquad i = m+1, \ldots, m+q. \quad (4.5)$$

**A.3** (*Losses Properties*) The losses $\ell_i$ for $i = 1 \ldots q + m$ associated with the constraints are convex, B-bounded and L-Lipschitz.

**A.4** (*Square Integrable Primal Function*) The functional space $\mathcal{F}$ is such that any $\phi \in \mathcal{F}$

$$\int_{\mathcal{X}} |\phi(\mathbf{x})|^2 < \infty \quad (4.6)$$

**A.5** (*Square Integrable Perturbation Function*) The functional space $\mathcal{U}$ is such that any perturbation function $u_i \in \mathcal{U}$

$$\int_{\mathcal{X}_i} |u_i(\mathbf{x})|^2 < \infty \quad (4.7)$$

where $\mathcal{X}_i = \{\mathbf{x} \mid \mathbf{x} \sim \mathcal{D}_i\}$.

Assumption 1 restricts objective losses to be strongly convex. This is needed to guarantee that optimal functions $\phi^*(\mathbf{u}; \mathbf{u})$ are recoverable from Lagrangian maximizers.

Assumption 2 is a constraint qualification requirement that is needed to guarantee strong duality. It is satisfied if there exist a function $\phi$ that attains finite losses in expectation for $1 \leq i \leq m$ and for almost all $\mathbf{x}$ for $m + 1 \leq i \leq m + q$.

## 4.2 Equivalent Formulations of Resilient Statistical Learning

In regular convex optimization problems optimal Lagrange multipliers are subgradients of the perturbation function. We will see that this is true of (P-RLX) despite the presence of a dense set of constraints. To state this formally let $\boldsymbol{\lambda}$ represent $m$ Lagrange multipliers $\lambda_i$ for $1 \leq i \leq m$ and $q$ Lagrange multiplier functions $\lambda_i(\mathbf{x})$ for $m+1 \leq i \leq m+q$. We define the Lagrangian associated with (P-RLX) as

$$\begin{aligned} \mathcal{L}(\phi, \boldsymbol{\lambda}; \mathbf{u}) \;=\; & \mathbb{E}_{\mathcal{D}_0}\Big[\ell_0(\phi(\mathbf{x}), y)\Big] \\ & + \sum_{i=1}^{m} \lambda_i \Big[\mathbb{E}_{\mathcal{D}_i}\Big[\ell_i(\phi(\mathbf{x}), y)\Big] - u_i\Big] + \sum_{i=m+1}^{m+q} \mathbb{E}_{\mathcal{D}_i}\Big[\lambda_i(\mathbf{x})\Big[\ell(\phi(\mathbf{x}), y) - u_i(\mathbf{x})\Big]\Big] \end{aligned} \tag{4.8}$$

Observe that in the first $m$ constraints the perturbations $u_i$ and the Lagrange multipliers $\lambda_i$ are outside of the expectation operator. This is because the constraints are on average statistical losses. In the remaining $q$ constraints the perturbations $u_i(\mathbf{x})$ and the Lagrange multipliers $\lambda_i(\mathbf{x})$ are inside the expectation operator. This is because the constraints are pointwise. They are required to hold for almost all $\mathbf{x}$ with respect to the distribution $\mathcal{D}_i$.

From the Lagrangian in (4.8) we construct the dual function by minimizing over the primal variable $\phi$,

$$g(\boldsymbol{\lambda}; \mathbf{u}) = \min_{\phi} \mathcal{L}(\phi, \boldsymbol{\lambda}; \mathbf{u}). \tag{4.9}$$

And we further define the dual optimum by minimizing the dual function over nonnegative multipliers,

$$\mathsf{D}(\mathbf{u}) \;=\; \max_{\boldsymbol{\lambda} \succeq \mathbf{0}} g(\boldsymbol{\lambda}; \mathbf{u}) \;=\; g(\boldsymbol{\lambda}^*(\mathbf{u}); \mathbf{u}), \tag{D-RLX}$$

where in the second equality we defined $\boldsymbol{\lambda}^*(\mathbf{u})$ as a dual variable that maximizes the dual function for given perturbation $\mathbf{u}$. We remark that $\boldsymbol{\lambda}^*(\mathbf{u})$ is not necessarily unique. We prove next that optimal multipliers $\boldsymbol{\lambda}^*(\mathbf{u})$ are Fréchet subdifferentials of $\mathsf{P}(\mathbf{u})$.

**Proposition 5.** *Let $\boldsymbol{\lambda}^*(\mathbf{u})$ be a dual variable that attains the dual maximum in* (D-RLX) *for given $\mathbf{u}$. If assumptions 2,3 and 5 hold, this multiplier is a Fréchet subdifferential of the*

*perturbation function of* (P-RLX)*,*

$$\boldsymbol{\lambda}^*(\mathbf{u}) = -\mathbf{D}\mathsf{P}(\mathbf{u}). \tag{4.10}$$

*Proof.* The proof follows the same steps used to prove that optimal Lagrange multipliers are subgradients of the dual function in problems with a finite number of constraints; see, e.g., (Boyd and Vandenberghe, 2004, Section 5.6.2). We just need to verify that the proof holds when we consider dense sets of pointwise constraints as in (P-RLX). Consider then relaxation $\mathbf{u}$. Since the losses are convex and constraint qualifications are met by hypothesis we have no duality gap. Then, $\mathsf{P}(\mathbf{u}) = \mathsf{D}(\mathbf{u})$ and as per the definition of the dual function in (4.9) we can write

$$\mathsf{P}(\mathbf{u}) \;=\; \mathsf{D}(\mathbf{u}) \;=\; g(\boldsymbol{\lambda}^*;\mathbf{u}) \;=\; \min_{\phi} \mathcal{L}\Big(\,\phi, \boldsymbol{\lambda}^*(\mathbf{u}); \mathbf{u}\,\Big) \;\leq\; \mathcal{L}\Big(\,\phi, \boldsymbol{\lambda}^*(\mathbf{u}); \mathbf{u}\,\Big) \tag{4.11}$$

where the inequality is true for any function $\phi$. We particularize this inequality to a function $\phi^*(\cdot;\mathbf{v})$ that attains the minimum of (P-RLX) for relaxation $\mathbf{v}$. We can therefore write

$$\mathsf{P}(\mathbf{u}) \;\leq\; \mathcal{L}\Big(\,\phi^*(\cdot;\mathbf{v}), \boldsymbol{\lambda}^*(\mathbf{u}); \mathbf{u}\,\Big), \tag{4.12}$$

We now substitute the definition of the Lagrangian in (4.8) for the right hand side of (4.12) to write

$$\begin{aligned}\mathsf{P}(\mathbf{u}) \leq\; & \mathbb{E}_{\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x};\mathbf{v}), y)\Big] \\ & + \sum_{i=1}^{m} \lambda_i^*(\mathbf{u})\Big[\mathbb{E}_{\mathcal{D}_i}\Big[\ell_i(\phi^*(\mathbf{x};\mathbf{v}), y)\Big] - u_i\Big] + \sum_{i=m+1}^{m+q} \mathbb{E}_{\mathcal{D}_i}\Big[\lambda_i^*(\mathbf{x};\mathbf{u})\Big[\ell(\phi^*(\mathbf{x};\mathbf{v}), y) - u_i(\mathbf{x})\Big]\Big].\end{aligned} \tag{4.13}$$

The important property to observe in (4.13) is that we consider perturbation $\mathbf{u}$ and corresponding dual variable $\boldsymbol{\lambda}^*(\mathbf{u})$ while evaluating the Lagrangian at the function $\phi^*(\mathbf{x};\mathbf{v})$ that is primal optimal for perturbation $\mathbf{v}$. This latter fact implies that the following equality and

inequalities are true

$$\mathbb{E}_{\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x};\mathbf{v}),y)\Big] = \mathsf{P}(\mathbf{v}), \quad \mathbb{E}_{\mathcal{D}_i}\Big[\ell_i(\phi^*(\mathbf{x};\mathbf{v}),y)\Big] \leq -v_i, \quad \ell(\phi^*(\mathbf{x};\mathbf{v}),y) \leq v_i(\mathbf{x}). \quad (4.14)$$

Using the relationships in (4.14) in (4.13) we conclude that

$$\mathsf{P}(\mathbf{u}) \leq \mathsf{P}(\mathbf{v}) + \sum_{i=1}^{m} \lambda_i^*(\mathbf{u})\Big[v_i - u_i\Big] + \sum_{i=m+1}^{m+q} \mathbb{E}_{\mathcal{D}_i}\Big[\lambda_i^*(\mathbf{x};\mathbf{u})\Big[v_i(\mathbf{x}) - u_i(\mathbf{x})\Big]\Big]. \quad (4.15)$$

The two sums in the right hand side of (4.15) equal the inner product of multiplier $\lambda^*(\mathbf{u})$ with $\mathbf{v}-\mathbf{u}$. Implementing this substitution in (4.15) and reordering terms yields

$$\mathsf{P}(\mathbf{v}) \geq \mathsf{P}(\mathbf{u}) - \Big\langle \boldsymbol{\lambda}^*(\mathbf{u}), \mathbf{v}-\mathbf{u} \Big\rangle. \quad (4.16)$$

Comparing (4.16) with (4.1) we see that $\boldsymbol{\lambda}^*(\mathbf{u})$ satisfies Definition 4. ■

Using proposition (5) we can propose an alternative formulation of (P-RSL). Instead of equating the Fréchet derivative of $h(\mathbf{u})$ to the negative of the Fréchet subdifferential of the perturbation function, we equate the Fréchet derivative of $h(\mathbf{u})$ to the optimal Lagrange multiplier $\boldsymbol{\lambda}^*(\mathbf{u})$,

$$\mathbf{D}h(\mathbf{w}) = \boldsymbol{\lambda}^*(\mathbf{w}). \quad \text{(P'-RSL)}$$

The advantage of the resilient equilibrium in (P'-RSL) is that Lagrangian multipliers are accessible in primal-dual optimization algorithms; see Section 4.5. Conceptually, the relative difficulty of different constraints is given by their marginal effect in the primal objective. The relaxation of more challenging constraints has more of an effect on the optimal yield. Proposition 5 states that the different entries of the optimal Lagrange multiplier $\boldsymbol{\lambda}^*(\mathbf{u})$ are these relative measures of the difficulty of satisfying different constraints. The resilient equilibrium in (P'-RSL) relaxes constraints in proportion to the value of the multiplier. Thus, as is the case of (P-RSL), constraints are relaxed in proportion to their effect on the optimal yield.

Two other interesting properties of the resilient equilibrium are obtained by observing that a relaxation that achieves the resilient equilibrium satisfies

$$\mathbf{w} \;=\; \operatorname*{argmin}_{\mathbf{u}} \; \mathsf{P}(\mathbf{u}) + h(\mathbf{u}). \tag{4.17}$$

This has to be true because the relaxation cost function $h(\mathbf{u})$ and the perturbation function $\mathsf{P}(\mathbf{u})$ are both convex. Thus, a necessary and sufficient condition for $\mathbf{w}$ to be a minimizer of their sum is that there exists a subdifferential with $\mathbf{D}\mathsf{P}(\mathbf{w}) + \mathbf{D}h(\mathbf{w}) = \mathbf{0}$. Thus, any $\mathbf{w}$ that solves (4.17) is such that (P-RSL) holds for some subdifferential $\mathbf{D}\mathsf{P}(\mathbf{w})$ of the perturbation function. This property of the resilient equilibrium yields a very simple characterization that we summarize in the following proposition.

**Proposition 6.** *A perturbation* $\mathbf{w}$ *and a corresponding minimizer* $\phi^*(\mathbf{x};\mathbf{w})$ *of the relaxed problem* (P-RLX) *satisfy the resilient equilibrium condition in Definition 5 if and only if*

$$\begin{aligned}
\mathbf{w}, \phi^*(\mathbf{x};\mathbf{w}) = \operatorname*{argmin}_{\phi,\mathbf{u}} \quad & \mathbb{E}_{\mathcal{D}_0}\Big[\ell_0(\phi(\mathbf{x}), y)\Big] + h(\mathbf{u}), \\
s.t. \qquad & \mathbb{E}_{\mathcal{D}_i}\Big[\ell_i(\phi(\mathbf{x}), y)\Big] \leq u_i, \qquad\qquad i = 1, \ldots, m, \\
& \ell_i(\phi(\mathbf{x}), y) \leq u_i(\mathbf{x}), \quad \mathcal{D}_i - a.e., \quad i = m+1, \ldots, m+q.
\end{aligned} \tag{P"-RES}$$

*Proof.* In (4.17) the function $\mathsf{P}(\mathbf{u})$ is defined as the solution of (P-RLX). The proposition is true because a nested minimization over variables $\phi$ and $\mathbf{u}$ is equivalent to a joint minimization over $\phi$ and $\mathbf{u}$. To confirm that this holds here recall the definition of $\mathbf{w}$ as the resilient relaxation and of $\phi^*(\mathbf{x};\mathbf{u})$ as the minimizer of the relaxed problem (P-RLX) – which holds for all $\mathbf{u}$ and $\mathbf{w}$ in particular. As we have already shown, $\mathbf{w}$ is the solution of (4.17). We therefore have that for all perturbations $\mathbf{u}$

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x};\mathbf{w}), y)\Big] + h(\mathbf{w}) \leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x};\mathbf{u}), y)\Big] + h(\mathbf{u}). \tag{4.18}$$

Further observe that $\phi^*(\mathbf{x};\mathbf{u})$ is the minimizer of the relaxed problem (P-RLX) associated

with perturbation $\mathbf{u}$. We then have that for any feasible function $\phi$ we must have

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x};\mathbf{u}),y)\Big] + h(\mathbf{u}) \leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi(\mathbf{x}),y)\Big] + h(\mathbf{u}). \tag{4.19}$$

The bound in (4.19) is true for all $\phi$ and $\mathbf{u}$. In particular, it is true for the solution $\phi^*, \mathbf{u}^*$ of (P"-RES). Particularizing (4.19) to this pair and combining the resulting inequality with the bound in (4.18) we conclude that

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x};\mathbf{w}),y)\Big] + h(\mathbf{w}) \leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x}),y)\Big] + h(\mathbf{u}^*). \tag{4.20}$$

On the other hand, given that $\phi^*, \mathbf{u}^*$ solve (P"-RES) we have that for all feasible $\mathbf{u}$ and $\phi$

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x}),y)\Big] + h(\mathbf{u}^*) \leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi(\mathbf{x}),y)\Big] + h(\mathbf{u}). \tag{4.21}$$

In particular, this is true if we make $\mathbf{u} = \mathbf{w}$ and $\phi = \phi^*(\mathbf{x};\mathbf{w})$. We can then write,

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x}),y)\Big] + h(\mathbf{u}^*) \leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}\Big[\ell_0(\phi^*(\mathbf{x};\mathbf{w}),y)\Big] + h(\mathbf{w}) \tag{4.22}$$

For (4.20) and (4.22) to hold we must have that $\phi^*, \mathbf{u}^*$ is a solution of (P"-RES) if and only if $\mathbf{w}$ and $\phi = \phi^*(\mathbf{x};\mathbf{w})$ are a resilient perturbation and a corresponding resilient minimizer ■

Proposition 6 states that resilient learning is equivalent to adding a relaxation regularization to the objective. This is important for the derivation of algorithms (Section 4.5) and for the analysis of empirical versions of the statistical resilient learning problem (Section 4.3). Of the three equivalent definitions of resilient learning, (P"-RES) is the simplest. It shows that resilient relaxations can be found by solving an optimization problem whose complexity is comparable to the complexity of the original constrained statistical learning problem (P-CSL).

## 4.3 Parameterized Resilient Statistical Risk Minimization

In the previous, previous section we introduced the functional statistical risk minimization problem which is convex, has zero duality and has a clear solution. However, problem (P-RLX) is infinite dimensional and therefore, cannot be solved efficiently. However, we can formulate a finite dimensional problem by using parametrization. Let $f_\theta$ be a function characterized by the finite parameter $\theta$, such that for any function $\phi \in \mathcal{F}$ there exists a parameter $\theta \in \Theta$ and a corresponding function $f_\theta$ which is $\epsilon$-close.

$$|\phi(\mathbf{x}) - f_\theta(\mathbf{x})| \leq \epsilon, \quad \mathbf{x} \in \mathcal{X}. \tag{4.23}$$

Given the approximation, we can solve the finite dimensional problem ($\text{PI}_\epsilon$),

$$\begin{aligned} P_\epsilon^* \;=\; \min_{\theta\in\Theta} \quad & \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_0}\Big[\ell_0(f_\theta(\mathbf{x}),y)\Big] + h(\mathbf{u}) \\ \text{s.t.} \quad & \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\Big[\ell_i(f_\theta(\mathbf{x}),y)\Big] \leq u_i, \qquad i = 1,\dots,m, \\ & \ell_i(f_\theta(\mathbf{x}),y) \leq u_i, \mathcal{D}_i - \text{a.e.}, \qquad j = m+1,\dots,q. \end{aligned} \tag{$\text{PI}_\epsilon$}$$

Notice, that (P-RLX) and ($\text{PI}_\epsilon$) solve the same problem, however ($\text{PI}_\epsilon$) solves it over a smaller set of possible parameters $\theta$. Moreover, while the original problem is convex, due to the parametrization ($\text{PI}_\epsilon$) is no longer guaranteed to be convex and therefore strong duality is also no longer a guarantee. Nonetheless, we propose to solve the dual problem

$$D_\epsilon = \max_{\boldsymbol{\lambda}\geq 0} \min_{\mathbf{u}\in\mathcal{U},\theta\in\Theta} \mathcal{L}_\epsilon(\theta,\boldsymbol{\lambda};\mathbf{u}), \tag{$\text{DI}_\epsilon$}$$

for which $\mathcal{L}_\epsilon(\theta,\boldsymbol{\lambda};\mathbf{u})$ represents the Lagrangian of problem ($\text{PI}_\epsilon$)

$$\begin{aligned} \mathcal{L}_\epsilon(\theta,\boldsymbol{\lambda};\mathbf{u}) = \; & \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_0}\left[\ell_0(f_\theta(\mathbf{x}),y)\right] + h(\mathbf{u}) \\ & + \sum_{i=1}^{m} \boldsymbol{\lambda}_i\{\mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_i}\left[\ell_i(f_\theta(\mathbf{x}),y)\right] - u_i\} \\ & + \sum_{i=m+1}^{q} \int_{\mathbf{x}\in\mathcal{X}} \lambda_i(\mathbf{x})\{\ell_i(f_\theta(\mathbf{x}),y) - u_i(\mathbf{x})dx\}. \end{aligned} \tag{4.24}$$

Equation ($\text{DI}_\epsilon$) is a parameterized version of the dual of (P"-RES) and . Before we present the main result of this section, we want to make the following assumption about the optimal $\mathbf{u}^*$

**A.6** Let $\mathbf{u}^*$ be the optimal perturbation variable of problem (P"-RES), then $h(\mathbf{u}^* + L\epsilon) < \infty$ also.

Thew assumption states that our optimal variable $\mathbf{u}^*$ is not at least $L\epsilon$ away from the upper limit of $\mathcal{U}$. Generally, it is always possible to extend $\mathcal{U}$ to infinity and use a rapidly increasing cost function when $\mathbf{u}$ is large. We prove next that the optimal dual value is close to

**Proposition 7.** *Let $\theta$ be an $\epsilon$ parametrization of $\phi \in \mathcal{F}$, $P^*_{res}$ be the optimal value of* (P"-RES) *and $D_\epsilon$ be the optimal value of* ($\text{DI}_\epsilon$)*, then given assumptions 3 and 6 hold*

$$P^*_{res} \leq D_\epsilon \leq P^*_{res} - h(\mathbf{u}^*) + h(\mathbf{u}^* + L\epsilon) + L\epsilon, \tag{4.25}$$

*where $\boldsymbol{\lambda}^*$ is the optimal dual variable of* (P"-RES)*, $L$ is the Lipschitz constant.*

*Proof.* The left inequality stems from the fact that the space of parameterized functions is included in the functional space $\mathcal{H} = \{f_\theta \mid \theta \in \Theta\} \subset \mathcal{F}$. First notice that

$$D_\epsilon \geq \min_{\theta \in \Theta, \mathbf{u} \in \mathcal{U}} \mathcal{L}(\theta, \boldsymbol{\lambda}; \mathbf{u}), \text{ for all } \boldsymbol{\lambda} \in \mathbb{R}^{m+q}_+. \tag{4.26}$$

Then it follows, that

$$D_\epsilon \geq \min_{\theta \in \Theta, \mathbf{u} \in \mathcal{U}} \mathcal{L}(\theta, \boldsymbol{\lambda}^*; \mathbf{u}) \geq \min_{\phi \in \mathcal{F}, \mathbf{u} \in \mathcal{U}} \mathcal{L}(\phi, \boldsymbol{\lambda}^*; \mathbf{u}) = P_{res}, \tag{4.27}$$

For the upper bound of $D_\epsilon$, we will first show that the difference between the solution of (P"-RES) and ($\text{DI}_\epsilon$) is equal to the difference of the non-resilient problems with constant

$\mathbf{u} = 0$.

$$\begin{aligned} D_\epsilon =& \max_{\boldsymbol{\lambda}\geq 0}\{\min_{\phi\in\mathcal{F},\mathbf{u}\in\mathcal{U}} \mathcal{L}(\phi,\boldsymbol{\lambda};\mathbf{u}) + \min_{\theta\in\Theta,\mathbf{u}\in\mathcal{U}} \mathcal{L}(\theta,\boldsymbol{\lambda};\mathbf{u}) - \min_{\phi\in\mathcal{F},\mathbf{u}\in\mathcal{U}} \mathcal{L}(\phi,\boldsymbol{\lambda};\mathbf{u})\} \\ =& \max_{\boldsymbol{\lambda}\geq 0}\{\min_{\phi\in\mathcal{F},\mathbf{u}\in\mathcal{U}} \mathcal{L}(\phi,\boldsymbol{\lambda};\mathbf{u}) + \min_{\theta\in\Theta} \mathcal{L}(\phi,\boldsymbol{\lambda};0) \\ &+ \min_{\mathbf{u}\in\mathcal{U}} \mathcal{L}_{\mathbf{u}}(\boldsymbol{\lambda},\mathbf{u}) - \min_{\phi\in\mathcal{F}} \mathcal{L}(\phi,\boldsymbol{\lambda};0) - \min_{\mathbf{u}\in\mathcal{U}} \mathcal{L}_{\mathbf{u}}(\boldsymbol{\lambda},\mathbf{u})\} \\ =& \max_{\boldsymbol{\lambda}\geq 0}\{\min_{\phi\in\mathcal{F},\mathbf{u}\in\mathcal{U}} \mathcal{L}(\phi,\boldsymbol{\lambda};\mathbf{u}) + \min_{\theta\in\Theta} \mathcal{L}(\phi,\boldsymbol{\lambda};0) - \min_{\phi\in\mathcal{F}} \mathcal{L}(\phi,\boldsymbol{\lambda};0)\} \end{aligned} \tag{4.28}$$

Let $\phi^{'} = \text{argmin}_{\phi\in\mathcal{F}} \mathcal{L}(\phi,\boldsymbol{\lambda};0)$ and $\theta^{'} = \text{argmin}_{\theta\in\Theta} \mathcal{L}(\phi,\boldsymbol{\lambda};0)$, then, using Hölder inequality we can conclude

$$\begin{aligned} D_\epsilon =& \max_{\boldsymbol{\lambda}\geq 0}\{\min_{\phi\in\mathcal{F},\mathbf{u}\in\mathcal{U}} \mathcal{L}(\phi,\boldsymbol{\lambda};\mathbf{u}) + \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_0}\left[\ell_0(f_{\theta'}(\mathbf{x}),y) - \ell_0(\phi^{'}(\mathbf{x}),y)\right] \\ &+ \sum_{i=1}^{m} \boldsymbol{\lambda}_i\{\mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_i}\left[\ell_i(f_{\theta'}(\mathbf{x}),y) - \ell_i(\phi^{'}(\mathbf{x}),y)\right] \\ &+ \sum_{i=m+1}^{q+m} \int \boldsymbol{\lambda}_i\{\ell_i(f_{\theta'}(\mathbf{x}),y) - \ell_i(\phi^{'}(\mathbf{x}),y)\}dx \\ &\leq \max_{\boldsymbol{\lambda}\geq 0}\{\min_{\phi\in\mathcal{F},\mathbf{u}\in\mathcal{U}} \mathcal{L}(\phi,\boldsymbol{\lambda};\mathbf{u}) + \left(1 + \sum_{i=1}^{m}\boldsymbol{\lambda}_i + \sum_{i=m+1}^{q+m}\int \lambda_i(\mathbf{x})dx\right) \max_{i=0\ldots q} c_i(\phi^{'},\theta^{'},\mathbf{x},y)\}, \end{aligned} \tag{4.29}$$

where $c_i(\phi,\theta,\mathbf{x},y)$ represents the difference between the $i^{\text{th}}$ constraint of the functional problem and the parameterized problem

$$c_i(\phi,\theta,\mathbf{x},y) = \begin{cases} \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_i}\left[\ell_i(f_\theta(\mathbf{x}),y) - \ell_i(\phi(\mathbf{x}),y)\right], & \text{for } i = 0\ldots m \\ \ell_i(f_\theta(\mathbf{x}),y) - \ell_i(\phi(\mathbf{x}),y), & \text{for } i = m+1\ldots m+q. \end{cases} \tag{4.30}$$

The functions $c_i$ can because the losses are Lipschitz continuous and the parametrization is $\epsilon$ close.

$$c_i(\phi^{'},\theta^{'},\mathbf{x},y) \leq L \min_{\theta\in\Theta} \mid \ell_i(f_\theta(\mathbf{x}),y) - \ell_i(\phi^{'}(\mathbf{x}),y) \mid \leq L\epsilon. \tag{4.31}$$

The second inequality comes from the fact that there exists a $\theta$ such that $\mid \ell_i(f_\theta(\mathbf{x}),y) - \ell_i(\phi^{'}(\mathbf{x}),y) \mid \leq \epsilon$ therefore the minimum also satisfies the inequality. Notice that for $i \leq m$

it is sufficient for the parametrization to be close in expectation for the inequality to hold, however, when point-wise inequality constraints are present the requirement is stricter. By combining (4.29) and (4.31) we obtain

$$D_\epsilon \leq \max_{\boldsymbol{\lambda}\geq 0}\{\min_{\phi\in\mathcal{F},\mathbf{u}\in\mathcal{U}} \mathcal{L}(\phi,\boldsymbol{\lambda};\mathbf{u}) + (1+\|\boldsymbol{\lambda}\|_1)L\epsilon\} \tag{4.32}$$

The right side of the inequality is the solution of the dual problem of (P"-RES) perturbed by $-L\epsilon$

$$\begin{aligned} P_{L\epsilon} = \min_{\phi\in\mathcal{F},\mathbf{u}\in\mathcal{U}} \quad & \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_0}\left[\ell_0(\phi(\mathbf{x}),y)\right] + h(\mathbf{u}) + L\epsilon \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_i}\left[\ell_i(\phi(\mathbf{x}),y)\right] \leq u_i - L\epsilon, \quad i=1,\ldots,m, \\ & \ell_i(\phi(\mathbf{x}),y) \leq u_i - L\epsilon, \quad j=m+1,\ldots,m+q, \end{aligned} \tag{P-L$\epsilon$}$$

Let $\mathbf{u}^*$ and $\phi^*$ be the optimal variables of (P"-RES), then $\mathbf{u}^{'} = \mathbf{u}^* + L\epsilon$ and $\phi^*$ are feasible variables for problem (P-L$\epsilon$) such that

$$P_{L\epsilon} \leq \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_0}\left[\ell_0(\phi^*(\mathbf{x}),y)\right] + h(\mathbf{u}^*+L\epsilon) + L\epsilon = P_{res} - h(\mathbf{u}^*) + h(\mathbf{u}^*+L\epsilon) + L\epsilon \tag{4.33}$$

This concludes the proof. ■

Proposition 7 shows that, the solution of the parameterized dual problem (DI$_\epsilon$) is close to that of the functional resilient problem (P"-RES), when the parametrization is $\epsilon$-close. The parametrization gap between the two solutions depends on how fast the loss functions $\ell_i$ and the change in the cost function $h(\mathbf{u})$ change at the optimal $\phi^*$ and, $\mathbf{u}^*$ respectively. From proposition 7 we can infer the following corollary.

**Corollary 3.** *The optimal parameters $\theta^*$ and $\mathbf{u}^*$ obtained from solving problem* (DI$_\epsilon$)*, lead to a feasible function $f_{\theta^*}$ and $\mathbf{u}^*$ for problem* (PI$_\epsilon$)*.*

*Proof.* Corollary 3 can be proven by contradiction. Assume $\theta^*$ and $\mathbf{u}^*$ are infeasible, there-

fore, there exists at least one constraint $c_i(\theta, \mathbf{x}, y) - \mathbf{u}_i > 0$, where

$$c_i(\theta, \mathbf{x}, y) = \begin{cases} \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_i}\left[\ell_i(f_\theta(\mathbf{x}), y)\right], & \text{for } i = 0 \dots m \\ \ell_i(f_\theta(\mathbf{x}), y), & \text{for } i = m+1 \dots m+q. \end{cases} \tag{4.34}$$

then

$$D_\epsilon \geq \max_{\boldsymbol{\lambda}>0} \mathcal{L}_\epsilon(\theta, \boldsymbol{\lambda}; \mathbf{u}) = \infty \tag{4.35}$$

where $\boldsymbol{\lambda}_i^* = \infty$. At the same time, we know from proposition 7 that

$$\begin{aligned} D_\epsilon &\leq P_{res} + h(\mathbf{u}^* + L\epsilon) - h(\mathbf{u}^*) + L\epsilon \\ &= \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_0}\left[\ell_0(\phi(\mathbf{x}), y)\right] + h(\mathbf{u}^* + L\epsilon) + L\epsilon < \infty \end{aligned} \tag{4.36}$$

Each element of the sum has a finite value and therefore the sum is also finite: $h(\mathbf{u}^* + L\epsilon)$ is finite because $\mathbf{u} + L\epsilon \in \mathcal{U}$ (see assumption 6) and $\mathbb{E}\left[\ell_0(\phi(\mathbf{x}), y)\right] < B$. Therefore, the assumption that $\theta^*$ and $\mathbf{u}^*$ are infeasible is contradicted. ■

We have presented a parametrized version of the original resilient problem (P"-RES) which find a feasible set of parameters $\theta^*$ and $\mathbf{u}^*$. More than that, the optimal value of the parametrized problem is close to that of the resilient problem. However, the parameterized problem requires the computation of expectations over distributions $\mathcal{D}_i$. These distributions are often unknown and difficult to estimate. In the next section, we propose a problem that uses empirical data to estimate the expectations.

## 4.4 Resilient Empirical Risk Minimization

In the previous section, we introduced a parameterized problem (4.23) which approximates the resilient problem (P"-RES). Solving problem (4.23) requires computing expectations over distributions $\mathcal{D}$, however, these distributions are not known. Instead, we have a set of sample pairs $(\mathbf{x}, y)$ randomly selected from the distribution $\mathcal{D}$. Therefor, we can formulate

the parameterized empirical loss resilient minimization problem

$$\begin{aligned} P_N^* \;=\; \min_{\theta\in\Theta,\mathbf{u}\in\mathcal{U}} \quad & \frac{1}{N}\sum_{n=1}^{N}\ell_0(f_\theta(\mathbf{x}_n),y_n)+h(\mathbf{u}) \\ \text{s.t.} \quad & \frac{1}{N}\sum_{n=1}^{N}\ell_i(f_\theta(\mathbf{x}_n),y_n)\le u_i, \qquad i=1,\dots,m, \\ & \ell_i(f_\theta(\mathbf{x}_n),y_n)\le u_{in}, \qquad i=m+1,\dots,q,\ n=1\dots N. \end{aligned} \tag{P-ERM}$$

This problem is not guaranteed to be convex because of the parametrization, however, we will solve its dual problem and show that it is close to the primal resilient problem (P"-RES). To this end we formulate the empirical Lagrangian

$$\begin{aligned} \mathcal{L}_N(\theta,\boldsymbol{\lambda};\mathbf{u}) = h(\mathbf{u}) + \sum_{n=1}^{N}\Bigg[ & \frac{1}{N}\ell_0(f_\theta(\mathbf{x}_n),y_n) \\ & + \sum_{i=0}^{m}\boldsymbol{\lambda}_i\left(\frac{1}{N}\ell_i(f_\theta(\mathbf{x}_n),y_n)-u_i\right) \\ & + \sum_{i=m+1}^{q}\boldsymbol{\lambda}_{in}(\ell_i(f_\theta(\mathbf{x}_n),y_n)-u_{in})\Bigg]. \end{aligned} \tag{4.37}$$

From the Lagrangian we can formulate the dual

$$D_{\epsilon,N} = \max_{\boldsymbol{\lambda}\ge 0}\ \min_{\theta\in\Theta,\mathbf{u}in\mathcal{U}} \mathcal{L}_N(\theta,\boldsymbol{\lambda};\mathbf{u}). \tag{$D_{\epsilon,N}$}$$

Solving problem $(D_{\epsilon,N})$ gives us a solution that is close to the solution of the parameterized problem.

**Proposition 8.** *Let $D_\epsilon$ be the solution of the dual parametrized problem* $(\mathrm{DI}_\epsilon)$*, $D_{\epsilon,N}$ be the solution of the dual parametrized empirical problem $(D_{\epsilon,N})$ and $d_{VC}$ the VC dimension of the hypothesis class $\mathcal{P}$, then with probability $1-\gamma$ it holds that*

$$|D_\epsilon - D_{\epsilon,N}| \le \xi(N), \tag{4.38}$$

*where $\xi(N)$ is as in (Chamon and Ribeiro, 2020, Thm. 1).*

*Proof.* Let $\theta^*$, $\boldsymbol{\lambda}^*$, and $\mathbf{u}^*$ be the optimal variables to the parametrized dual problem ($\mathrm{DI}_\epsilon$) and $\hat{\theta}^*$, $\hat{\boldsymbol{\lambda}}^*$, and $\hat{\mathbf{u}}^*$ be the optimal variables for the empirical dual problem ($D_{\epsilon,N}$), then from Chamon and Ribeiro (2020) it holds that

$$\begin{aligned}\left|D_\epsilon - \hat{D}_{\epsilon,N}\right| &= \left|\mathcal{L}_\epsilon(\theta^*, \boldsymbol{\lambda}^*; \mathbf{u}^*) - \mathcal{L}_{\epsilon,N}(\hat{\theta}^*, \hat{\boldsymbol{\lambda}}^*; \hat{\mathbf{u}}^*)\right| \\ &= \left|\mathbb{E}\left[\ell_0(f_{\theta^*}(\mathbf{x}), y)\right] + h(\mathbf{u}^*) - \frac{1}{N}\sum_{n=1}^{N} \ell_0(f_{\hat{\theta}^*}(\mathbf{x}_n), y_n) - h(\hat{\mathbf{u}}^*)\right|\end{aligned} \tag{4.39}$$

Since $\theta^*$, $\mathbf{u}^*$ , $\hat{\theta}^*$, and $\hat{\mathbf{u}}^*$ minimize their respective Lagrangian, it must be that

$$\begin{aligned}&\left|\mathcal{L}_\epsilon(\theta^*, \boldsymbol{\lambda}^*; \mathbf{u}^*) - \mathcal{L}_{\epsilon,N}(\hat{\theta}^*, \hat{\boldsymbol{\lambda}}^*; \hat{\mathbf{u}}^*)\right| \\ &\leq \max\{|\mathcal{L}_\epsilon(\theta^*, \hat{\boldsymbol{\lambda}}^*; \mathbf{u}^*) - \mathcal{L}_{\epsilon,N}(\theta^*, \boldsymbol{\lambda}^*; \mathbf{u}^*)|, |\mathcal{L}_\epsilon(\hat{\theta}^*, \boldsymbol{\lambda}^*; \hat{\mathbf{u}}^*) - \mathcal{L}_{\epsilon,N}(\hat{\theta}^*, \hat{\boldsymbol{\lambda}}^*; \hat{\mathbf{u}}^*)|\} \\ &= \max\{|\mathbb{E}\left[\ell_0(f_{\theta^*}(\mathbf{x}), y)\right] - \frac{1}{N}\sum_{n=1}^{N} \ell_0(f_{\theta^*}(\mathbf{x}_n), y_n)|, \\ &|\mathbb{E}\left[\ell_0(f_{\hat{\theta}^*}(\mathbf{x}), y)\right] - \frac{1}{N}\sum_{n=1}^{N} \ell_0(f_{\hat{\theta}^*}(\mathbf{x}_n), y_n)|\}.\end{aligned} \tag{4.40}$$

By applying the VC generalization bound from (Vapnik, 2013, Sec. 3.4) we obtain

$$\left|D_\epsilon - \hat{D}_{\epsilon,N}\right| \leq \xi(N), \tag{4.41}$$

with probability $1-\gamma$, for $\xi(N)$ as in (Chamon and Ribeiro, 2020, Thm. 1). This concludes the proof. ■

Finally, by combining proposition 7 and proposition 8, we can relate the dual of the empirical problem ($D_{\epsilon,N}$) to the statistical problem (P"-RES).

**Theorem 5.** *Let $P$ be the solution to problem* (P"-RES) *and $D_{\epsilon,N}$ be the solution to problem ($D_{\epsilon,N}$). Then given assumption 6, it holds with probability of $1-\gamma$ that*

$$|P - D_{\epsilon,N}| \leq h(\mathbf{u}^* + L\epsilon) - h(\mathbf{u}^*) + L\epsilon + \xi(N) \tag{4.42}$$

*Proof.* The proof is obtained by applying the triangle inequality to the results obtained in proposition (7) and (8). ■

We have shown that the optimal values of $(D_{\epsilon,N})$ and (P"-RES) are close, next we will prove that the corresponding primal variables are close.

**Theorem 6** (Primal bound)**.** *Let $\phi^*$ be the optimal primal variable of problem* (P-RSL) *and $\theta^*$ optimal primal variable of $(D_{\epsilon,N})$, then for any $\mathbf{x} \in \mathcal{X}$ under assumptions 1, 3, 6 it holds with probability of $1-\gamma$ that*

$$|\phi'(\mathbf{x}) - \theta^*(\mathbf{x})| \leq \frac{2}{\mu}\big[h(\mathbf{u}^* + L\epsilon) - h(\mathbf{u}^*) + L\epsilon + \xi(N)\big], \tag{4.43}$$

*where $\mathbf{u}^*$ is the optimal perturbation, $\mu$ is the strong convexity parameter of $\ell_0$ and $\xi(N)$ as in (Chamon and Ribeiro, 2020, Thm. 1).*

*Proof.* Given that $\theta$ is an $\epsilon$ approximation of $\phi$ it stands to reason that any $f_\theta \in \mathcal{F}$, then given that $\ell_0$ is strongly convex we have the following inequality

$$|f_{\theta^*}(\mathbf{x}) - \phi^*(\mathbf{x})| \leq \frac{2}{\mu}(D_{\epsilon,N} - P^*) \leq \frac{2}{\mu}\big[h(\mathbf{u}^* + L\epsilon) - h(\mathbf{u}^*) + L\epsilon + \xi(N)\big] \tag{4.44}$$

this concludes the proof. ■

## 4.5 Learning the Resilient Formulation

In the previous sections, we have argued that we should solve problem $(D_{\epsilon,N})$ because it is an unconstrained problem which approximates our original problem (P"-RES). Although, solving $(D_{\epsilon,N})$ is not trivial, a solution can be obtained by finding the saddle point $(\theta^*, \boldsymbol{\lambda}^*, \mathbf{u}^*) = \max_{\boldsymbol{\lambda}} \min_{\theta,\mathbf{u}} \mathcal{L}(\theta, \boldsymbol{\lambda}; \mathbf{u})$, using the Arrow-Hurwicz algorithm Arrow et al. (1958). The saddle point is obtained by alternating the minimization and the maximization of the Lagrangian with respect to the primal variables and dual variables respectively. The

optimization is done via gradient descent for the primal variables $\theta$ and $\mathbf{u}$

$$\begin{aligned}\theta(t+1) &= \theta(t) - \eta_\theta d\theta(t) \\ \mathbf{u}(t+1) &= \mathbf{u}(t) - \eta_{\mathbf{u}} d\mathbf{u}(t),\end{aligned} \tag{4.45}$$

and gradient ascent for the dual variables

$$\boldsymbol{\lambda}(t+1) = \boldsymbol{\lambda}(t) + \eta_{\boldsymbol{\lambda}} d\boldsymbol{\lambda}(t). \tag{4.46}$$

In order to obtain the gradients of the Lagrangian with respect to the primal problem, recall that the minimization of the Lagrangian can be separated into two parts that each depend on only one primal variable.

$$\mathcal{L}(\theta, \boldsymbol{\lambda}, \mathbf{u}) = \mathcal{L}_\theta(\theta, \boldsymbol{\lambda}) + \mathcal{L}_{\mathbf{u}}(\boldsymbol{\lambda}, \mathbf{u}) \tag{4.47}$$

where $\mathcal{L}_\phi(\phi, \boldsymbol{\lambda}) = \mathcal{L}(\phi, \boldsymbol{\lambda}; 0)$ and $\mathcal{L}_{\mathbf{u}}(\boldsymbol{\lambda}, \mathbf{u})$ is

$$\mathcal{L}_{\mathbf{u}}(\boldsymbol{\lambda}, \mathbf{u}) = h(\mathbf{u}) - \boldsymbol{\lambda}^\top \mathbf{u}. \tag{4.48}$$

The gradient with respect to $\mathbf{u}$ is obtained from (4.48)

$$d\mathbf{u} = \frac{\partial \mathcal{L}(\boldsymbol{\lambda}, \mathbf{u})}{\partial \mathbf{u}} = \nabla h(\mathbf{u}) - \boldsymbol{\lambda}. \tag{4.49}$$

Solving for the optimal $\theta$ is similar to solving a regularized empirical risk minimization problem. The gradient with respect to $\theta$ is obtained from $\mathcal{L}(\theta, \boldsymbol{\lambda}; 0)$

$$\begin{aligned}d\theta = \frac{1}{N} \sum_{n=1}^{N} \Bigg( &\nabla_\theta \ell_0(f_\theta(\mathbf{x}_n), y_n) + \sum_{i=1}^{m} \nabla_\theta \ell_i(f_\theta(\mathbf{x}_n), y_n) \\ &+ N \sum_{j=m+1}^{q} \boldsymbol{\lambda}_{jn} \nabla_\theta \ell_j(f_\theta(\mathbf{x}_n), y_n) \Bigg).\end{aligned} \tag{4.50}$$

**Algorithm 5** Algorithm

Initialize $\theta(0)$, $\boldsymbol{\lambda}(0)$, $\mathbf{u}(0)$, and $0 < \eta \ll 1$.
**for** $t = 1 \dots T$
Compute the gradient $d\theta$ as in (4.50)
Compute the gradient $d\mathbf{u}$

$$d\mathbf{u} = h(u) - \boldsymbol{\lambda} \quad d\mathbf{v} = h(v) - \boldsymbol{\mu} \tag{4.52}$$

Update primal variables $\theta$ and $\mathbf{u}$:
$\theta(t) = \theta(t-1) - \eta d\theta(t-1)$
$\mathbf{u}(t) = \mathbf{u}(t-1) - \eta_{\mathbf{u}} d\mathbf{u}(t-1)$
Update dual variable:
$\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}(t-1) + \eta_{\boldsymbol{\lambda}} \left[\ell(f_\theta(\mathbf{x}), y) - \mathbf{u}(t-1)\right]$
**end**

Gradient ascent is done by updating the dual variable $\boldsymbol{\lambda}$ with the super-gradient. The constraints evaluated at the optimal Lagrangian minimizers are supergradients of the corresponding Lagrange multipliers Boyd and Vandenberghe (2004).

$$d\boldsymbol{\lambda} = \begin{cases} d\boldsymbol{\lambda}_i = \frac{1}{N} \sum_{n=1}^{N} \ell_i(f_\theta(\mathbf{x}_n), y_n) - \mathbf{u}_i & i = 1 \dots m \\ d\boldsymbol{\lambda}_{i,n} = \ell_{i,n}(f_\theta(\mathbf{x}_n), y_n) - \mathbf{u}_{i,n} & i = m+1 \dots m+q. \end{cases} \tag{4.51}$$

In this work, we use the Arrow-Hurwicz algorithm, however, it can be replaced with any primal dual algorithms Zhu and Chan (2008); Korpelevich (1976), which have been shown to work well in practice.

## 4.6 Applications

In the previous sections, we have presented a method which is resilient to outliers and corrupted data. In this section, we will exemplify through numerical experiment on the CIFAR-10 dataset Krizhevsky et al. (2009). First, we will compare the performance of a ResNet18 trained traditionally with that of a resilient network. The traditional network solves the problem.

$$p^* = \min_{\theta} \gamma\rho(\theta) + \frac{1}{N} \sum_{n=1}^{N} \ell(f_\theta(\mathbf{x}_n), y_n), \tag{PIV}$$

where $\rho(\theta)$ is a function of the weights of the network only

$$\rho(\theta) = \sum_{\theta_i \in \mathcal{W}} \|\theta\|_2^2, \tag{4.53}$$

$\ell(f_\theta(\mathbf{x}_n), y_n)$ is the cross-entropy loss:

$$\ell(f_\theta(\mathbf{x}_n), y_n) = -y_n \log(f_\theta(\mathbf{x}_n)), \tag{4.54}$$

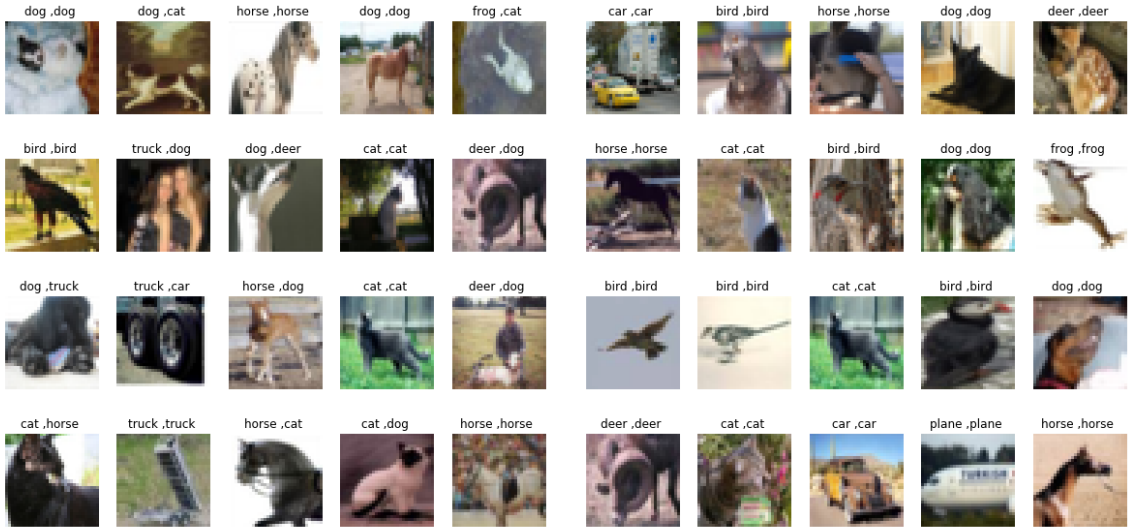
and $\gamma$ is the regularizing parameter, that mitigates the trade-off between minimizing the loss and avoiding overfitting. The resilient network solved the following problem:

$$\begin{aligned} p_r^* = \underset{\theta, \mathbf{u}}{\text{minimize}} \quad & \gamma\rho(\theta) + h(\mathbf{u}) \\ \text{subj. to} \quad & \ell(f_\theta(\mathbf{x}_n), y_n) \leq \mathbf{u}_n, \quad n = 1 \dots N, \end{aligned} \tag{PV}$$

where $h(\mathbf{u})$ is the cost function. Two different cost functions were considered the quadratic cost function, $h(\mathbf{u}) = \|\mathbf{u}\|_2^2$ and the Huber loss $h(\mathbf{u}, \delta) = \sum_{n=0}^{N} h(u_n, \delta)$

$$h(u, \delta) = \begin{cases} \frac{1}{2}u^2 & |u| \leq \delta \\ \delta(|u| - \frac{1}{2}\delta) & |u| > \delta. \end{cases} \tag{4.55}$$

The quadratic cost function penalizes losses as they increase, however, makes an increased effort to fit the function to those samples. By using the quadratic loss, we can identify the samples which are most difficult to learn from the optimal dual parameter $\boldsymbol{\lambda}$. The Huber loss allows for the loss of the samples that are more difficult to fit, to grow with linear penalty. This means that the function will fit well to the samples which are easier to fit and ignore outliers. The networks were each trained for 100 epochs with a learning rate $\eta_\theta = 0.01$. The resilient network used learning rates of $\eta_{\mathbf{u}} = 1$ and $\eta_{\boldsymbol{\lambda}} = 0.01$. We obtain a classification accuracy of 88.01% for the traditionally trained network, 90.76% for the resilient network with a quadratic cost function, and 89.93% for the resilient network with Huber loss cost function. Additionally, the resilient formulation gives us a score of how difficult a particular




(a) Quadratic cost function. (b) Huber loss cost function.

Figure 19: The images in the training set which are hardest to classify and their corresponding $\boldsymbol{\lambda}_n^*$.

constraint is to satisfy. Figure 19a shows the twenty images that were most difficult to classify using a quadratic cost and figure 19b shows the most difficult images to classify when using the Huber loss as a cost function. Given that these are color images it is not always obvious which aspect made these images more difficult, however, it is still possible to pick out some similarities between classes that lead to some of these pictures being missclassified. Wrong labels, multiple objects in one picture and ambiguities all make these images harder to classify.

### 4.6.1 Training in the Presence of Corrupted Data - Label flipping

In this section we create artificial outliers by randomly flipping the labels of a subset of our training sample. The performance of the resilient network was compared with a traditionally trained network, an oracle network and an iterative trimmed loss minimization (ITLM) Shen and Sanghavi (2019) on a clean dataset. The traditionally trained network trains a regularized resnet18 with weight decay 0.0005. The oracle networks trains only on the samples which have not been corrupted. ITML is a method for learning model parameters when a fraction of the training samples are corrupted. In order to learn the model parameters

the training set is trimmed by removing the samples which have the highest training losses.

The resilient network uses the Huber loss cost function with $\delta = 0.2$ and is trained over 120 epochs. The learning rates for the resilient network were set to $\eta_\theta = 0.001$, $\eta_{\boldsymbol{\lambda}} = 0.01$, and $\eta_\nu = 1$. The traditional and the oracle networks are trained for 80 epochs with a learning rate $\eta = 0.01$. For ITML, a proportion of $5\% + \alpha$ is trimmed, where $\alpha$ is the percent of corrupted training samples. The extra 5% is added because in practice the exact proportion of corrupted samples is unknown. ITLM is trained for 80 epochs with a learning rate of 0.01. All networks were trained uing a batch size of 256. Figure 20 shows a boxplot comparing the performance, over 8 repetitions, of each network as the percent of corrupted samples in the training set increases. The oracle network has a little drop in accuracy as the percentage of corrupted samples grows, due to the smaller training set. The traditionally trained network is affected by the corrupted samples and has a significant decrease in performance. The ITML network and the resilient networks have a similar performance. The ITML network suffers only a little drop in accuracy as the percentage of corrupted samples increases under the assumption that it trains on mostly clean samples. The resilient network adapts the slack of each constraint as it learns and can allow higher losses for corrupted samples. Therefore, it can maintain a better generalization and has a smaller drop inaccuracy.

Next, we create artificial outliers by flipping the labels of a subset of the training data in a systematic way. A derangement of the labels is used to reassign the classes of a percentage of the training set samples, such that an artificial outlier from class $A$ always get mislabeled to class $B$. This systematic approach is different from the random label flipping in that each class only has one other possible label.

The oracle network and the traditional network are trained as in the case of the random label flipping for 80 epochs with a learning rate of 0.01. The ITML network is trained with a trim interval of 10, a trim amount of $5+\alpha$, and a learning rate of 0.01 for 80 epochs. The resilient network is trained for 120 epochs with learning rates $\eta_{\boldsymbol{\lambda}} = 0.01$, $\eta_{nu} = 1$, and $\eta_\theta = 0.001$.

Figure 21 shows the performance of the traditionally trained, oracle, ITML, and resilient

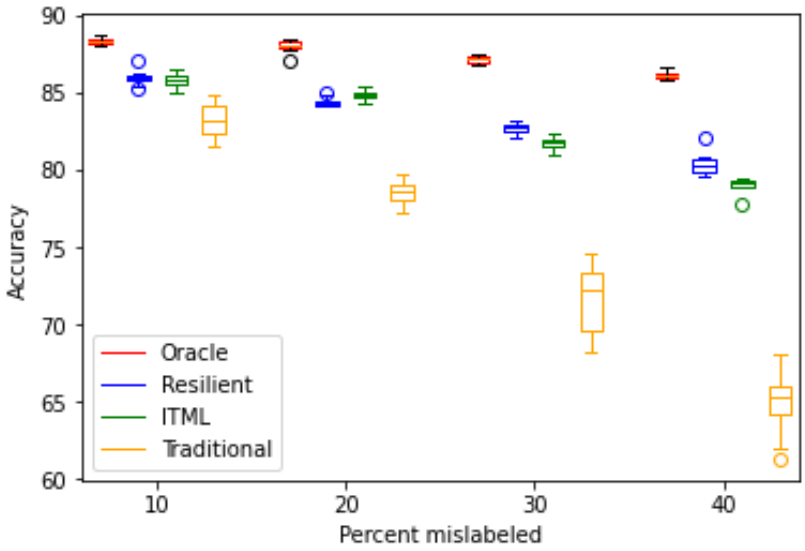



Figure 20: Performance of resilient network, traditional network, ITML, and oracle network are compared as a function of the number of labels flipped.

network as the percentage of systematically flipped labels increases. The oracle networks performs similarly as in the case of random flipped labels since it only trains on the clean labels. The traditional network and the ITML networks both have a significant drop in accuracy, with an average accuracy drop of 8.62% and 10.16% respectively as the percentage of corrupted samples increases from 10% to 40%. However, the performance of the traditionally trained network is architecture dependent. For example in Shen and Sanghavi (2019) the traditionally trained network drops to 62.03% while a resnet18 network dropped to an accuracy of 77.53%. The resilient network achieves a drop of only 3.24% when the percent of corrupted data points increases form 10% to 40%.

### 4.6.2 Training in the Presence of Outliers - Gaussian Blurring

In this section, samples are directly corrupted but are still assigned the same label. A Gaussian filter is used to blur a percentage of the training images. Two levels of blurring are used in order to observe the effect of the Gaussian blur radius as well as the percentage of blurry images. An example of the resulting images can be seen in figure 22.

The performances of an oracle network a traditionally trained network, the ITML network

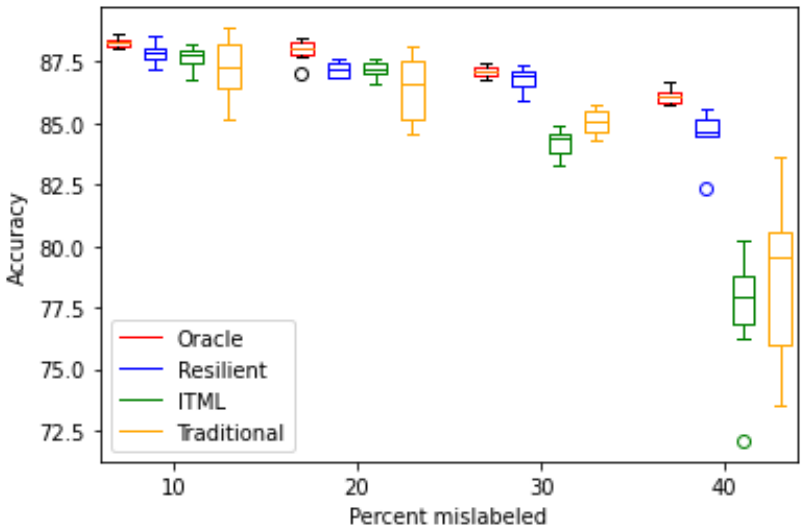



Figure 21: Performance of resilient network, traditional network, ITML, and oracle network are compared as a function of the number of labels flipped.

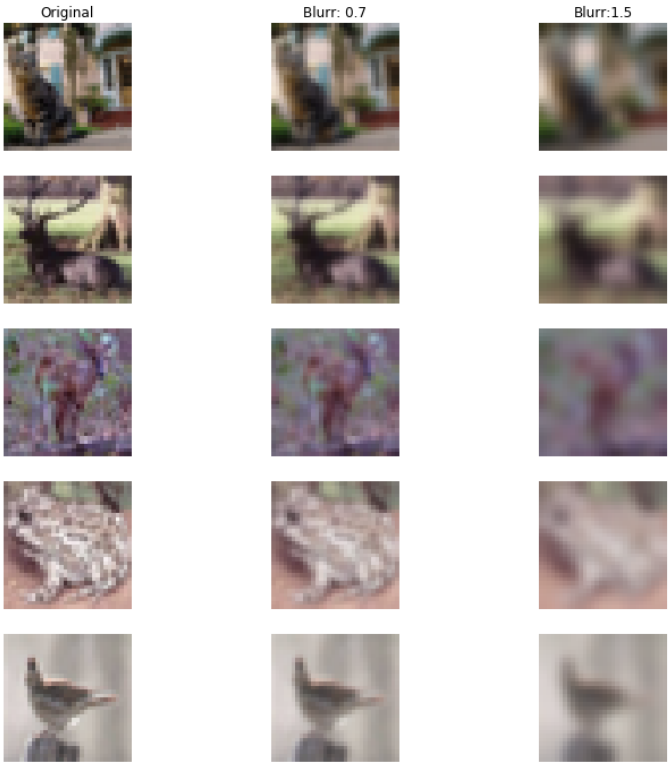



Figure 22: Examples of images with no blurring and Gaussian blurring with radius 0.7 and 1.5.

and the resilient network are compared as the number of blurry images grows. The networks are trained over 50 epochs with a batch size of 128. The learning rate is 0.01 for the oracle, traditional and ITML networks and 0.001 for the resilient network. Additionally, ITML uses a trim interval of 10 epochs and trims $5\% + \alpha$, where $\alpha$ is the true percentage of blurred images. The resilient network uses a Huber loss cost function with $\delta = 0.5$ and learning rates $\eta_{\boldsymbol{\lambda}} = 0.1$

In the problem of training with corrupted data, there is a trad-off between the sample size and the quality of the samples. Training on a larger sample size often improves the performance Perez and Wang (2017); Shorten and Khoshgoftaar (2019), however, if the samples are corrupted the resulting machine learning method might not generalize well. The four networks compared each have a different approach to this problem. The oracle network maximizes the quality of the samples and the traditionally trained network maximizes the sample size. The resilient network finds a compromise by only allowing certain losses to grow, while training on all samples. The trim loss achieves a compromise by recurrently changing the training samples based on the training losses.

Figure 23 shows the change in performance when a Gaussian blur with radius 0.7% is used. The oracle network has a higher accuracy than the other networks when only 10% of images are blurred, however, as the percentage of corrupted images increases the traditional and the resilient networks have a higher testing accuracy. The corrupted images still contain a good amount of information which the oracle network misses. The trim network has a lower accuracy because it trains on a smaller sample set that is not guaranteed to only have clean samples.

Figure 24 shows the change in performance when a Gaussian blur with radius 1.5 was applied to a fraction of the training images. The information in each blurred image is less useful than in the previous example and therefore we can see that the oracle network performs better than the traditional network. The resilient network is able to find a compromise between sample size and clean samples and has a better performance than the traditional method.

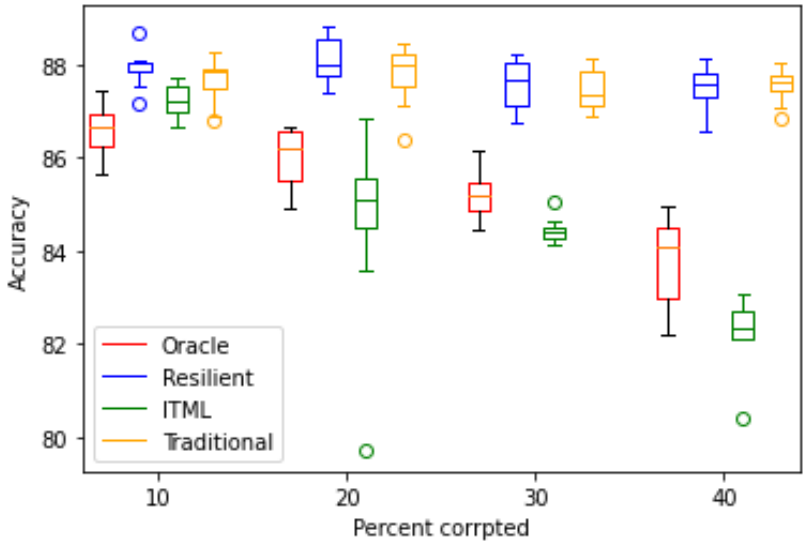



Figure 23: Performance of resilient network, traditional network, ITML, and oracle network on datasets with blurred images with a Gaussian blur with radius 0.7.

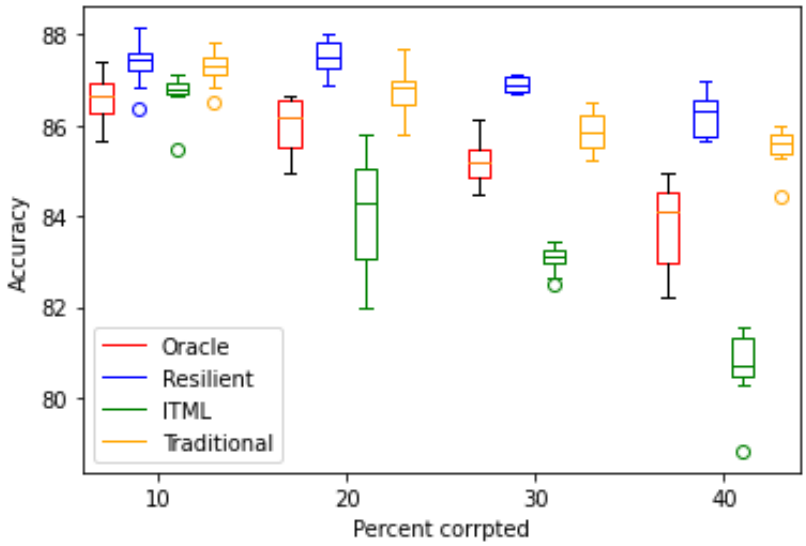



Figure 24: Performance of resilient network, traditional network, ITML, and oracle network on datasets with blurred images with a Gaussian blur with radius 1.5.

Similarly to the previous example, the trim network has a poor generalization accuracy.

The resilient networks generalizes better than the traditional network and is able to find the best compromise for the trade-off between sample size and clean data. Unlike the ITML network, it does not require the number of corrupted samples to be known, instead it allows

samples to have a higher loss based on the difficulty of fitting that sample.

# CHAPTER 5

# Conclusions

The focus of this dissertation was to develop the theory for balancing the fit and the complexity of learned representations. To achieve this goal, the problem is formulated as the minimization of the complexity measure subject to constraints that measure the fit. The constrained problem was shown to be solvable in the dual domain. Moreover, the dual optimal variables provide a measure of the difficulty of each constraint specification. This observation was applied to federated learning by sharing only a subset of samples that contribute to the solution. What's more, it is a useful tool for constraint specification.

The first part presented a method for finding parsimonious representations in RKHSs without compromising the fit by locally adapts the kernel centers and the kernel parameter. This was achieved by introducing a new integral representation of functions in RKHSs. The optimization problem uses a sparsity objective function to select both kernel centers and kernel parameters. The optimization problem is formulated as an SFP, which despite being infinite dimensional and non-convex can be solved via the dual problem. Moreover, from the dual, a novel representer theorem is established which holds for regularizers that promote sparsity. This technique of finding sparse representations has been used for federated classification. This was achieved by first introducing a method for traditional learning, which obtains both a sparse representation and identifies the critical samples for the classification problem. By leveraging the ability to detect the critical samples to the classification problem our federated learner can reduce traffic over the network and send less information. The federated classification method was shown to converge to a traditional learning method in which the learner has access to the entire data set as the sample size grows.

The second part tackled the challenges that arise from learning fit from individual constraints. It provides a framework for relaxing constraints based on resilient learning. We argued that

each constraint should be relaxed until the marginal effect on the primal objective is equal to the marginal cost of relaxing that constraint. Furthermore, we showed that the marginal effect was given by the optimal dual variable of the perturbed problem. Having shown that a solution exists for the statistical problem, we moved to a parametrized empirical problem that can be readily solved and bounded the difference between the solution of the resilient statistical problem and the resilient empirical problem.

APPENDIX

## A.1 Proof of Theorem 1

*Proof.* In order to show strong duality, it is sufficient to show that the perturbed function $P(\xi)$ in (A.1) is convex Rockafellar (2015); Shapiro and Scheinberg (2000). Consider the perturbed version of the optimization function

$$\begin{aligned} P(\xi) = \min_{\alpha,\boldsymbol{\theta}} \; & f_0(\alpha) \\ & s.t.\ c(z_i, y_i) \leq \xi_i \\ & z_i = \int \alpha(\mathbf{z}, w) \cdot k(\mathbf{x}_i, \mathbf{z}; w)\, dw\, d\mathbf{z}. \end{aligned} \tag{A.1}$$

In equation (A.1) $f_0(\alpha)$ represents the objective function of our original problem: $f_0(\alpha) = \gamma \int \mathbb{I}(\alpha(\mathbf{z}, w) \neq 0) + 0.5\alpha^2(w, \mathbf{x})\, dw\, d\mathbf{z}$. The optimal solution for $P(0)$ is the solution to the primal problem.

Convexity of the perturbed problem can be shown, by proving that given an arbitrary pair of perturbations $\xi_1$ and $\xi_2$ and the corresponding optimal values $P(\xi_1)$ and $P(\xi_2)$, for any $\beta \in [0, 1]$ the solution $P(\xi_\beta)$ has the following property, where $\xi_\beta$ is defined by $\xi_\beta = \beta\xi_1 + (1-\beta)\xi_2$:

$$P(\xi_\beta) \leq \beta P(\xi_1) + (1 - \beta)P(\xi_2). \tag{A.2}$$

In order to prove the convexity of the perturbed problem we need to introduce the following lemma.

**Lemma 5.** *The set of constraints given by*

$$\begin{aligned} \mathcal{B} = \{b,\, & b = f_0(\alpha),\ c(z_i, y_i) < \xi_i, \\ & z_i = \int \alpha(\mathbf{z}, w) \cdot k(\mathbf{x}_i, \mathbf{z}; w)\, dw\, d\mathbf{z}, \quad i = 1 \cdots N.\} \end{aligned} \tag{A.3}$$

*is convex.*

*Proof.* Given an arbitrary pair $b_1, b_2 \in \mathcal{B}$, there exists a corresponding $\alpha_1, \alpha_2 \in L_2$ such that $b_1 = f_0(\alpha_1)$ and $b_2 = f_0(\alpha_2)$. In order to prove the convexity of the set, we will show that there exists a feasible $\alpha_\beta \in L_2$ such that for any $\beta \in [0, 1]$

$$f_0(\alpha_\beta) = \beta\, b_1 + (1 - \beta) b_2 \tag{A.4}$$

Let $\mathbb{B}$ be the Borel field of all possible subsets of $\mathcal{U}$, where $\mathcal{U} = \{\mathcal{X} \times \mathcal{W}\}$ is the set of all possible kernel centers and kernel widths and. Let us construct a measure over $\mathbb{B}$, where $\mathcal{V} \subset \mathbb{B}$.

$$\boldsymbol{m}(\mathcal{V}) = \begin{bmatrix} \int_{\mathcal{V}} \alpha_1(\mathbf{v})\mathbf{k}(\mathbf{v})\, d\mathbf{v} \\ \int_{\mathcal{V}} \alpha_2(\mathbf{v})\mathbf{k}(\mathbf{v})\, d\mathbf{v} \\ \int_{\mathcal{V}} \gamma\mathbb{I}(\alpha_1(\mathbf{v}) \neq 0) + \alpha_1^2(\mathbf{v})\, d\mathbf{v} \\ \int_{\mathcal{V}} \gamma\mathbb{I}(\alpha_2(\mathbf{v}) \neq 0) + \alpha_2^2(\mathbf{v})\, d\mathbf{v} \end{bmatrix} \tag{A.5}$$

The first $2N$ elements of the measure represent the estimated function of the signal $\mathbf{y}$ using $\alpha_1$ and $\alpha_2$ and a subset of the kernels, where $\mathbf{k}(\mathbf{v})_i = k(\mathbf{X}, \mathbf{z}_v; w_v)$ and $\mathbf{v} = [\mathbf{z}_v^T, w_v]^T$. The last two elements of $\boldsymbol{m}(\mathcal{V})$ measure the sparsity of functions $\alpha_1$ and $\alpha_2$ over the set $\mathcal{V}$ respectively. Two sets are of interest, the empty set and $\mathcal{U}$. The measure of the former is $\boldsymbol{m}(\emptyset) = 0$ and the measure for $\mathcal{U}$ can be inferred from our optimization problem.

$$\boldsymbol{m}(\mathcal{U}) = \begin{bmatrix} \int_{\mathcal{U}} \alpha_1(\mathbf{v})\mathbf{k}(\mathbf{v})\, d\mathbf{v} \\ \int_{\mathcal{U}} \alpha_2(\mathbf{v})\mathbf{k}(\mathbf{v})\, d\mathbf{v} \\ \int_{\mathcal{U}} \gamma\mathbb{I}(\alpha_1(\mathbf{v}) \neq 0) + \frac{1}{2}\alpha_1^2(\mathbf{v})\, d\mathbf{v} \\ \int_{\mathcal{U}} \gamma\mathbb{I}(\alpha_2(\mathbf{v}) \neq 0) + \frac{1}{2}\alpha_2^2(\mathbf{v})\, d\mathbf{v} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ b_1 \\ b_2 \end{bmatrix} \tag{A.6}$$

Lyapunov's convexity theorem Liapounoff (1940) states that a non-atomic measure vector on a Borel field is convex. Note that the representation in (2.7) allows us to construct the measure with non-atomic masses and is essential to the proof of strong duality. Since $\alpha$ does not contain any point masses and we can choose the kernel $k(\cdot, \mathbf{z}, w)$ to not have any point masses, our measure $\boldsymbol{m}$ is convex . Therefore, for any $\beta \in [0, 1]$, there exists a set $\mathcal{V}_\beta \subset \mathbb{B}$ such that:

$$\boldsymbol{m}(\mathcal{V}_\beta) = \beta\, \boldsymbol{m}(\mathcal{U}) + (1-\beta)\boldsymbol{m}(\emptyset) = \beta\, \boldsymbol{m}(\mathcal{U}) \tag{A.7}$$

The measure of the complement of the set $\mathcal{V}_\beta$, as defined by $\mathcal{V}_\beta^c \cup \mathcal{V}_\beta = \mathcal{U}$ and $\mathcal{V}_\beta^c \cap \mathcal{V}_\beta = \emptyset$, can be computed, due to the additivity property of measures:

$$\boldsymbol{m}(\mathcal{V}_\beta^c) = \boldsymbol{m}(\mathcal{U}) - \boldsymbol{m}(\mathcal{V}_\beta) = (1-\beta)\boldsymbol{m}(\mathcal{U}) = \boldsymbol{m}(\mathcal{V}_{(1-\beta)}). \tag{A.8}$$

We can define the function $\alpha_\beta$ from (A.7) and (A.8):

$$\alpha_\beta(\mathbf{v}) = \begin{cases} \alpha_1(\mathbf{v}) & \mathbf{v} \in \mathcal{V}_\beta \\ \alpha_2(\mathbf{v}) & \mathbf{v} \in \mathcal{V}_\beta^c \end{cases} \tag{A.9}$$

From this construction of $\alpha_\beta(\mathbf{v})$ it can be easily seen that $f_0(\alpha_\beta) = \beta f_0(\alpha_1) + (1-\beta) f_0(\alpha_2)$. Next we will show that $\alpha_\beta$ is feasible. Define $\hat{\mathbf{y}}_\beta$ as:

$$\begin{aligned} \hat{\mathbf{y}}_\beta &= \int_{\mathcal{U}} \alpha_\beta(\mathbf{v})\mathbf{k}(\mathbf{v})\, d\mathbf{v} = \\ &= \int_{\mathcal{V}_\beta} \alpha_1(\mathbf{v})\mathbf{k}(\mathbf{v})\, d\mathbf{v} + \int_{\mathcal{V}_\beta^c} \alpha_2(\mathbf{v})\mathbf{k}(\mathbf{v})\, d\mathbf{v} = \\ &= \beta\hat{\mathbf{y}}_1 + (1-\beta)\hat{\mathbf{y}}_2 \end{aligned} \tag{A.10}$$

Since $c(z_i, y_i)$ is convex it follows that for any $i \in [1, N]$, $c(\beta z_{i,1}+(1-\beta)z_{i,2}, y_i) \leq \beta c(z_{i,1}, y_i)+(1-\beta)c(z_{i,2}, y_i)$. We can use this property to show that $c(z_{i,\beta}, y_i) \leq \xi_\beta$.

$$\begin{aligned} c(z_{i,\beta}, y_i) \leq \beta c(z_{i,1}, y_i) + (1-\beta)c(z_{i,2}, y_i) \leq \\ \leq \beta\xi_1 + (1-\beta)\xi_2 = \xi_\beta \end{aligned} \tag{A.11}$$

Thus it was proven that $\alpha_\beta$ is also feasible and therefore the set of constraints is convex. ■

Let $(\alpha_1, \mathbf{z}_1, \xi_1)$ and $(\alpha_2, \mathbf{z}_2, \xi_2)$ be the pair of optimal solutions to the two perturbed problems $P(\xi_1)$ and $P(\xi_2)$. We have shown that there exists a feasible point $\alpha_\beta$ for the problem perturbed by $\xi_\beta = \beta\xi + (1-\beta)\xi'$, which satisfies $f_0(\alpha_\beta) = P(\xi_1) + (1-\beta)P(\xi_2)$. Given that it is a feasible point the objective function is greater or equal to the solution of the problem

$$\beta P(\xi_1) + (1-\beta)P(\xi_2) = f_0(\alpha_\beta) \geq P(\beta\xi_1 + (1-\beta)\xi_2). \tag{A.12}$$

Since the perturbed problem is convex, the original problem has zero duality gap.

■

## A.2 Proof of Corollary 1

*Proof.* Theorem 1 implies that any solution $(\alpha^\star, \hat{\boldsymbol{y}}^\star)$ of (PII) is such that Boyd and Vandenberghe (2004)

$$(\alpha^\star, \hat{\boldsymbol{y}}^\star) \in \underset{\alpha,\, \hat{y}_i}{\operatorname{argmin}} \mathcal{L}(\alpha, \hat{\boldsymbol{y}}, \boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star). \tag{A.13}$$

Since $\mathcal{L}$ in (2.10) separates across $\alpha$ and $\hat{\boldsymbol{y}}$, we can consider the minimizations individually to obtain

$$\alpha^\star \in \underset{\alpha \in L_2}{\operatorname{argmin}} \mathcal{L}_\alpha(\alpha, \boldsymbol{\lambda}^\star), \tag{A.14}$$

for $\mathcal{L}_\alpha$ from (2.14). We also know from Proposition 2 that $\alpha_d^\star$ is in the argmin set of (A.14). In the sequel, we show that it is (essentially) its only element.

To do so, we construct $\alpha^\star$ piece by piece by partitioning the integral in $\mathcal{L}_\alpha$ into three disjoint sets depending on the value of $\alpha_d^\star$. On $\mathcal{A}_> = \{(\mathbf{z}, w) \in \mathcal{X} \times \mathcal{W} \mid |\alpha_d^\star(\mathbf{z}, w)| > \sqrt{2\gamma}\}$, we know that $\alpha_d^\star$ takes values from (2.19), the unique minimizer of $\mathcal{L}_\alpha$ since it stems from the minimization of the strongly convex function (2.18). Moreover, our assumptions on the reproducing kernel together with (2.19) imply that $\alpha_d^\star \in L_2$ when restricted to $\mathcal{A}_>$. Hence,

$$\alpha^\star(\mathbf{z}, w) = \alpha_d^\star(\mathbf{z}, w), \quad \text{for } (\mathbf{z}, w) \in \mathcal{A}_>. \tag{A.15}$$

Over the set $\mathcal{A}_< = \{(\mathbf{z}, w) \in \mathcal{X} \times \mathcal{W} \mid |\alpha_d^\star(\mathbf{z}, w)| < \sqrt{2\gamma}\}$, notice from (2.14) that the integrand of $\mathcal{L}_\alpha$ is always non-negative. What is more, it is always strictly positive unless $\alpha \equiv 0$. This is ready by applying Lemma (1) and the fact that the minimum of (2.18) is positive. Thus, from the monotonicity of the integral operator, $\alpha^\star$ is again unique and equal to zero on $\mathcal{A}_<$. From (2.15), so is $\alpha_d^\star$ and we obtain

$$\alpha^\star(\mathbf{z}, w) = \alpha_d^\star(\mathbf{z}, w) = 0, \quad \text{for } (\mathbf{z}, w) \in \mathcal{A}_<. \tag{A.16}$$

Immediately, we have that $\alpha^\star \in L_2$ over $\mathcal{A}_> \cup \mathcal{A}_<$.

To conclude the proof, observe that $\mathfrak{m}\left[\mathcal{A}_> \cup \mathcal{A}_<\right] = \mathfrak{m}\left[\mathcal{X} \times \mathcal{W}\right]$, where $\mathfrak{m}$ denotes the Lebesgue measure. Indeed, the complement of $\mathcal{A}_> \cup \mathcal{A}_<$ is the set $\mathcal{A}_= = \{(\mathbf{z}, w) \in \mathcal{X} \times \mathcal{W} \mid |\alpha_d^\star(\mathbf{z}, w)| = \sqrt{2\gamma}\}$. From our assumption on the reproducing kernel, $\mathcal{A}_=$ is the set of zeros of a real analytic function, which are isolated and therefore countable Krantz and Parks (2002). In other words, $\alpha^\star$ and $\alpha_d^\star$ are in the same equivalence class in $L_2$ since they are equal except perhaps on a set of measure zero. ■

## A.3 Proof of Lemma 3

*Proof.* For this proof we will establish the following notation in order to make the proof easier to read:

$$k_n = k(x_n, s; w); \tag{A.17}$$

The first term of dual function can be rewritten as:

$$\begin{aligned}\boldsymbol{\lambda}^\top \mathbf{Q}\boldsymbol{\lambda} &= \frac{1}{N^2}\int \sum_n \sum_m \lambda_n \lambda_m k_n k_m y_n y_m d\mathbf{s}\, dw \\ &= \frac{1}{N^2}\int \left(\lambda_n y_n k_n + \sum_{m\neq n} \lambda_m y_m k_m\right)^2 ds\, dw \\ &= \boldsymbol{\lambda'}^\top \mathbf{Q'}\boldsymbol{\lambda'} + \frac{1}{N^2}\left(2\lambda_n y_n k_n \sum_{m\neq n}\lambda_m y_m k_m + \lambda_n^2 k_n^2\right)\end{aligned} \tag{A.18}$$

where $\boldsymbol{\lambda'}$ and $\mathbf{Q'}$ are the variables $\boldsymbol{\lambda}$ without the $n^{th}$ element and $\mathbf{Q}$ without the $n^{th}$ row and column respectively. Using (A.18) we can rewrite the dual function:

$$\begin{aligned}g(\boldsymbol{\lambda}) = -0.5\boldsymbol{\lambda'}^\top \mathbf{Q'}\boldsymbol{\lambda'} + \frac{1}{N}\boldsymbol{\lambda'}^\top(1-\epsilon) + \gamma m(\mathcal{X},\mathcal{W}) \\ +\frac{1}{N}\lambda_n\left(1-\epsilon - y_n\frac{1}{N}\int\sum_{m\neq n}\lambda_m y_m k_m k_n ds\, dw\right) \\ -\frac{0.5}{N^2}\int \lambda_n^2 k_n^2 ds\, dw\end{aligned} \tag{A.19}$$

Notice that $(1/N)\int\sum_{i\neq n}\lambda_i k_i k_n d\mathbf{s} dw$ evaluated at the optimal $\boldsymbol{\lambda'}$ is precisely $\hat{y}_n$ considering $\boldsymbol{\lambda}'_m = \boldsymbol{\lambda}_m(N-1)/N)$ for all $m$ and $g(\boldsymbol{\lambda}|\boldsymbol{\lambda}_n = 0) = g((N/(N-1)\boldsymbol{\lambda'})$. Since, $\boldsymbol{\lambda}_n = 0$ it follows from complementary slackness that $1-\epsilon-y_n\hat{y}_n < 0$. Therefore, it follows that the optimal values for the two dual functions are equal if $\lambda_n^* = 0$. Moreover, the optimal primal variables are equal, i.e., $\alpha^*(\mathbf{s},w) = \alpha'^*(\mathbf{s},w)$. This concludes the first part of the proof.

Next, we will show that if a model that is optimal for $X'$ and that has the property $1-\epsilon-y_n\hat{y}_n < 0$ for a new sample $\mathbf{x}_n$, the optimal dual variable corresponding to that point for the model trained on the set $X = X'\cup\{x_n\}$ has value $\boldsymbol{\lambda}_n^* = 0$. Equation (A.19) implies that optimizing for the variable $\boldsymbol{\lambda}_n$, given the solution to the model using $X'$ results in $\lambda_n = 0$. This value maximizes the dual function $g(\boldsymbol{\lambda})$. It is necessary to prove that there is not a value for $\boldsymbol{\lambda}$ different from $\boldsymbol{\lambda'}^*$ for which $g(\boldsymbol{\lambda'}^*) < g(\boldsymbol{\lambda}^*)$. Since $1-\epsilon-y_n\hat{y}_n < 0$, the optimal $\alpha'^*$ is feasible for the model which uses $x_n$ as a sample as well and it has not been proven yet

to be optimal for the full set we can say $P'^* \geq P^*$. However, since we have strong duality it is also true that

$$g(\boldsymbol{\lambda}'^*) = P'^* \geq P^* = g(\boldsymbol{\lambda}^*) \tag{A.20}$$

Since $g(\boldsymbol{\lambda}^*)$ is the maximum over $\boldsymbol{\lambda}$ it follows that $g(\boldsymbol{\lambda}'^*) = g(\boldsymbol{\lambda}^*)$, which implies that $\lambda_n = 0$. This concludes the proof.

■

## A.4 Proof of Theorem 2

*Proof.* Given two data sets $\mathbf{X}_i$ and $\mathbf{X}_j$ drawn over partitions of the space $\mathcal{X} = \mathcal{X}_i \cup \mathcal{X}_j$, let $\alpha^*(\mathbf{s}, w)$ be the solution to the problem (PC) given $[\mathbf{X}_i, \mathbf{X}_j]$ as a training set and, $\alpha^*_{(i)}(\mathbf{s}, w)$ and $\alpha^*_{(j)}(\mathbf{s}, w)$ be the solution to the problem (Pi) trained on $\mathbf{X}_i$ and $\mathbf{X}_j$ respectively. Additionally, let the overlap be large enough that Hypothesis 3 holds for the non-overlapping spaces. Let $\mathcal{X}_o = \mathcal{X}_i \cap \mathcal{X}_j$ be the overlapping space and $\mathcal{X}'_i = \mathcal{X}_i \setminus \mathcal{X}_o$, $\mathcal{X}'_j = \mathcal{X}_j \setminus \mathcal{X}_o$. Then we can write the $\alpha(\mathbf{s}, w)$ as a sum of functions which are nonzero only over one space where $\alpha_i(\mathbf{s}, w) = 0$ for all $\mathbf{s} \notin \mathcal{X}'_i$, $\alpha_j(\mathbf{s}, w) = 0$ for all $\mathbf{s} \notin \mathcal{X}'_j$ and $\alpha_o(\mathbf{s}, w) = 0$ for all $\mathbf{s} \notin \mathcal{X}_o$

$$\alpha(\mathbf{s}, w) = \alpha_i(\mathbf{s}, w) + \alpha_j(\mathbf{s}, w) + \alpha_o(\mathbf{s}, w) \tag{A.21}$$

it follows from Theorem 4 that the each $\alpha_{(j),o}(\mathbf{s}, w)$ and $\alpha_{(i)o}$ converge to each other as the number of samples grows

$$|\alpha^*_{(j)o} - \alpha^*_{(i)o}| \leq 2\left(2\sqrt{\frac{M}{\mu N^{1.5}}} + \frac{c\rho}{\sigma^3\sqrt{N}}\right) \tag{A.22}$$

As the sample size grows the functions $f_i$ and $f_j$ over the overlapping space $\mathcal{X}_o$ converge and therefore if for a point $\mathbf{x} \in \mathcal{X}_o$, if $1 - \epsilon_{\mathbf{x}} - yf_i(\mathbf{x}) < 0$ then it must also hold that $1 - \epsilon_{\mathbf{x}} - yf_j(\mathbf{x}) < 0$. Because the agents sample over the overlapping area, as the sample size grows and the agents agree on the critical samples, they will also agree with the centralized

learner on the critical samples. Then according to Lemma 3 the solution of (PF) and (PC) will converge over $\mathcal{X}_o$. Although this was illustrated for two agents, the proof holds for any number of agents.

Over the spaces which do not overlap, consider (Pi) trained on $\mathbf{X}_i$ and (PC) trained on $\mathbf{X}$ and let $\alpha_i^*$ and $\alpha_{(i)i}^*$ be their respective optimal values. Then for $\mathbf{s} \in \mathcal{X}_i'$ we can establish the following

$$\begin{aligned} &|\alpha_i(s,w)^* - \alpha_{(i)i}^*(s,w)| \leq \\ &\left|\alpha^*(\mathbf{s},w) - \sum_i \alpha_{(i)}^*(\mathbf{s},w)\right| + |\sum_{j\neq i}\alpha_{(j)}(s,w)| \\ &\leq \frac{2\sqrt{2\xi mL}}{N\sqrt{\mu N}} + \sum_{j\neq i}|\alpha_{(j)}(s,w)| \\ &\leq \frac{2\sqrt{2\xi mL}}{N\sqrt{\mu N}} + \xi\sum_{j\neq i}\frac{\|\boldsymbol{\lambda}_j\|_1}{N_j}. \end{aligned} \tag{A.23}$$

Notice that $\alpha_j(\mathbf{s}, w)$ has little effect on the value of $f(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}_i$. As the sample size grows, if $1 - \epsilon_{\mathbf{x}} y f_i(\mathbf{x}) < 0$ then it must also hold that $1 - \epsilon_{\mathbf{x}} y f(\mathbf{x}) < 0$, where $f_i$ and $f$ are the solutions found by agent i and the centralized learner respectively. Then according to Lemma 3 as the sample size grows the agents and the centralized learner agree on the critical samples and the federated learner (PF) and the centralized learner (PC) solve more similar problems. ■

## A.5 Proof of Lemma 3

*Proof.* We first show that it is true for two agents and then expand it for multiple agents. Let $\boldsymbol{\lambda}_1$ be the dual Recall the dual function (3.6) is a quadratic function

$$g(\boldsymbol{\lambda}) = -0.5\boldsymbol{\lambda}^\top \mathbf{Q}\boldsymbol{\lambda} + \frac{1}{N}\boldsymbol{\lambda}^\top(\mathbf{1} - \boldsymbol{\epsilon}) + m(\mathcal{X}, \mathcal{W}), \tag{A.24}$$

with

$$\mathbf{Q}_{nm} = \frac{y_n y_m}{N^2}\int_{\mathcal{X}\times\mathcal{W}} k(\mathbf{x}_n, \mathbf{s}; w)k(\mathbf{x}_m, \mathbf{s}; w)d\mathbf{s}\, dw \tag{A.25}$$

The matrix $\mathbf{Q}$ can be divided into sub-matrices based on the agents, to which the kernels centers belong:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}, \tag{A.26}$$

for which

$$\begin{aligned} \mathbf{Q}_{ij(nm)} &= \frac{1}{N^2} \int_{\mathcal{X}\times\mathcal{W}} k(\mathbf{x}_n^{(i)}, \mathbf{s}; w) k(\mathbf{x}_m^{(j)}, \mathbf{s}; w) d\mathbf{s}\, dw, \\ & \mathbf{x}_n^{(i)} \in \mathbf{X}_i. \end{aligned} \tag{A.27}$$

Then notice that

$$\begin{aligned} \boldsymbol{\lambda}^\top \mathbf{Q} \boldsymbol{\lambda} &= \left(\frac{N}{N_1}\boldsymbol{\lambda}_1^\top\right) \mathbf{Q}_{11} \left(\frac{N}{N_1}\boldsymbol{\lambda}_1\right) + \left(\frac{N}{N_2}\boldsymbol{\lambda}_2^\top\right) \mathbf{Q}_{22} \left(\frac{N}{N_2}\boldsymbol{\lambda}_2\right) \\ &\quad + 2\left(\frac{N}{N_1}\boldsymbol{\lambda}_1^\top\right) \mathbf{Q}_{12} \left(\frac{N}{N_2}\boldsymbol{\lambda}_2\right) \\ &= \boldsymbol{\lambda}_1^\top \mathbf{Q}_1 \boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2^\top \mathbf{Q}_2 \boldsymbol{\lambda}_2 + 2\left(\frac{N}{N_1}\boldsymbol{\lambda}_1^\top\right) \mathbf{Q}_{12} \left(\frac{N}{N_2}\boldsymbol{\lambda}_2\right) \end{aligned} \tag{A.28}$$

Additionally, the measure of the support of the dual function is equal to the sum of the measures of the individual agents

$$m(\mathcal{X}, \mathcal{W}) = m(\mathcal{X}_1, \mathcal{W}) + m(\mathcal{X}_2, \mathcal{W}). \tag{A.29}$$

Then we can conclude the following

$$\begin{aligned} & |g(\boldsymbol{\lambda}) - (g_1(\boldsymbol{\lambda}_1) + g_2(\boldsymbol{\lambda}_2))| \\ &= \left| 2\left(\frac{N}{N_1}\boldsymbol{\lambda}_1^\top\right) \mathbf{Q}_{12} \left(\frac{N}{N_2}\boldsymbol{\lambda}_2\right) \right| \\ &= \frac{2}{N_1 N_2} \boldsymbol{\lambda}_1^\top \int k(\mathbf{X}_1, \mathbf{s}; w) k(\mathbf{X}_2, \mathbf{s}; w) d\mathbf{s} dw \boldsymbol{\lambda}_2 \\ &\leq \frac{2\xi m(\mathcal{X}, \mathcal{W})}{N_1 N_2} \boldsymbol{\lambda}_1^\top \mathbf{J} \boldsymbol{\lambda}_2 \end{aligned} \tag{A.30}$$

The last inequality stems from (3.16) and the fact that a value of a kernel is at most 1. This result can be extended to multiple agents by considering all pairs of $\mathbf{Q}_{ij}$ in the difference

between the global dual function and the local dual functions. Therefore we obtain:

$$\left| g(\boldsymbol{\lambda}) - \sum_i g_i(\boldsymbol{\lambda}_i) \right| \leq \frac{2\xi m L}{N^2} \tag{A.31}$$

for $L = (N^2/(N_i N_j))\boldsymbol{\lambda}_i^\top \mathbf{J} \boldsymbol{\lambda}_j$. ■

## A.6 Proof of Theorem 3

*Proof.* Recall the relationship between the dual functions of the two problems (3.17). The relationship between the dual functions optimal values can be obtained through triangle inequality

$$|g(\boldsymbol{\lambda}^*) - \sum_i g_i(\boldsymbol{\lambda}_i^*)| \leq \frac{4\xi m L}{N^2}, \tag{A.32}$$

The dual function is strongly concave near the optimal value such that we can establish the relationship between the dual optimal variables

$$\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_a\|^2 \leq \frac{2}{\mu} |g(\boldsymbol{\lambda}^*) - g(\boldsymbol{\lambda}_a^*)| - \frac{2}{\mu} \nabla g(\boldsymbol{\lambda}^*)(\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_a^*), \tag{A.33}$$

where $\boldsymbol{\lambda}_a^* = [(N_1/N)\boldsymbol{\lambda}_1^\top, \ldots, (N_K/N)\boldsymbol{\lambda}_K^*]$. The gradient at the optimal value $\nabla g(\boldsymbol{\lambda}^*) = 0$ or $\boldsymbol{\lambda}^* = 0$. Therefore, the equation can be reduced to

$$\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_a\|^2 \leq \frac{2}{\mu} |g(\boldsymbol{\lambda}^*) - g(\boldsymbol{\lambda}_a^*)| \leq \frac{8\xi m L}{\mu N^2}, \tag{A.34}$$

The optimal primal value can be obtained from the dual value and therefore we can establish the following inequality

$$\begin{aligned}
&\left|\alpha^*(\mathbf{s},w) - \sum_i \alpha_i^*(\mathbf{s},w)\right| = \\
&\left|\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\lambda}_n^* y_n k(\mathbf{x}_n, s; w) - \sum_i \frac{1}{N_i} \sum_{\mathbf{x}_n \in \mathbf{X}_i} \boldsymbol{\lambda}_{a,n}^* y_n k(\mathbf{x}, s; w)\right| \\
&\leq \sum_{n=1}^{N} \left| y_n k(\mathbf{x}, \mathbf{s}; w) \left(\frac{1}{N}\boldsymbol{\lambda}_n^* - \frac{1}{N_i}\boldsymbol{\lambda}_{a,n}^*\right)\right| \\
&\leq \sum_{n=1}^{N} \left|\frac{1}{N}\boldsymbol{\lambda}_n^* - \frac{1}{N_i}\boldsymbol{\lambda}_{a,n}^*\right| \leq \frac{1}{\sqrt{N}}\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_a^*\| \\
&\leq \frac{2\sqrt{2\xi m L}}{N\sqrt{\mu N}}
\end{aligned} \tag{A.35}$$

where $\alpha^*(\mathbf{s}, w)$ represents the optimal variable learned by the centralized learner (PC) and $\alpha_i^*(\mathbf{s}, w)$ represents the optimal variable learned by the agent (Pi). ■

## A.7 proof of theorem 4

*Proof.* In order to prove the theorem we first formulate the Lagrangian.

$$\mathcal{L}_i(\alpha, \boldsymbol{\lambda}) = \rho(\alpha) + \frac{1}{N_i}\sum_{x \in \mathcal{X}_i} \lambda(x)\left[\ell(f(x), y) - \epsilon(x)\right] \tag{A.36}$$

Similarly, we can construct a function $\mathcal{L}(f, \lambda)$ which is not associated with any primal function

$$\mathcal{L}(\alpha, \boldsymbol{\lambda}) = \rho(\alpha) + \int_{x \in \mathcal{X}} \lambda(x)\left[\ell(f(x), y) - \epsilon(x)\right] p(x) dx \tag{A.37}$$

Notice that the integral is precisely the expected value:

$$\mathbb{E}_x\left(\lambda(x)\left[\ell(f(x), y) - \epsilon(x)\right]\right)$$

Therefore the minimization of (A.36) can be viewed as an empirical risk minimization problem which approximates the statistical loss minimization problem in (A.37). From Lugosi

et al. (2004) and Cortes et al. (2009) it follows that:

$$|\mathcal{L}(\alpha, \lambda) - \mathcal{L}_i(\alpha, \boldsymbol{\lambda})| \leq \frac{M}{\sqrt{N_i}} \tag{A.38}$$

where $M$ is a constant, such that $\|f\|^2 \leq M$ and $N_i$ is the sample size. We can construct the dual function and a function based on $\mathcal{L}(\alpha, \lambda)$

$$g_i(\boldsymbol{\lambda}) = \min_{\alpha} \mathcal{L}(\alpha, \boldsymbol{\lambda}) \tag{A.39}$$

$$g(\lambda) = \min_{\alpha} \mathcal{L}(\alpha, \lambda) \tag{A.40}$$

For which we can compute the optimal $\lambda$

$$\boldsymbol{\lambda}_i^* = \underset{\boldsymbol{\lambda} \geq 0}{\operatorname{argmax}}\, g_i(\boldsymbol{\lambda}) \tag{A.41}$$

$$\lambda^* = \underset{\lambda \geq 0}{\operatorname{argmax}}\, g(\lambda). \tag{A.42}$$

Since the difference between $\mathcal{L}_i(\alpha, \boldsymbol{\lambda})$ and $\mathcal{L}(\alpha, \lambda)$ is bounded, so is there minimums,

$$|g(\lambda) - g_i(\boldsymbol{\lambda})| \leq \frac{M}{\sqrt{N_i}} \tag{A.43}$$

Because the inequality holds for any $\lambda$, it must hold for the optimal values. Let $\lambda_s^*$ be the optimal function $\lambda(\mathbf{x})$ evaluated at the sample points and $\boldsymbol{\lambda}_i^*$ be the optimal dual variable of (Pi), then by the triangle inequality it follows that:

$$|g(\lambda_s^*) - g_i(\boldsymbol{\lambda}_i^*)| \leq \frac{2M}{\sqrt{N_i}} \tag{A.44}$$

Furthermore, the dual function is strongly convex near the optimal value:

$$g_i(\boldsymbol{\lambda}_i^*) - g_i(\boldsymbol{\lambda}_s^*) \geq \frac{\mu}{2}\|\boldsymbol{\lambda}_i^* - \boldsymbol{\lambda}_s^*\|^2 + \nabla g_i(\boldsymbol{\lambda}_i^*)(\boldsymbol{\lambda}_i^* - \boldsymbol{\lambda}_s^*), \tag{A.45}$$

where $\boldsymbol{\lambda}_s^*$ is a vector for which $\boldsymbol{\lambda}_{s,n}^* = \lambda_s^*(\mathbf{x}_n)$. Notice that the term $\nabla g_i(\boldsymbol{\lambda}_i^*)(\boldsymbol{\lambda}_i^* - \lambda_s^*) \geq 0$. Most of the terms of the gradient are zero since $\boldsymbol{\lambda}_i$ maximizes the dual function. For the other terms, $\nabla g(\boldsymbol{\lambda}_i^*)_n < 0$ only if $\boldsymbol{\lambda}_{i,n}^* = 0$. In the latter case $(\boldsymbol{\lambda}_{i,n}^* - \lambda_{s,n}^*) \leq 0$. Then we can conclude

$$\|\boldsymbol{\lambda}_i^* - \lambda_s^*\|^2 \leq \frac{4M}{\mu\sqrt{N_i}}. \tag{A.46}$$

Next a bound on the functions $\alpha$ can be established which are defined as

$$\alpha_i^*(s,w) = \begin{cases} \frac{1}{N_i}\sum_j \lambda_j^* y_j k(\mathbf{x}_j, s; w), & |\alpha_i(s,w)| > \sqrt{2\gamma} \\ 0 & otherwise. \end{cases} \tag{A.47}$$

Similarly, a function $\alpha(s,w) = \underset{\alpha \in L_2}{\operatorname{argmin}} \mathcal{L}(\alpha, \lambda^*)$ can be be computes as:

$$\alpha(s,w) = \begin{cases} \int \lambda(x) y(x) k(x, s; w) p(x)\, dx, & \alpha^2(s,w) > 2\gamma \\ 0 & otherwise. \end{cases} \tag{A.48}$$

The difference between $\alpha$ and $\alpha_i$ is bounded as follows

$$\begin{aligned} &|\alpha(s,w) - \alpha_i(s,w)| = \\ &\left| \int \lambda(x) y_x k(x, s; w) p(x)\, dx - \frac{1}{N_i}\sum_j \lambda_j^* y_j k(\mathbf{x}_j, s; w) \right| \leq \\ &\frac{1}{N_i}\left| \sum_j (\boldsymbol{\lambda}_{i,j}^* - \lambda_{s,j}) y_j k(\mathbf{x}_j, s; w) \right| + \\ &\left| \int \lambda(x) y_x k(x, s; w) p(x)\, dx - \frac{1}{N_i}\sum_j \lambda_{s,j}^* y_j k(\mathbf{x}_j, s; w) \right| \leq \\ &\frac{1}{N_i}\sum_j |\boldsymbol{\lambda}_{i,j}^* - \lambda_{s,j}| + \frac{c\rho}{\sigma^3\sqrt{N_i}} \\ &\leq \frac{1}{\sqrt{N_i}}\|\boldsymbol{\lambda}_i^* - \lambda_s^*\|_2 + \frac{c\rho}{\sigma^3\sqrt{N_i}} \leq \frac{2\sqrt{M}}{\sqrt{N_i^{1.5}\mu}} + \frac{c\rho}{\sigma^3\sqrt{N_i}}, \end{aligned} \tag{A.49}$$

for which $c > 0$ is a constant, $\rho = \mathbb{E}_{\mathbf{x}}\left[|\lambda(\mathbf{x})y_{\mathbf{x}}k(\mathbf{x}, s; w)|^3\right]$ and $\sigma^2 = \mathbb{E}_{\mathbf{x}}\left[|\lambda(\mathbf{x})y_{\mathbf{x}}k(\mathbf{x}, s; w)|^2\right]$ Given two models trained on independently drawn data sets, with optimal variables $\alpha_1$ and $\alpha_2$ respectively, then the absolute difference between the two variables is

$$\begin{aligned} &|\alpha_1(s, w) - \alpha_2(s, w)| \\ &\leq \frac{2\sqrt{M}}{\sqrt{N_1^{1.5}\mu_1}} + \frac{c\rho}{\sigma^3\sqrt{N_1}} + \frac{2\sqrt{M}}{\sqrt{N_2^{1.5}\mu_2}} + \frac{c\rho}{\sigma^3\sqrt{N_2}} \\ &\leq 2\left(2\sqrt{\frac{M}{\mu N^{1.5}}} + \frac{c\rho}{\sigma^3\sqrt{N}}\right), \end{aligned} \tag{A.50}$$

where $N = min(N_i, N_j)$. ■

# BIBLIOGRAPHY


J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pages 22–31. PMLR, 2017.

A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Comput. Sci.*, 209(1-2):237–260, 1998.

D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Esann*, 2013.

J. Arenas-Garcia, K. Petersen, G. Camps-Valls, and L. Hansen. Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods. *IEEE Signal Processing Magazine*, 30[4]:16–29, 2013.

K. J. Arrow, H. Azawa, L. Hurwicz, and H. Uzawa. *Studies in linear and nonlinear programming*. Number 2. Stanford University Press, 1958.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

J. Benesty, M. M. Sondhi, Y. Huang, et al. *Springer handbook of speech processing*, volume 1. Springer, 2008.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *J. of Mach. Learning Research*, 13(Feb):281–305, 2012.

R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

D. Bertsekas. *Convex optimization algorithms*. Athena Scientific, 2015.

P. J. Bickel, Y. Ritov, A. B. Tsybakov, et al. Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 37(4):1705–1732, 2009.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning, 2016. arXiv:1606.04838.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

M. Brockmann, T. Gasser, and E. Herrmann. Locally adaptive bandwidth choice for kernel regression estimators. *J. of the Amer. Statistical Assoc.*, 88[24]:1302–1309, 1993.

E. J. Candès and C. Fernandez-Granda. Towards a mathematical theory of super-resolution. *Commun. on Pure and Appl. Math.*, 67[6]:906–956, 2014.

L. Chamon and A. Ribeiro. Probably approximately correct constrained learning. *Advances in Neural Information Processing Systems*, 33, 2020.

L. F. Chamon, Y. C. Eldar, and A. Ribeiro. Functional nonlinear sparse models. *arXiv preprint arXiv:1811.00577*, 2018.

L. F. Chamon, A. Amice, S. Paternain, and A. Ribeiro. Resilient control: Compromising to adapt. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5703–5710. IEEE, 2020.

B. Chen, J. Liang, N. Zheng, and J. C. Príncipe. Kernel least mean square with adaptive kernel size. *Neurocomputing*, 191(5):95–106, 2016.

Y. Chen, C. Caramanis, and S. Mannor. Robust sparse regression under adversarial corruption. In *ICML*, pages 774–782, 2013.

C. Cortes, M. Mohri, and A. Rostamizadeh. New generalization bounds for learning kernels. *arXiv preprint arXiv:0912.3309*, 2009.

A. Cotter, H. Jiang, M. R. Gupta, S. Wang, T. Narayan, S. You, and K. Sridharan. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *Journal of Machine Learning Research*, 20(172):1–59, 2019.

A. Datta, M. C. Tschantz, and A. Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *Proceedings on privacy enhancing technologies*, 2015(1):92–112, 2015.

M. Donini, L. Oneto, S. Ben-David, J. Shawe-Taylor, and M. Pontil. Empirical risk minimization under fairness constraints. *arXiv preprint arXiv:1802.08626*, 2018.

D. Donoho and I. Johnstone. Minimax estimation via wavelet shrinkage. *The Ann. of Stat.*, 26[3]:879–921, 1998.

H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

I. Erel, L. H. Stern, C. Tan, and M. S. Weisbach. Selecting directors using machine learning. Technical report, National Bureau of Economic Research, 2018.

B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline. Machine learning for medical imaging. *Radiographics*, 37(2):505–515, 2017.

J. Feng, H. Xu, S. Mannor, and S. Yan. Robust logistic regression and classification. In *NIPS*, pages 253–261, 2014.

G. M. Fung, O. L. Mangasarian, and A. J. Smola. Minimal kernel classifiers. *J. of Mach. Learning Research*, 3(Nov):303–321, 2002.

S. Gao, I. W.-H. Tsang, and L.-T. Chia. Sparse representation with kernels. *IEEE Transactions on Image Processing*, 22(2):423–434, 2013.

J. Garcıa and F. Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

A. K. Ghosh. Kernel discriminant analysis using case-specific smoothing parameters. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 38(5):1413–1418, 2008.

M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *J. of Mach. Learning Research*, 12(Jul):2211–2268, 2011.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

K. M. Hermann and P. Blunsom. Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173*, 2013.

T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *The Ann. of Stat.*, pages 1171–1220, 2008.

C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al. A practical guide to support vector classification, 2003.

P.-L. Hsu and H. Robbins. Complete convergence and the law of large numbers. *Proceedings of the National Academy of Sciences of the United States of America*, 33(2):25, 1947.

C. Jud, N. Mori, and P. C. Cattin. Sparse kernel machines for discontinuous registration and nonstationary regularization. In *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognition Workshops*, pages 9–16, 2016.

M. Kay, C. Matuszek, and S. A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3819–3828, 2015.

M. Kearns, S. Neel, A. Roth, and Z. S. Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2564–2572. PMLR, 2018.

G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. of Math. Anal. and Applicat.*, 33(1):82–95, 1971.

A. A. Kodiyan. An overview of ethical issues in using ai systems in hiring with a case study of amazon's ai based hiring tool. 2019.

V. Koltchinskii, D. Panchenko, et al. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of statistics*, 30(1):1–50, 2002.

J. Konečnỳ, B. McMahan, and D. Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

J. Konečnỳ, H. B. McMahan, P. Yu, F. X .and Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016b.

A. Koppel, G. Warnell, E. Stump, and A. Ribeiro. Parsimonious online learning with kernels via sparse projections in function space. In *J. of Mach. Learning Research*, January 2019.

G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

S. G. Krantz and H. R. Parks. *A primer of real analytic functions.* Springer Science & Business Media, 2002.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

M. Kuhn and K. Johnson. *Applied Predictive Modeling.* Springer, 2016.

B. Kuo, H. Ho, C. Li, C. Hung, and J. Taur. A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification. *IEEE J. of Selected Topics in Appl. Earth Observations and Remote Sensing*, 7(1):317–326, 2014.

G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *J. of Mach. Learning Research*, 5(Jan): 27–72, 2004.

Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

C.-H. Li, H.-H. Ho, Y.-L. Liu, C.-T. Lin, B.-C. Kuo, and J.-S. Taur. An automatic method for selecting the parameter of the normalized kernel function to support vector machines. *J. Inform. Sci. Eng.*, 28:1–15, 2012.

T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

A. Liapounoff. Sur les fonctions-vecteurs completement additives. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 4(6):465–478, 1940.

X. Liu, L. Wang, J. Zhang, and J. Yin. Sample-adaptive multiple kernel learning. In *AAAI Conf. on Artificial Intell.*, pages 1975–1981, 1993.

D. Lowd and C. Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, 2005.

G. Lugosi, N. Vayatis, et al. On the bayes-risk consistency of regularized boosting methods. *The Annals of Stat.*, 32(1):30–55, 2004.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

B. McMahan and D. Ramage. Federated learning: Collaborative machine learning without centralized training data. *Google Research Blog*, 3, 2017.

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *J. of Mach. Learning Research*, 6(Jul):1099–1125, 2005.

S. J. Narayanan, R. B. Bhatt, and I. Paramasivam. User localisation using wireless signal strength-an application for pattern classification using fuzzy decision tree. *Int. J. of Internet Protocol Technology*, 9(2-3):138–150, 2016.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Computing*, 24[2]: 227–234, 1995.

C. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *J. of Mach. Learning Research*, 6:1043–1071, 2005.

S. Paternain, M. Calvo-Fullana, L. F. Chamon, and A. Ribeiro. Learning safe policies via primal-dual methods. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6491–6497. IEEE, 2019.

M. Peifer, C. L. FO, S. Paternain, and A. Ribeiro. Locally adaptive kernel estimation using sparse functional programming. In *Asilomar Conference on Signals, Systems and Computers*, pages 2022–2026. IEEE, 2018.

M. Peifer, C. L. FO, S. Paternain, and A. Ribeiro. Sparse learning of parsimonious reproducing kernel Hilbert space models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019.

M. Peifer, L. F. Chamon, S. Paternain, and A. Ribeiro. Sparse multiresolution representations with adaptive kernels. *IEEE Transactions on Signal Processing*, 68:2031–2044, 2020.

L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

M. Poch, N. Bel, S. Espeja, and F. Navío. Ranking job offers for candidates: learning hidden knowledge from big data. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2076–2082. ACL (Association for Computational Linguistics), 2014.

A. Ribeiro. Ergodic stochastic optimization algorithms for wireless communication and networking. *IEEE Transactions on Signal Processing*, 58[12]:6369–6386, 2010.

R. T. Rockafellar. *Convex analysis*. Princeton university press, 2015.

R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

M. Rudelson and R. Vershynin. On sparse reconstruction from fourier and gaussian measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 61(8):1025–1045, 2008.

W. Rudin. *Functional analysis*. McGraw-Hill Science, Engineering & Mathematics, 1991.

A. Ruszczyński and W. Syski. On convergence of the stochastic subgradient method with on-line stepsize rules. *J. of Math. Anal. and Applicat.*, 114[2]:512–527, 1986.

B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Int. Conf. on Computational Learning Theory*, pages 416–426. Springer, 2001.

J. T. Schwartz. *Nonlinear functional analysis*, volume 4. CRC Press, 1969.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

A. Shapiro. On duality theory of convex semi-infinite programming. *Optimization*, 54[6]: 535–543, 2006.

A. Shapiro and K. Scheinberg. Duality and optimality conditions. In *Handbook of semidefinite programming*, pages 67–110. Springer, 2000.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

Y. Shen and S. Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pages 5739–5748. PMLR, 2019.

R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

A. Sinha, H. Namkoong, R. Volpi, and J. Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. 2000.

G. Tang, B. N. Bhaskar, P. Shah, and B. Recht. Compressed sensing off the grid. *IEEE Transactions on Information Theory*, 59[11]:7465–7490, 2013.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. of the Royal Stat. Soc.: Series B (Methodological)*, 58(1):267–288, 1996.

J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE transactions on information theory*, 52(3):1030–1051, 2006.

V. Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

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

P. Vincent and Y. Bengio. Kernel matching pursuit. *Mach. Learning*, 48(1-3):165–187, 2002.

S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

G. M. Weiss, K. Yoneda, and T. Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019.

A. G. Wilson and R. P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Int. Conf. on Mach. Learning*, pages III–1067–III–1075, 2013.

J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31 (2):210–227, 2008.

J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. Broeck. A semantic loss function for deep learning with symbolic knowledge. In *International Conference on Machine Learning*, pages 5502–5511. PMLR, 2018.

Z. Yang, A. Wilson, A. Smola, and L. Song. A la carte – Learning fast kernels. In *Int. Conf. on Artificial Intell. and Stat.*, pages 1098–1106, 2015.

H. Yu, S. Yang, and S. Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.

J. Yuan, L. Bo, K. Wang, and T. Yu. Adaptive spherical Gaussian kernel in sparse Bayesian learning framework for nonlinear regression. *Expert Syst. with Applicat.*, 36(2, Part 2): 3982–3989, 2009.

M. Yuan, T. T. Cai, et al. A reproducing kernel Hilbert space approach to functional linear regression. *The Ann. of Stat.*, 38(6):3412–3444, 2010.

M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: A flexible approach for fair classification. *J. Mach. Learn. Res.*, 20(75):1–42, 2019.

H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.

Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang. A survey of sparse representation: algorithms and applications. *IEEE access*, 3:490–530, 2015.

Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, 34:8–34, 2008.