INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT015

Thèse de doctorat

Structured Prediction with Output Regularization: Improving Statistical and Computational Efficiency

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 Ecole Doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Mathématiques et Informatique

Thèse présentée et soutenue à Palaiseau, le 19/04/2023, par

LUC BROGAT-MOTTE

Composition du Jury :

Stephan Clémençon Professor, Télécom Paris	Président, examinateur
Bharath Sriperumbudur Associate Professor, Pennsylvania State University	Rapporteur
Alain Rakotomamonjy Professor, Université de Rouen	Rapporteur (absent)
Anna Korba Assistant Professor, ENSAE Paris	Examinatrice
Carlo Ciliberto Associate Professor, University College London	Examinateur
Florence d'Alché-Buc Professor, Télécom Paris	Directrice de thèse
Juho Rousu Professor, Aalto university	Co-encadrant de thèse
Céline Brouard Researcher, INRAE Toulouse	Invitée
Alessandro Rudi Researcher, INRIA Paris	Invité

Contents

Remerciements

First of all, thank you to my thesis advisors, Florence, Juho, Céline, and Ale, for their guidance during my thesis. I am grateful for the opportunity I had to work with diversified and high-quality expertise. Thank you to Rémi Flamary for his precious help and for sharing with me his enthusiasm for optimal transport and machine learning. It has been a real pleasure working with you.

Thank you to Barath Sriperumbudur and Alain Rakotomamonjy for having reviewed this manuscript, and Stephan Clémençon, Anna Korba, Massimiliano Pontil, and Carlo Ciliberto, for accepting to be part of my jury. Thank you very much for your time, and your consideration of my work.

Ensuite, merci à toutes les personnes qui m'ont entouré pendant ma thèse, en particulier Baptiste et Charlotte, pour avoir constitué un environnement bienveillant et encourageant dans lequel je me suis senti bien. Enfin, je remercie mes parents, ainsi que mes grands frères et ma grande soeur, pour leur soutien inconditionnel, qui m'a toujours facilité les choses.

Abstract

Supervised learning algorithms aim at identifying relationships between inputs and outputs thanks to training sets of couples (input, output). The most studied setting of supervised learning deals with high-dimensional inputs but low-dimensional outputs, as, for example, real numbers in the case of regression, and the values zero or one in the case of binary classification. Nevertheless, being able to predict complex outputs, such as graphs, sequences, or images, allows for addressing much more practical tasks. This is the so-called *structured output prediction setting*.

The question that has motivated this thesis is the following:*How to take advantage of the structure of the output space in order to obtain statistically and computationally efficient structured prediction methods?* We try to answer this question through the lens of the structured prediction framework of *surrogate methods*.

More precisely, this manuscript starts by considering the problem of graph prediction. We propose to leverage the Gromov-Wasserstein (GW) distance, carrying a natural geometry for graph spaces, as a loss function. From this idea, we derive a new family of models for graph prediction: *GW barycentric models*. In a second contribution, we propose *a generalization of reduced-rank regression* which allows handling nonlinear output spaces. It consists in solving the surrogate regression problems appearing in surrogate methods thanks to a reduced-rank regression estimator. We carry out a theoretical study of the reduced-rank estimator, taking values in a Hilbert space of possibly infinite dimension, and prove under output regularity assumptions that the rank regularization is statistically and computationally beneficial. Our results extend the interest of reduced-rank regression beyond the standard setting where the optimum is assumed to be low-rank. In a third contribution, we propose the principle of *loss regularization*. The method aims at obtaining a statistical and computational gain in structured prediction, by exploiting additional output data, and regularity information on the loss function. We study theoretically under which setting the method is beneficial. Our results show, intuitively, that one had better adapt the level of detail of the structured outputs predicted with respect to the quantity of training data, to reduce the effects of the output variance (or labeling noise), and also to alleviate the computational complexity of the pre-image in surrogate methods.

Résumé

Les algorithmes d'apprentissage supervisé ont pour objectif d'identifier des relations entre des entrées et des sorties en utilisant des ensembles d'entraînement constitués de couples (entrée, sortie). La situation d'apprentissage supervisé la plus étudiée considère des entrées de grande dimension et des sorties de faible dimension, comme les nombres réels dans le cas de la régression, ou les valeurs zéro ou un dans le cas de la classification binaire. Néanmoins, être capable de prédire des sorties complexes, comme des graphes, des séquences ou des images, permet de résoudre un éventail plus large de tâches en pratique. C'est précisément le défi adressé par la prédiction structurée.

La question qui a motivé cette thèse est la suivante : comment exploiter la structure de l'espace de sortie pour obtenir des méthodes de prédiction structurée qui soient à la fois statistiquement et computationnellement performantes ? Plus précisément, comment tirer parti d'une faible dimension intrinsèque des données de sortie pour obtenir des gains statistiques et computationels ? Nous cherchons à répondre à cette question à travers le prisme des méthodes à noyaux, et plus particulièrement des méthodes de substitution pour la prédiction structurée. Ces méthodes de substitutions consistent à substituer les problèmes de prédiction structurée par des problèmes de régression, plus aisés à résoudre, car bénéficiant eux d'espaces de sortie avec des structures linéaires. Pour faire cela, chacune des sorties possibles est associée à une représentation dans un même espace de Hilbert. Cette famille de méthode peut être appliquées à une très large variété de problèmes de prédiction structurée, et bénéficie de solides garanties théoriques.

Ce manuscrit commence par aborder le problème de la prédiction de graphes. Nous proposons de mettre à profit la distance de Gromov-Wasserstein, définissant une géométrie naturelle pour les espaces de graphes, en tant que fonction de perte. Cela nous conduit à une nouvelle famille de modèles pour la prédiction de graphes : les modèles barycentriques de Gromov-Wasserstein. Deux versions de ces modèles sont proposées : une version non-paramétrique et une version utilisant un réseau de neurones. Nous fournissons des garanties statistiques pour la version non-paramétrique, notamment en démontrant la consistance, et aussi des bornes d'excès de risque. En outre, nous réalisons des expériences numériques à la fois sur un problème synthétique et sur un problème de prédiction de métabolites. Une implémentation Python de cette méthode est disponible sur GitHub.

Dans notre deuxième contribution, nous proposons une généralisation de la régression à rang réduit aux espaces de sortie non linéaires. La méthode proposée consiste à résoudre les problèmes de régression des méthodes de substitution en utilisant un estimateur de régression à rang réduit. Nous menons une étude théorique de

l'estimateur à rang réduit proposé, et prouvons, sous des hypothèses de régularité de sortie, que la régularisation de rang offre des avantages à la fois statistiques et computationnels. En particulier, nos résultats étendent l'intérêt de la régression à rang réduit au-delà du cas standard où l'optimum est supposé de rang fini et faible. La méthode de prédiction structurée de substitution induite par cet estimateur à rang réduit bénéficie des mêmes garanties statistiques que l'estimateur à rang réduit. Nos résultats théoriques sont illustrés par une étude expérimentale sur différentes tâches de prédiction structurée : la reconstruction d'image, la classification multilabels, et l'identification de métabolite.

Dans une troisième contribution, nous introduisons un principe de régularisation de la fonction de perte. La méthode proposée vise à obtenir des améliorations à la fois statistiques et computationnelles en prédiction structurée, grâce à l'exploitation de données de sortie supplémentaires, d'informations sur la régularité de la fonction de perte, et de la faible dimension intrinsèque des données de sortie. Nous étudions théoriquement les situations dans lesquelles cette méthode est effectivement bénéfique sur le plan statistique et computationnel. Les résultats théoriques sont illustrés par des études expérimentales sur différentes tâches de prédiction structurée: la reconstruction d'image, et la prédiction sur une sphère.

Finalement, les résultats de la deuxième et troisième contributions répondent à la question motivant la thèse en démontrant comment il est possible de tirer parti de la faible dimension intrinsèque des données de sortie pour obtenir des gains statistiques et computationels. Ces gains sont obtenus en contrôlant le niveau de détail des objets structurés prédits (via le contrôle de la dimension des représentations des sorties), en fonction de la quantité de données d'entraînement disponible, pour réduire les effets de la variance de sortie (ou du bruit d'étiquetage) d'une part, et pour alléger la complexité computationnelle de la prédiction d'autre part.

Notation

$\mathcal{X}$	Input space
$\mathcal{Y}$	Structured output space
$\mathcal{Z}$	Output Hilbert space
$\mathcal{H}_x$	Input embedding Hilbert space
$\mathcal{H}_y$	Output embedding Hilbert space
$\tilde{\mathcal{H}}_y$	Alternate output embedding Hilbert space
$\phi : \mathcal{X} \mapsto \mathcal{H}_x$	Input embedding map
$\psi : \mathcal{Y} \mapsto \mathcal{H}_y$	Output embedding map
$\tilde{\psi} : \mathcal{Y} \mapsto \tilde{\mathcal{H}}_y$	Alternate output embedding map
$\Delta : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$	Loss function
$\chi, \psi : \mathcal{Y} \mapsto \mathcal{H}_y$	Implicit Loss Embedding maps
$\mathcal{H}$	Hypothesis space for regression
$\mathcal{F}$	Hypothesis space for structured prediction
n/n_{te}	Quantity of training/test data
$\|.\|_{\mathrm{HS}}$	Hilbert-Schmidt norm
$\|.\|_{\infty}$	Operator norm
$h^*_\psi = x \to \mathbb{E}_{y\|x}[\psi(y)]$	Least-squares solution
$\hat{h}_\psi = x \to \mathbb{E}_{y\|x}[\psi(y)]$	Least-squares estimator
$c_\psi = \sup_{y\in\mathcal{Y}} \|\psi(y)\|^2$	Upper bound of the embedding norm
$C_\psi = \mathbb{E}[\psi(y) \otimes \psi(y)]$	Covariance of the embedding
$\epsilon_\psi = \psi(y) - h^*_\psi(x)$	Noise
$E_\psi = \mathbb{E}[\epsilon_\psi \otimes \epsilon_\psi]$	Output covariance or noise covariance
$\mathbb{1}_A$	Indicator function of the set A

For the sake of readability, subscripts and superscripts may be omitted when the dependency is clear from the context.

1
Introduction

Structured prediction goes beyond the standard supervised learning settings of classification and regression by dealing with *complex outputs*. Being able to predict sequences, graphs, functions, probability distributions, rankings of sets, allows to expand the interest of supervised learning to much more real-world applications: in computational biology (molecule structure prediction (Brouard et al., 2016a), enzyme network prediction (Geurts et al., 2006)), in natural language processing (handwriting recognition (Cortes et al., 2005; LeCun et al., 2015), language translation (Bahdanau et al., 2015), part-of-speech tagging and parsing (Collins, 2002)), or in computer vision (image segmentation (Nowozin et al., 2011), reconstruction of images (Weston et al., 2003), and 3D human pose estimation (Li and Chan, 2014), and scene graph prediction (Chen et al., 2019)).

The challenge of structured prediction is to deal with *high-dimensional* and *nonlinear* output spaces. Without correct handling of the output structure, learning methods suffer from the *curse of dimensionality*. Namely, statistical and computational performances deteriorate exponentially when the output dimension increases.

In this thesis, we try to address this challenge through the lens of *surrogate methods*. We propose structured prediction methods, and support them both with theoretical guarantees and experimental assessments on synthetic and real-world data. Our contributions mainly rely on nonparametric estimation tools, from the literature of kernel methods. We detail the research work undertaken throughout this thesis in the following section.

1.1 Motivations and contributions

The goal of this section is to present the questions that have motivated this thesis and provide an overview of the contributions. To this end, we start by introducing the questions and the main lines of research that we considered. Then, for the sake of illustration, we give an example of structured prediction task. Then, we propose a definition of the notion of structured space. Then, we present the structured prediction framework under which the research work in this manuscript has been mainly carried out. Finally, it will allow us to outline our different contributions.

Motivating questions and starting point. The general question that guided us in this work is the following: How to exploit the structure of the output space in order to design *statistically and computationally* efficient structured prediction methods? We were interested in tackling it through the lens of surrogate methods, motivated by their generality (as applying to a wide range of structured problems), and their amenability to theoretical analysis, allowing us to propose theoretically grounded methods

with well-understood behaviors. More precisely, we started with the idea of dealing with the curse of dimensionality occurring from the dimension of the output space in structured prediction, by choosing, or learning, appropriate outputs' representations. Intuitively, one aims at obtaining representations with a dimension equal to or close to the intrinsic dimension of the data, which directly hinges on the available a priori knowledge on the geometry of the output space. Nevertheless, we were interested in studying the possible benefits of controlling the complexity of the output representations depending on the quantity of training data. Furthermore, we were motivated by exploiting additional data sets of outputs (without the corresponding inputs), often available in large quantities in practice, typically allowing one to leverage partial knowledge on the output structure, as, for instance, a low-rank assumption. Moreover, we had in mind possible links between output representations learning and metric learning, or more generally loss learning. This led us to the following more specific questions: Would it be beneficial to add control on the complexity of the output representations, or on the regularity of the loss function, in order to obtain statistical and/or computational gains? How to leverage additional output data in structured prediction?

The example of metabolite identification. An important problem in metabolomics is to identify the small molecules, called metabolites, that are present in a biological sample. Mass spectrometry is a widespread method to extract distinctive features from a biological sample in the form of a tandem mass (MS/MS) spectrum. From this spectrum, one can identify the molecular graph structure of the molecules in the sample. Designing supervised learning methods able to accurately predict the molecular graph structure of a metabolite given its tandem mass spectrum is an active area of research. All the proposed methods in this work are experimentally assessed on the metabolite identification problem, which is illustrated in Figure 1.1. In this thesis, it will provide us with an archetypal example of structured prediction problem, as graph spaces are high-dimensional and non-linear spaces.

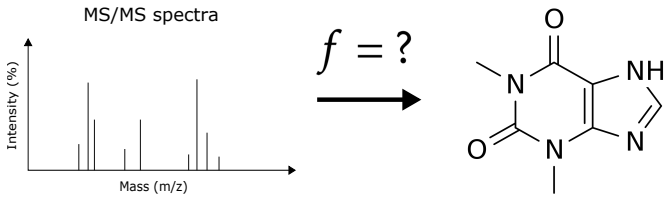

Figure 1.1: The metabolite identification problem.

Structured space. We define a structured space $(\mathcal{Y}, \psi)$ as a set equipped with an *embedding map* taking values in a Hilbert space $\mathcal{H}_y$. The linear structure of $\mathcal{H}_y$ provides a non-linear structure to $\mathcal{Y}$ through the non-linear map $\psi : \mathcal{Y} \to \mathcal{H}_y$. Interestingly, it turns out to be a very general definition. Indeed, we will see that equipping a set $\mathcal{Y}$ with an explicit map $\psi : \mathcal{Y} \to \mathcal{H}_y$, with a kernel $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ (Brouard et al., 2016b), with a metric $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, or more generally with a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ (Ciliberto et al., 2020) , all boils down, in most of the cases, to equip $\mathcal{Y}$, explicitly or implicitly, with an embedding map ψ taking values in a Hilbert space.

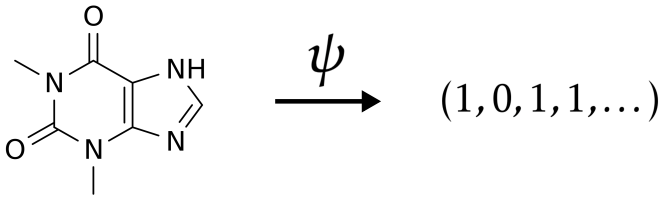

Figure 1.2: Fingerprints are explicit embeddings for molecular graphs.

Surrogate methods for structured prediction. Surrogate methods can be described as follows. First, an output embedding map ψ is chosen (explicitly or implicitly), proving $\mathcal{Y}$ with a structure. This gives rise to a surrogate regression problem $x \to \psi(y)$, which is more convenient to address than the structured prediction problem $x \to y$ as taking values in a linear space. This is solved using standard regression methods. Finally, a structured prediction estimator is obtained via a *pre-image* step (or decoding step) of an estimator of the surrogate regression problem. The construction of surrogate methods can be illustrated as in Figure 1.3. Surrogate methods is a general framework for structured prediction, applicable to a wide range of structured prediction problems, and benefiting from strong theoretical guarantees (Ciliberto et al., 2020). It will provide us with a sound framework for answering the questions raised above.

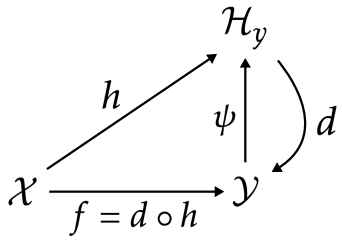

Figure 1.3: Surrogate methods for Structured Prediction.

Now, we are ready to introduce the three contributions presented in this manuscript.

How to deal with the non-linearity of the output space? Surrogate methods deals with the non-linear structure of the output space by choosing the output embedding map ψ. As said above, being able to define a relevant embedding map ψ relies, in practice, on the *available a priori information on the geometry of the output space*. For instance, molecular graphs can be represented with explicit embeddings called fingerprints, which are high-dimensional multi-label vectors, whose each label indicates the presence or absence of a certain molecular property (See Figure 1.2), or by using kernels for molecules (Ralaivola et al., 2005). In Section 2.8, we discuss in detail the role of ψ in surrogate methods. In particular, we will see that the choice of the output structure ψ is crucial for obtaining learnable surrogate problems, and also computationally tractable pre-image steps. The following contribution (I) proposes a method

dealing with the non-linearity of the output space, in the case of graph spaces, by choosing an appropriate loss.

(I) A general output geometry for graph prediction. The first contribution presented in this manuscript deals with *graph prediction*. Namely, supervised prediction with graphs as outputs. The main proposal is to use the *Gromov-Wasserstein distance*, and its extension *fused Gromov-Wasserstein distance*, as a loss in graph prediction. We argue that it provides a generic metric over graph spaces that deals with their non-linear geometries, and with the challenge of pre-image computation. We propose a kernel-based estimator based on kernel ridge surrogate regression, benefiting from strong theoretical guarantees. Then, we develop the idea by proposing a neural-network version of the method, with a sparse parametrization of the output graph space, in order to obtain both statistical and computational efficiency. The implementations are provided on GitHub. This work is presented in Chapter 3.

How to deal with the high dimensionality of the output space? Once the structure ψ is chosen, it remains to solve the induced high-dimensional surrogate regression problem. Fortunately, in most real-world problems, the intrinsic dimension of the data is smaller than the ambient dimension, namely the high-dimensional outputs of the supervised learning problem lie in fact in a subspace $\mathcal{Y}_0$ with a smaller dimension than the one of the known output space $\mathcal{Y}$. For instance, in the case of vector-valued regression, that is supervised prediction with a linear output space $\mathcal{Y} = \mathbb{R}^d$ for $d \in \mathbb{N}^*$, it is likely that some components of the outputs are linearly correlated. This means that the outputs lie in a linear subspace $\mathcal{Y}_0$ of $\mathcal{Y}$. In this case, a good idea, to obtain statistical and computational improvements, is to perform reduced-rank regression, which enforces the predictions to respect such structure of the output space. However, in structured prediction, $\mathcal{Y}$ is not linearly structured. Hence, it is unlikely that the outputs lie in a linear structured subspace $\mathcal{Y}_0 \subset \mathcal{Y}$. In this thesis, we refer to the principle of making the predictions respect regularities of the output spaces as *output regularization*, extending the idea of reduced-rank regression useful for high-dimensional linear output spaces, to high-dimensional non-linear output spaces. The two following contributions (II) and (III) propose methods dealing with the high-dimensionality of the output space by means of output regularization techniques.

(II) Reduced-rank regression for non-linear output spaces. In structured prediction, $\mathcal{Y}$ is not linearly structured, but the embedding space $\mathcal{H}_y$ is a Hilbert space. Hence, it is natural to extend the reduced-rank method to non-linear output spaces by solving the surrogate regression problems with reduced-rank regression. For instance, in the case of molecule prediction with fingerprints representation, i.e. $\mathcal{H}_y = \{0,1\}^d$ with $d \in \mathbb{N}^*$, it would correspond to assuming that some labels are linearly correlated. In this contribution, we consider the more general case where ψ can be induced by an output positive definite kernel over $\mathcal{Y}$ (Brouard et al., 2016b). To this end, we propose a reduced-rank estimator to solve surrogate regression problems with infinite dimensional output Hilbert space. We study theoretically this estimator. In particular, under *output regularity* assumptions, we prove that the estimator is statistically and computationally beneficial, in comparison with its full-rank counterpart. The structured predictor obtained from this regression estimator inherits the same benefits. The statistical gain is obtained by reducing the *output variance* (or noise). The computational gain is obtained by alleviating the pre-image step. The proposed method

is tested experimentally on various structured prediction problems. We present this work in Chapter 4.

(III) Calibrated structured prediction with loss regularization. The approach proposed in (II) makes use of the structure provided by ψ, and is shown to be calibrated with the induced loss: $\Delta(y, y') = \|\psi(y) - \psi(y')\|^2$. We may ask ourselves: Can we exploit the geometry provided by an embedding $\tilde{\psi} : \mathcal{Y} \to \tilde{\mathcal{H}}_y$ while being calibrated with another target loss $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$? Can we generalize the previous reduced-rank approach (II) to losses that are not of the form $\Delta(y, y') = \|\psi(y) - \psi(y')\|^2$? Indeed, for instance, if $\mathcal{Y} = \mathbb{R}^d$ for $d \in \mathbb{N}^*$, $\psi(y) = y \in \mathbb{R}^d$, and $\mathcal{Y}_0$ is the hypersphere with radius one in $\mathbb{R}^d$, one would prefer to exploit an embedding $\tilde{\psi}$ induced by a Gaussian kernel (as $\mathcal{Y}_0$ has a small intrinsic dimension through $\tilde{\psi}$, but not through ψ), but to be calibrated with the euclidean loss (the loss induced by ψ). Or, one may want to be calibrated with the geodesic distance on the hypersphere, which cannot be written as $\Delta(y, y') = \|\psi(y) - \psi(y')\|^2$. This third work proposes a method answering positively these questions. More precisely, given a target loss Δ, and an embedding $\tilde{\psi}$ defined via a kernel $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we propose an estimator able to exploit a low intrinsic dimension of the output data, by leveraging the structure defined by $\tilde{\psi}$, but being calibrated with the loss Δ. This generalizes the setting of contribution (II). Similarly to work (II), we show both theoretically and experimentally, how it allows improving the statistical and/or computational performance, in comparison with the full-rank counterpart. The proposed approach can be directly thought of as *regularizing the target loss* with respect to the regularity defined by the kernel k. Interestingly, this allows an intuitive understanding of the statistical gain obtained by the resulting structured estimator: one had better adapt the coarseness of the problem with respect to the output space depending on the quantity of training data, by controlling the regularity of the loss, or similarly, by making more or less fine-grained predictions, to reduce the effects of the output variance (or labeling noise), and also to alleviate the computational complexity of the pre-image in surrogate methods.

1.2 Publications

- (Brogat-Motte et al., 2022a) Luc Brogat-Motte, Rémi Flamary, Céline Brouard, Juho Rousu, and Florence d'Alché-Buc. Learning to predict graphs with fused Gromov-Wasserstein barycenters. In *International Conference on Machine Learning*, 2022. Reproduced in Chapter 3.

- (Brogat-Motte et al., 2022b) Luc Brogat-Motte, Alesandro Rudi, Céline Brouard, Juho Rousu, and Florence d'Alché-Buc. Vector-Valued Least-Squares Regression under Output Regularity Assumptions. In *Journal of Machine Learning Research*, 2022. Reproduced in Chapter 4, along with additional work.

- Luc Brogat-Motte, and Florence d'Alché-Buc. Structured Prediction with Loss Regularization. Preprint. Reproduced in Chapter 5.

- (Laforgue et al., 2020) Pierre Laforgue, Alex Lambert, Luc Brogat-Motte, and Florence d'Alché-Buc. Duality in RKHSs with infinite dimensional outputs: Application to robust losses. In *International Conference on Machine Learning*, 2020. The paper Laforgue et al. (2020) is a contribution to vector-valued regression with infinite dimensional output spaces using operator-valued kernels. It

provides methods for solving empirical risk minimization over vector-valued reproducing Hilbert spaces, for a wide range of loss functions, thanks to the use of a Double Representer Theorem. Not reproduced in this manuscript.

2
Background

Contents

This chapter provides the necessary background for the next chapters of this manuscript. We focus more particularly on the tools and results that play an important role in this thesis. Furthermore, we conclude this chapter by discussing the importance of a priori information on the output structure to overcome the curse of dimensionality with respect to the output dimension in structured prediction, and by commenting on our contributions in the light of this discussion.

2.1 Machine learning

In this section, we introduce supervised learning, and its mathematical formulation through the framework of statistical learning theory.

2.1.1 Learning from Data

What is Machine Learning? Computer programming aims at making computers able to perform tasks. This requires being able to design a finite sequence of basic operations solving the tasks in an accurate and computationally efficient manner. There are tasks for which such sequence is unknown. This is typically the case when dealing with high-dimensional input or output data such as, for instance, in image understanding, speech recognition, and sentence translation. On the other hand, there has been enormous growth in available data, and computers' computational capacities. *Machine Learning* (ML) tries to leverage this fact to overcome the aforementioned difficulty. ML algorithms aim at making computers able to automatically learn to carry out a task from a set of solved examples of the task. The science of ML lies at the intersection between statistics and computer science.

Supervised learning. Supervised learning methods aims at automatically identifying the underlying relationship between the inputs and the outputs, thanks to a data set of input/output couples $(x_i, y_i)_{i=1}^n$ called the *training set*. Hence, in supervised learning, tasks take the form of predicting the output associated with any given input.

Examples of supervised learning tasks. Supervised machine learning algorithms have met with great success in numerous applications. To name a few, this includes language translation (Bahdanau et al., 2015), image classification (Krizhevsky et al., 2012), speech recognition (Dahl et al., 2011), handwriting recognition (LeCun et al., 2015), spam detection (Wang et al., 2016), text categorization (Joachims, 1998), face detection (Deng et al., 2019), or weather forecasting (Deb et al., 2017). Two examples of supervised tasks are illustrated in Figures 2.1 and 2.2.

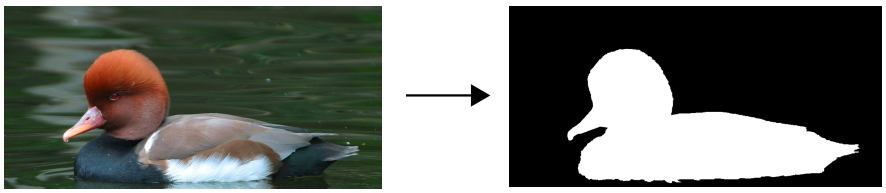

Figure 2.1: Example of supervised learning task: Image segmentation. (source: Nowozin et al. (2011))

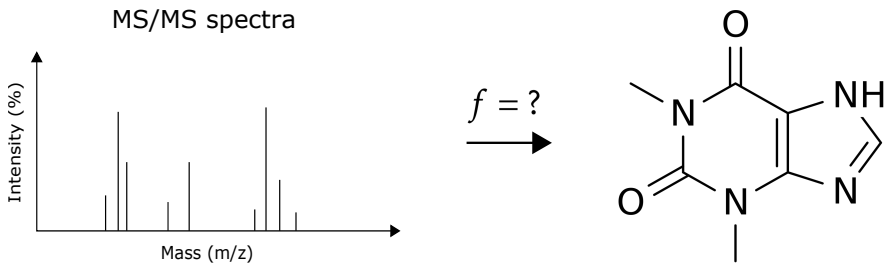

Figure 2.2: Example of supervised learning task: Metabolite identification (Brouard et al., 2016a).

Unsupervised learning. In the unsupervised learning setting, one is given a data set $(x_i)_{i=1}^n$, which is said to be *unlabeled* as, in contrast with supervised learning, the data are not couples (input, output/label). Despite this lack of labelling, there are valuable information to get from such data sets.

Examples of unsupervised learning tasks. Unsupervised learning problems include, for example, clustering (Hartigan and Wong, 1979; Von Luxburg, 2007), anomaly detection (Liu et al., 2008; Chandola et al., 2009), representation learning (Hoffmann, 2007). Representation learning plays an important role as a pre-processing step in supervised learning, and can also be used for data visualization (Ranzato et al., 2007). As an example of representation learning method, the kernel principal components analysis (KPCA) method estimates a fixed-size set of orthonormal functions maximizing the variance of its values over the data.

$$\max_{f\in\mathcal{F},\|f\|_{\mathcal{F}}=1} \sum_{i=1}^{n} f(x_i)^2 \tag{2.1}$$

Put in another way, one seeks for the functions, with controlled regularity, defining level sets that are higher on the data positions. Such functions constitute the non-linear main components of the data clouds. They allow to reduce the dimension of the data at hand by describing them with a number of components that is much smaller than the ambient dimension, and with a very small loss of information.

2.1.2 Statistical learning framework

Supervised Learning has been mathematically formalized through the framework of Statistical Learning Theory (Vapnik, 1999; Devroye et al., 2013). It makes it possible to carry out mathematical analysis of learning methods, and hence, helps design novel methods, allows to provide them with theoretical guarantees, to have a good understanding of their behaviors, and to compare them in terms of statistical and computational efficiency.

Statistical learning. Consider an input space $\mathcal{X}$, an output space $\mathcal{Y}$, a probability measure ρ on $\mathcal{X}\times\mathcal{Y}$, and a loss function $\Delta : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$ measuring the discrepancy between two outputs. The problem of supervised learning formulates as the one of finding the measurable function $f^* : \mathcal{X} \to \mathcal{Y}$ minimizing the expected risk

$$\mathcal{R}(f) = \int \Delta(f(x), y) d\rho(x, y), \tag{2.2}$$

by using a *training set* of couples $(x_i, y_i)_{i=1}^n$ independently drawn from the unknown distribution ρ. f^* is called the *Bayes predictor* and can be written as

$$f^*(x) = \arg\min_{\hat{y}\in\mathcal{Y}} \int \Delta(\hat{y}, y) d\rho(y|x) \tag{2.3}$$

where $\rho(y|x)$ is the conditional probability of y given x.

Empirical risk minimization (ERM). A general principle to obtain estimators f_n of f^* is to choose the function $f_n : \mathcal{X} \to \mathcal{Y}$ in a chosen hypothesis space $\mathcal{F}$ that minimizes the empirical risk (Devroye et al., 2013):

$$\mathcal{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \Delta(f(x_i), y_i). \tag{2.4}$$

$\mathcal{R}_n(f)$ is used as an approximation of the unknown expected risk $\mathcal{R}(f) = \mathbb{E}[\Delta(f(x), y)]$.

Statistical challenge of supervised learning. The statistical challenge of supervised learning is to obtain good prediction accuracy out of the training set. For instance, when using ERM, for example, one minimizes the empirical risk but actually wish to minimize the true risk. In general, without further information than the training data, successful learning is not possible, regardless of how large the training set is. Inferring the map f^* from a finite set of (noisy) observed values $(x_i, f^*(x_i) + \epsilon)_{i=1}^n$ is only made possible by the exploitation of a priori information on f^* (also refereed as inductive bias) restraining the hypothesis set $\mathcal{F}$. Such results are known as the *No-Free-Lunch theorems* (Devroye et al., 2013; Wolpert, 1996). The exploited bias goes from general assumptions on f^*, as for instance smoothness or manifold regularity (Belkin et al., 2005), verified by many real-world problems, to more specific assumptions exploiting knowledge from experts on given learning problems. The smaller the training data set, the stronger should be the bias in order to obtain the same statistical performance. This can be seen from the insightful and standard following decomposition of the risk:

$$\mathcal{R}(f) - \mathcal{R}(f^*) = \underbrace{\mathcal{R}(f) - \mathcal{R}(f^*_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{\mathcal{R}(f^*_{\mathcal{F}}) - \mathcal{R}(f^*)}_{\text{approximation error}} \tag{2.5}$$

where $f^*_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} \mathcal{R}(f)$.

The approximation error comes from the fact that one may choose a hypothesis space that does not contain the target f^*. It decreases with the "size" of $\mathcal{F}$. The estimations error stems from the error done when substituting $\mathcal{R}_n(f)$ with $\mathcal{R}(f)$. It increases with the "size" of $\mathcal{F}$. This leads to a trade-off between the two errors when choosing $\mathcal{F}$ to obtain the smallest risk possible. This is also called the *overfitting* and *underfitting* trade-off. We illustrate this trade-off in Figure 2.3.

How to deal with the trade-off overfitting/underfitting? It turns out that the effective measure on $\mathcal{F}$ that controls the trade-off between the estimation and the approximation errors is a notion of *expressiveness* of $\mathcal{F}$, namely its ability to fit any data set. In the case of the ERM estimator, the estimation error can be bounded as follows

$$\mathcal{R}(f_n) - \mathcal{R}(f^*_{\mathcal{F}}) \leq \mathcal{R}(f_n) - \mathcal{R}_n(f_n) + \underbrace{\mathcal{R}_n(f_n) - \mathcal{R}_n(f^*_{\mathcal{F}})}_{\leq 0} + \mathcal{R}_n(f^*_{\mathcal{F}}) - \mathcal{R}(f^*_{\mathcal{F}}) \tag{2.6}$$

$$\leq 2 \sup_{f \in \mathcal{F}} |\mathcal{R}_n(f) - \mathcal{R}(f)|. \tag{2.7}$$

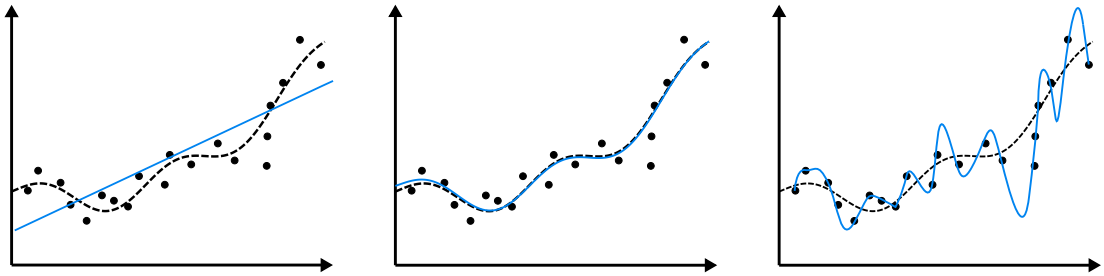

Figure 2.3: Illustration of the trade-off between overfitting and underfitting in regression. The blue line is estimated from the data points (black points). The dotted line is the unknown target function.

Then, one can try to prove a so-called *uniform bound* on $\sup_{f\in\mathcal{F}}|\mathcal{R}_n(f)-\mathcal{R}(f)|$. For instance, the *Vapnik-Chervonenkis (VC) dimension* of a hypothesis set is a measure of expressiveness for binary classifiers (Christmann and Steinwart, 2008), such that, for the $0-1$ loss

$$\sup_{f\in\mathcal{F}}|\mathcal{R}_n(f)-\mathcal{R}(f)| \leq O\left(\frac{\mathrm{VC}(\mathcal{F})}{\sqrt{n}}\right). \tag{2.8}$$

The *Rademacher complexity* (Bartlett and Mendelson, 2002) is another example of measure of expressiveness, which is also applicable to the regression setting, defined by

$$\mathrm{Rad}_n(\mathcal{F},\Delta,\rho) = \mathbb{E}_{\rho,\sigma}[\frac{1}{n}\sum_{i=1}^{n}\sigma_i\Delta(y_i,f(x_i))] \tag{2.9}$$

where the σ_i are i.i.d. Rademacher variables $P(\sigma_i = \pm 1) = 1/2$. One can show, in expectancy,

$$\sup_{f\in\mathcal{F}}|\mathcal{R}_n(f)-\mathcal{R}(f)| \leq 2\mathrm{Rad}(\mathcal{F},\Delta,\rho). \tag{2.10}$$

then $\mathrm{Rad}_n(\mathcal{F},\Delta,\rho)$ can be bounded as follows

$$\mathrm{Rad}_n(\mathcal{F},\Delta,\rho) \leq \frac{2L\kappa R}{\sqrt{n}} \tag{2.11}$$

if Δ is L-Lipschitz, and $\mathcal{F}$ are functions from a reproducing kernel Hilbert space $\mathcal{F}$ with bounded kernel $k(x,x') \leq \kappa^2$ such that $\|f\|_{\mathcal{F}} \leq R$ for any $f \in \mathcal{F}$. R can be understood then as controlling the expressiveness of $\mathcal{F}$. Therefore, one obtains a similar bound than with the VC-dimension.

How to choose the hypothesis space in practice? First, when choosing the hypothesis space $\mathcal{F}$, of course, one needs to *use the maximum a priori information* on the learning problem to find the strongest possible bias such that $f^* \in \mathcal{F}$. This will lead to a smaller estimation error with a zero approximation error. For instance, one might know that $f^* \in \mathcal{F}_0$ where $\mathcal{F}_0$ is a reproducing kernel Hilbert space (RKHS). Then, in order to obtain the best trade-off underfitting/overfitting, one also needs to control the expressiveness of the hypothesis space depending on the quantity of training data. For instance, in the case of the kernel ridge regression method, under standard assumptions, one considers $\mathcal{F} = \left\{f \in \mathcal{F}_0 \,|\, \|\hat{f}\|_{\mathcal{F}_0} \leq \frac{1}{\lambda}\right\}$ with $\lambda \sim \frac{1}{\sqrt{n}}$.

Computational challenge of supervised learning. Besides the statistical aspect, the other challenge of supervised learning is to obtain computationally tractable estimators. In particular, computing the ERM estimator can be very expensive when dealing with large amounts of data, high dimensional data, and the estimation of complex functions.

Different families of supervised learning problems. Depending on the output space and the loss $(\mathcal{Y}, \Delta)$ at hand, supervised learning problems fall into different subcategories. We give the most extensively studied subcategories in the Machine Learning literature in Table 2.1.

Subcategories	$\mathcal{Y}$	$\Delta(y, y')$
Regression	$\mathbb{R}^d$	$\|y - y'\|_2^2$
Binary classification	$\{0, 1\}$	$\mathbb{1}_{y \neq y'}$
Multi-class classification	$\{0, 1, \ldots, M\}$,	$\mathbb{1}_{y \neq y'}$
Multi-label classification	$\{0, 1\}^d$	$\sum_{i=1}^d \mathbb{1}_{y_i \neq y_i'}$

Table 2.1: Subcategories of Supervised learning depending on the output space $(\mathcal{Y}, \Delta)$ at hand. $d \in \mathbb{N}^*$, $M \in \mathbb{N}^*$.

2.2 Kernel methods

Kernel methods provide a tool for building space of functions. These spaces of functions are well-suited for Machine Learning. We start by presenting how they are constructed. Then, we show why they are convenient by using them for solving the empirical risk minimization problem.

2.2.1 Positive definite kernel

The building block of an RKHS is a positive definite kernel. It plays the role of an inner product for the possibly non-linear space $\mathcal{X}$, namely by mapping $\mathcal{X}$ into a feature space equipped with an inner product. In our case, this will offer us a minimum structure to the set $\mathcal{X}$ to perform statistical learning and will be seen as a priori knowledge on $\mathcal{X}$.

Definition. A positive definite (p.d.) kernel on $\mathcal{X}$ is an application $k : \mathcal{X} \to \mathcal{X} \to \mathbb{R}$ which is

1. Symmetric: $k(x, x') = k(x', x)$ for any $x, x' \in \mathcal{X}$.
2. Positive definite: $(k(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n \times n}$ is positive definite for any $(x_i)_{i=1}^n \in \mathcal{X}^n$.

$k(x, x')$ can be understood as a similarity measure between any $x, x' \in \mathcal{X}$. Moreover, notice that for any Hilbert space $\mathcal{H}$, and any embedding map $\phi : \mathcal{X} \to \mathcal{H}$, then $x, x' \to \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ defined a positive definite kernel. The opposite is also true as we shall see in the next section: any positive definite kernel can be written as a scalar product in a Hilbert space through an embedding map.

Examples of kernels. There is a vast literature that focuses on designing p.d. kernels. The goal is to construct kernel providing a relevant structure to the input space $\mathcal{X}$ at hand, by exploiting the a priori knowledge on $\mathcal{X}$, while also keeping the computations tractable. We give some examples of kernels below.

- Three examples of kernels on $\mathbb{R}^d$ with $d \in \mathbb{N}^*$:
  1. $k(x, x') = \langle x, x' \rangle_{\mathbb{R}^d}$ is a p.d. kernel on $\mathbb{R}^d$ called linear kernel.
  2. $k(x, x') = (1 + \langle x, x' \rangle_{\mathbb{R}^d})^p$ with $p \in \mathbb{N}^*$ is a p.d. kernel on $\mathbb{R}^d$ called polynomial kernel.
  3. $k(x, x') = \exp(\gamma \|x - x'\|^2)$ with $\gamma > 0$ is a p.d. kernel called gaussian kernel.
- An example of kernel for graphs is the random walk kernel: given two graphs x, x', random walks are performed on both, and $k(x, x')$ is defined as the number of matching walks.
- An example of kernel for strings is the spectrum kernel (Leslie et al., 2001): $k(x, x') = \sum_{s \in S(k)} o_s(x) o_s(x')$ where $S(k)$ is the set of all possible sequence of length k from a given finite alphabet $\mathcal{A}$ (hence $|S(k)| = |\mathcal{A}|^k$), and $o_s(x)$ is the number of occurrence of s in x.
- An example of kernel for probability distributions is the probability product kernel (PPK) (Jebara et al., 2004): $k(p, p') = \langle p^\beta, p'^\beta \rangle_{L^2}$ with $\beta > 0$ (requiring $p^\beta, p'^\beta \in L^2$).

2.2.2 Scalar-valued reproducing kernel Hilbert spaces

We present now how scalar-valued Reproducing Kernel Hilbert Spaces (RKHSs) are constructed. We refer the reader to (Aronszajn, 1950; Christmann and Steinwart, 2008) for more details on RKHSs.

Construction of RKHSs. Given a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, an RKHS is built by taking linear combinations of the functions $k(x, .)$

$$\mathcal{H}_x^0 = \operatorname{span}\{k(x, .) \,|\, x \in \mathcal{X}\} \tag{2.12}$$

and equipping it with the inner product $\langle k(x, .), k(x', .) \rangle_{\mathcal{H}_x^0} := k(x, x')$. The inner product space $\mathcal{H}_x^0$ is then completed into a Hilbert space taking the completion according to the norm induced by the scalar product $\langle ., . \rangle_{\mathcal{H}_x^0}$:

$$\mathcal{H}_x = \overline{\operatorname{span}\{k(x, .) \,|\, x \in \mathcal{X}\}}. \tag{2.13}$$

Completeness allows to have the good convergence properties of finite-dimensional Euclidean spaces.

By construction, the RKHS $\mathcal{H}_x$ verifies the following property.

Reproducing property. For any $h \in \mathcal{H}_x$ and $x \in \mathcal{X}$, the following *reproducing property* holds:

$$\langle h, k(x, .) \rangle_{\mathcal{H}_x} = h(x). \tag{2.14}$$

If a kernel k verifies the *reproducing property* for a Hilbert space $\mathcal{H}$, and if $\mathcal{H}$ contains all the functions $k(x, .)$, then k is called a *reproducing kernel* for $\mathcal{H}$.

Canonical feature map. As said above, any positive kernel can be written as a scalar product in a Hilbert space through a feature map. This holds from the Reproducing property, and defining $\phi(x) = k(x,.) \in \mathcal{H}_x$.

Notice that the canonical feature map $\phi(x)$ is infinite dimensional but there is no need to compute $\phi(x)$ explicitly to evaluate $k(x,x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}_x}$. Furthermore, there is no unicity of the feature map: one can find another Hilbert space $\tilde{\mathcal{H}}_x$ and a feature map $\tilde{\phi} : \mathcal{X} \to \tilde{\mathcal{H}}_x$ such that $k(x,x') = \langle \tilde{\phi}(x), \tilde{\phi}(x') \rangle_{\tilde{\mathcal{H}}_x}$. Furthermore, while any p. d. kernel can be seen as a scalar product, conversely, any feature map $\tilde{\phi} : \mathcal{X} \to \mathcal{H}$ with $\mathcal{H}$ a Hilbert space, defines a p. d. kernel $k(x,x') = \langle \tilde{\phi}(x), \tilde{\phi}(x') \rangle_{\mathcal{H}}$.

One-to-one correspondence between RKHSs and reproducing kernels. It turns out that there is a unique RKHS associated with a reproducing kernel, and conversely, a unique reproducing kernel associated with a RKHS.

Because of this correspondence, we are allowed to call $\mathcal{H}_x$ the *associated RKHS* of k, and conversely. Moreover, an alternative definition of RKHSs is possible as Hilbert spaces equipped with reproducing kernels.

Another possible definition of RKHSs comes from the following property.

Continuity of the evaluations functionals. From the Cauchy-Schwarz inequality, it is clear that the following map is continuous from the RKHS $\mathcal{H}_x$ to $\mathbb{R}$

$$F_x : h \to h(x). \tag{2.15}$$

It turns out that the reverse is also true, using the Riesz representation theorem: if the continuity of functions evaluations holds for a Hilbert space $\mathcal{H}$ then it is a RKHS. Notice that it leads to a second alternative definition of RKHS, which does not require introducing a kernel.

2.2.3 Vector-valued reproducing kernel Hilbert spaces

The theory of vector-valued RKHSs (vv-RKHSs) extends the theory of scalar-valued RKHS by enabling to construct space of vector-valued functions taking values in a Hilbert space $\mathcal{Y}$. We refer the reader to (Senkene and Tempel'man, 1973; Micchelli and Pontil, 2005; Carmeli et al., 2010; Caponnetto et al., 2008) for more details on vv-RKHSs.

Operator-valued positive definite kernel. We note A^* the adjoint of any operator A. An operator-valued kernel is a an application $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ with values in the set of bounded linear operator on $\mathcal{Y}$, satisfying the two following properties:

1. $K(x,x') = K(x',x)^*$.
2. $\sum_{i,j=1}^{n} \langle K(x_i,x'_j)y_i, y_j \rangle_{\mathcal{Y}} \geq 0$ for any $n \in \mathbb{N}^*$, $(x_1,y_1),\ldots,(x_n,y_n) \in \mathcal{X} \times \mathcal{Y}$.

Examples of operator-valued kernels. We give two examples of operator-valued kernels, and refer the reader to Caponnetto et al. (2008) for more examples.

- Separable kernel: $K(x,x') = k(x,x')A$ with k a p.d. scalar-valued kernel, and $A \in \mathcal{L}(\mathcal{Y})$.
- Sum of separable kernels: $K(x,x') = \sum_{i=1}^{d} K_i(x,x')$ with $d \in \mathbb{N}^*$, and $(K_i)_{i=1}^{d}$ are d separable kernels.

Operator-valued kernels are the building blocks of vector-valued RKHSs. vv-RKHSs are constructed in the same manner as RKHSs with scalar-valued kernels. All the properties and observations given in the previous sections will have their counterpart in this section.

Construction of Vector-valued reproducing kernel Hilbert spaces. Akin to scalar-valued kernel, the vector-valued RKHS $\mathcal{H}$ the associated RKHS of K is constructed by the completion of the space generated by linear combinations of the functions $K(x,.)y$ with $x \in \mathcal{X}, y \in \mathcal{Y}$:

$$\mathcal{H} = \overline{\text{span}\{K(x,.)y \,|\, (x,y) \in \mathcal{X} \times \mathcal{Y}\}}. \tag{2.16}$$

according to the norm induced by the scalar product

$$\langle K(x,.)y, k(x',.)y' \rangle_{\mathcal{H}} := \langle K(x,x')y, y' \rangle_{\mathcal{Y}}. \tag{2.17}$$

By construction, the RKHS $\mathcal{H}$ verifies the following property.

Reproducing property. For any $h \in \mathcal{H}$, $x \in \mathcal{X}, y \in \mathcal{Y}$, the following *reproducing property* holds:

$$\langle h, K(x,.)y \rangle_{\mathcal{H}_x} = \langle h(x), y \rangle_{\mathcal{Y}}. \tag{2.18}$$

Similarly to scalar-valued kernels, if an operator-valued kernel K verifies the *reproducing property* for a Hilbert space $\mathcal{H}$, and if $\mathcal{H}$ contains all the functions $K(x,.)y$, then K is called a *reproducing kernel* for $\mathcal{H}$.

Moreover, we also have the counterpart of the definition of an RKHS as a Hilbert space equipped with a reproducing kernel, because of the following property.

One-to-one correspondence between RKHSs and reproducing kernels. There is a unique vector-valued RKHS associated with an operator-valued reproducing kernel K, called the associated RKHS of K, and conversely, a unique operator-valued reproducing kernel K associated with a vector-valued RKHS.

Furthermore, we also have the counterpart of the definition of RKHS via the continuity of functions evaluations.

Continuity of functions evaluations. $\mathcal{H}$ is a vector-valued RKHS if and only if the functions evaluations

$$F_{x,y} : h \to \langle h(x), y \rangle_{\mathcal{Y}}. \tag{2.19}$$

are continuous for any $x \in \mathcal{X}, y \in \mathcal{Y}$.

2.2.4 Empirical risk minimization over RKHS

In this section, we explain why RKHSs are very good candidates to solve empirical risk minimization (ERM) problems.

For this purpose, we recall from Section 2.1.2, what is at stake when choosing the hypothesis space for solving ERM problems.

How to design the hypothesis space $\mathcal{H}$? When choosing the hypothesis spaces for solving ERM problems one faces two stakes:

(1) (Modelling). In order to obtain the best possible estimation of the target f^*, one needs to choose the hypothesis space, as follows:

  (a) (Minimum bias $\forall n$). $\mathcal{H}$ should be the smallest possible but containing f^* exploiting available a priori information on the problem.

  (b) (Complexity control w.r.t n). The expressiveness of $\mathcal{H}$ should be chosen with respect to the quantity of training data to achieve the best possible underfitting/overfitting trade-off.

(2) (Computational concern). The solution should be computationally tractable.

Let's examine these requirements in the case of RKHSs.

(1.a) Kernels allow to express prior knowledge. A priori regularity information on f^* can be quite conveniently induced on the RKHS by the choice of the kernel. As said above, there is a vast literature about designing p.d. kernels. General regularity assumptions as smoothness of the target can be carried by radial kernels. When dealing with discrete space as spaces of graphs, or strings, a common methodology consists in trying to find a "good similarity measure" between two objects of the input space, then verifying the positive definiteness. This has shown to be efficient on a large range of data, making RKHSs a general and user-friendly way of building function spaces going from any input space $\mathcal{X}$ to any Hilbert space $\mathcal{Y}$.

(1.a) Possible high expressiveness of RKHSs. One may aim at estimating a complex target function, or one may know little a priori information about the target. Therefore, it may be necessary to be able to build "large" hypothesis spaces, namely with high expressiveness. The construction method of RKHSs does allow to generate such expressive spaces. The concept of *universality* of a kernel has been defined as the density of the induced RKHS with respect to the supremum norm in the space of all continuous functions $h : \mathcal{X} \to \mathcal{Y}$ (Micchelli et al., 2006; Caponnetto et al., 2008; Christmann and Steinwart, 2008). For instance, in the case where $\mathcal{X}$ is a compact subset of $\mathbb{R}^d$, the Gaussian kernel has been shown to be universal.

(1.b) Continuous control of the expressiveness. Moreover, RKHSs come with a natural control of the expressiveness via the RKHS norm $\|h\|_{\mathcal{H}}$, typically by summing a so-called regularization term to the empirical risk, controlled by a parameter $\lambda > 0$:

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \Delta(h(x_i), y_i) + \lambda \|h\|_{\mathcal{H}}^2. \tag{2.20}$$

(2) Computational tractability. It is easy to show that the function $h \in \mathcal{H}$ minimizing the Eq. (2.20), can be written

$$h(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \quad \text{with} \quad \alpha \in \mathbb{R}^n. \tag{2.21}$$

This result, called *representer theorem* in the kernel method literature, makes the optimization problem Eq. (2.20), over an infinite dimensional space $\mathcal{H}$, a finite dimensional optimization problem over $\mathbb{R}^n$. Moreover, if Δ is convex then the ERM becomes a convex optimization problem over $\mathbb{R}^n$, making it computationally tractable.

To conclude, RKHSs fulfill the modeling and computational stakes given above.

Another important strength of kernel methods is the following.

Amenability to theoretical analysis. A good property of kernel methods is that they are very amenable to mathematical study, allowing, among others, to derive statistical guarantees. Indeed, the linear parametrization of the hypothesis space is not only beneficial for computational tractability, but also the convenience of mathematical analysis. Using the RKHS $\mathcal{H}$ induced by a positive definite kernel $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ corresponds to fit the model $h(x) = H\phi(x)$. Then, the theoretical analysis can be carried out as with a linear model, which is convenient, as we shall see in Section 2.4.

2.3 Neural networks

The goal of this section is to briefly present neural networks as another way of building hypothesis space. Such modeling has obtained great practical successes, in particular when dealing with large scale data sets. We refer the reader to Goodfellow et al. (2016) for a detailed introduction to neural networks.

Neural network (NN) modeling. A neural network f with depth $D \in \mathbb{N}^*$ is a parameterized family of functions defined as the composition of D layers $f_i = \mathbb{R}^{n_{i-1}} \mapsto \mathbb{R}^{n_i}$ of sizes $n_1, \ldots, n_D \in \mathbb{N}^*$

$$f_{W,b}(x) = f_D \circ f_{D-1} \circ \cdots \circ f_1(x) \tag{2.22}$$

where each layer f_i is an affine map composed with a non-linear map defined via a so-called *activation function* $\sigma : \mathbb{R} \mapsto \mathbb{R}$ applied pointwise on the outputs of each layer:

$$f_i(x) = \sigma(W_i f_{i-1}(x) + b_i) \tag{2.23}$$

with $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$, $b_i \in R^{n_i}$, and $n_0 \in \mathbb{N}^*$ is the dimension of the input space. We say that $f_{W,b} : \mathbb{R}^{n_0} \mapsto \mathbb{R}^{n_D}$ has $D-1$ hidden layers. The parameters of f are $(W_i, b_i)_{i=1}^D$. *Deep learning* refers to deep neural networks, namely neural networks with a large number of layers.

Architecture of a network. NN models can be designed by deciding how to connect the neurons of one layer to the neurons of the next layer. This corresponds to imposing zero values on some chosen components of the W_i. Finding the appropriate neural networks' architectures for solving given machine learning tasks has been a very active research area in the past few years. A non-comprehensive list of commonly used architectures includes:

- Fully-connected NNs (FC-NNs) do not impose constraints on the $(W_i)_{i=1}^D$.
- Convolutional NNs (CNNs) (LeCun et al., 1995) deal with pattern recognition tasks (e.g. on image, speech, or time-series data).
- Recurrent NNs (RNNs) (Sak et al., 2014) or Transformers (Vaswani et al., 2017) deal with sequential data (e.g. sentences, biological sequences, videos).
- Graph NNs (GNNs) (Scarselli et al., 2008) deal with graph data (e.g. molecules, social networks, physical systems (Sanchez-Gonzalez et al., 2018)).
- Generative NNs (e.g. GANs (Goodfellow et al., 2020), VAEs (Kingma and Welling, 2013), flow-based models (Prenger et al., 2019), diffusion models (Ho et al., 2020)) aim at learning to generate samples (e.g. images) from an unknown distribution.

The challenge of ERM with NNs. Solving the ERM problem using a NN model is very challenging as it is a *non-convex optimization problem* with local minima, saddle points, and wide flat regions. Moreover, performing gradient descent is also *computationally extensive* because of the gradient computations. Therefore, solving the ERM with NNs has required implementing multiple strategies including: the back-propagation algorithm (Rumelhart et al., 1995) for efficient computation of the gradient of the loss $\Delta(f(x_i), y_i)$ with respect to the parameters of the NN model f, batch or stochastic gradient computation, parallel computing using Graphics processing units (GPUs), choosing efficient optimizers (e.g. using momentum (Polyak, 1964), or adaptative learning rates (Duchi et al., 2011)), parameters initialization strategies, or batch normalization (Ioffe and Szegedy, 2015).

Learning theory of deep learning. In comparison with kernel methods NN-based methods are less amenable to theoretical study. In particular, quantifying the expressiveness of NN models is difficult, as also depends on the capacity of the optimization algorithms in solving the ERM problem, while existing theoretical guarantees for non-convex optimization problems are weak in general. It is worth mentioning the two following lines of research in the emerging theory of deep learning. The first line consists in studying the generalization behavior of over-parameterized neural networks, by considering infinite-width neural networks, showing that at this limit a NN model can be assimilated to a RKHS model with a particular kernel called Neural Tangent Kernel (Jacot et al., 2018). This allows using the generalization theory of kernel methods. Another line consists in studying the benefits of implicit bias, when fitting neural networks, due to the non-optimal optimization (Chizat and Bach, 2020).

Role of neural networks in this manuscript. The different contributions presented in this manuscript mainly consider RKHS as hypothesis spaces. As explained above, this allows us to make easier proofs, and obtain strong guarantees. Moreover, it is worth mentioning that kernel methods can outperform neural networks on a variety of real-world problems, typically when dealing with small and high-dimensional data sets, for instance in computational biology. Furthermore, if one problem is better suited for NNs, most of the proposed methods can be straightforwardly adapted to neural network modeling, by changing the hypothesis space by neural network models. In Chapter 3 one contribution is precisely to propose a neural net-

work graph prediction method, based on the structured prediction model of kernel-based surrogate methods. The method proposed in Chapter 4 can be adapted as follows: Let U be defined with the estimated main components in $\mathcal{H}_y$ as row, then learn to predict the coordinate $U\psi(y) \in \mathbb{R}^p$ of the projected $P\psi(y) = U^*U\psi(y)$ with a neural network $\hat{h} : \mathcal{X} \to \mathbb{R}^p$ instead of a kernel ridge, by minimizing the train mean squared error. Then, the pre-image can be computed as $\hat{y} = \arg\min_y \|U^*\hat{h} - \psi(y)\|^2_{\mathcal{H}_y} = \arg\min_y -2\langle \hat{h}(x), U\psi(y)\rangle_{\mathbb{R}^d} + k_y(y,y)$. The method proposed in Chapter 5 could be adapted as proposed in Chapter 3. Nevertheless, the adaptation of surrogate methods with neural networks is not the main focus of this manuscript and is left for further research. In particular, practical success would require choosing the best strategies for solving ERM as mentioned above. Finally, the theoretical results obtained can be valuable beyond kernel methods, as bringing insight into the understanding of the role of the output geometry in structured prediction.

2.4 Regularized least-squares regression

In this section, we present the kernel ridge regression (KRR) method for solving least-squares regression problems. First, we introduce the learning setting of least-squares regression. Then, we describe the kernel ridge method. Finally, we provide theoretical guarantees for the KRR estimator, with sketches of the proofs. We refer the reader to Caponnetto and De Vito (2007); Ciliberto et al. (2020) for more details.

2.4.1 Setting of least-squares regression

Vector-valued Least-squares regression (Vv-LS). Vv-LS is defined as the problem of estimating the function $h : \mathcal{X} \to \mathcal{Y}$, taking values in a separable Hilbert space $\mathcal{Y}$, minimizing the expected risk

$$\mathcal{R}(h) = \mathbb{E}_\rho[\|h(x) - y\|^2_{\mathcal{Y}}], \tag{2.24}$$

given a finite set $(x_i, y_i)_{i=1}^n$ independently drawn from an unknown distribution ρ on $\mathcal{X} \times \mathcal{Y}$.

It corresponds to the supervised learning setting (described above) where: (1) the output space $\mathcal{Y}$ is a Hilbert space, and (2) the loss Δ is the squared norm (squared loss) $\Delta(y, y') = \|y - y'\|^2_{\mathcal{Y}}$.

Characterization of the optimal solution. The measurable function $h^* : \mathcal{X} \to \mathcal{Y}$ minimizing $\mathcal{R}(h)$ is given by

$$h^*(x) = \mathbb{E}_{\rho(y|x)}[y]. \tag{2.25}$$

See, for instance, Lemma A.2 in Ciliberto et al. (2020) for a proof of this result.

2.4.2 Kernel ridge regression

The kernel ridge regression (KRR) method consists in solving the empirical risk minimization problem, associated with the squared loss, over a vector-valued RKHS as hypothesis space, and controlling the expressiveness of the estimator via the RKHS norm.

KRR estimator. The KRR estimator is defined as the minimizer of the following *regularized empirical risk*:

$$\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \|h(x_i) - y_i\|_{\mathcal{Y}}^2 + \lambda\|h\|_{\mathcal{H}}^2 \tag{2.26}$$

where $\mathcal{H}$ is the RKHS associated to an operator-valued kernel $K : \mathcal{X}\times\mathcal{X} \to \mathcal{L}(\mathcal{Y})$.

Closed-form expression. In this thesis, we will always consider operator-valued kernels of the form $K(x,x') = k(x,x')I_{\mathcal{Y}}$, where $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is a positive definite scalar-valued kernel on $\mathcal{X}$. In this case, the solution of the problem above can be computed in closed-form as follows:

$$\hat{h}(x) = \sum_{i=1}^{n} \alpha_i(x) y_i, \quad \text{with } \alpha(x) = (K + n\lambda)^{-1} k_x \tag{2.27}$$

where $K = (k(x_i,x_j))_{i,j=1}^n \in \mathbb{R}^{n\times n}$, and $k_x = (k(x,x_i))_{i=1}^n \in \mathbb{R}^n$.

Computational complexity. We recall the computational complexity of kernel ridge regression in Table 2.2. Notice that the time complexity of computing the kernel ridge estimator is dominated by the inversion of the gram matrix $K + \lambda I \in \mathbb{R}^{n\times n}$. The training computational complexity can be alleviated using approximation methods such as Nyström subsampling or random features. Under regularity assumptions on h^*, the approximated estimators benefit from the same learning rates as the not approximated ones (Rudi et al., 2015; Sterge et al., 2020; Rudi and Rosasco, 2017), with significantly reduced computational complexities.

	Time	Space
Training	$\mathcal{O}(n^3 + n^2 \times c)$	$\mathcal{O}(n^2)$
Evaluation	$\mathcal{O}(n \times c)$	$\mathcal{O}(n)$

Table 2.2: Computational complexity of kernel ridge regression. c is the cost of one kernel evaluation.

2.4.3 Statistical analysis of KRR

In this section, we recall theoretical results on the ridge estimate. We present the main ideas and steps of the proofs. We refer the reader to Caponnetto and De Vito (2007); Ciliberto et al. (2020) for the full details of the analysis of KRR. We denote $f(u) \lesssim g(u)$ if there exists $c > 0$ such that $\forall\, u, f(u) \leq cg(u)$, and use this notation to do not keep track of multiplicative constants that do not depend on the parameters of interest.

KRR estimator. Because $\hat{h}$ belongs to the chosen vv-RKHS, there exists $H_n \in \mathcal{H}_y \otimes \mathcal{H}_x$, with $\|H_n\|_{\text{HS}} < +\infty$, such that

$$\hat{h}(x) = H_n \phi(x). \tag{2.28}$$

So, defining the empirical covariances

$$V_n = \frac{1}{n}\sum_{i=1}^{n} y_i \otimes \phi(x_i), \qquad C_n = \frac{1}{n}\sum_{i=1}^{n} \phi(x)\otimes\phi(x), \tag{2.29}$$

the regularized empirical risk can be expressed as

$$\frac{1}{n}\sum_{i=1}^{n}\|\hat{h}(x_i)-y_i)\|_{\mathcal{Y}}^2+\lambda\|h\|_{\mathcal{H}_y\otimes\mathcal{H}_x}^2=\frac{1}{n}\sum_{i=1}^{n}\|H_n\phi(x_i)-y_i)\|_{\mathcal{Y}}^2+\lambda\|H_n\|_{\mathrm{HS}}^2 \tag{2.30}$$

$$=\frac{1}{n}\operatorname{Tr}((H_n\phi(x_i)-y_i))\otimes(H_n\phi(x_i)-y_i))+\lambda\|H_n\|_{\mathrm{HS}}^2 \tag{2.31}$$

$$=\operatorname{Tr}(H_n(C_n+\lambda I)H_n^*)-2\operatorname{Tr}(H_nV_n^*)+\text{cst indep. of } H_n \tag{2.32}$$

$$=\|H_n(C_n+\lambda I)^{1/2}-V_n(C_n+\lambda I)^{-1/2}\|_{\mathrm{HS}}^2+\text{cst indep. } H_n. \tag{2.33}$$

Then, one can obtain H_n as the minimizer of Eq. (2.39):

$$H_n=V_n(C_n+\lambda)^{-1}. \tag{2.34}$$

LS excess-risk is the L2-distance to target. First, from the characterization of the optimal solution: $h^*(x)=\mathbb{E}_{y|x}[y]$, it is easy to show that

$$\mathcal{R}(\hat{h})-\mathcal{R}(h^*)=\mathbb{E}_x[\|\hat{h}(x)-h^*(x)\|_{\mathcal{Y}}^2]. \tag{2.35}$$

Attainability assumption. Then, assuming the target h^* belongs to the chosen vv-RKHS, induced by the kernel $\Gamma(x,x')=k(x,x')I_{\mathcal{Y}}$, it corresponds to the existence of $H\in\mathcal{H}_y\otimes\mathcal{H}_x$ such that:

$$h^*(x)=H\phi(x) \tag{2.36}$$

with $\|H\|_{\mathrm{HS}}<+\infty$, and $\phi(x)=k(x,.)$. In this case, with similar derivations than previously for computing H_n, the H verifying this condition with minimal Hilbert-Schmidt norm (see Lemma B.9 in Ciliberto et al., 2020) is:

$$H=VC^{\dagger} \tag{2.37}$$

with

$$V=\mathbb{E}[y\otimes\phi(x)],\qquad C=\mathbb{E}[\phi(x)\otimes\phi(x)], \tag{2.38}$$

and where $C^{\dagger}$ denotes the Moore-Penrose generalized inverse (Engl et al., 1996).

KRR excess-risk as linear RR excess-risk. Moreover, with similar derivations than for the empirical risk, one can express the LS excess-risk as

$$\mathbb{E}[\|\hat{h}(x)-h^*(x)\|_{\mathcal{Y}}^2]=\|(H_n-H)C^{1/2}\|_{\mathrm{HS}}^2. \tag{2.39}$$

KRR as covariance estimation. At this point, we see that the kernel ridge estimate writes as the product of empirical covariance operators, while the optimal solution writes as the same product but with the ideal covariance operators. Moreover, the excess-risk writes as a Hilbert-Schmidt norm of the difference of the empirical and the optimal operators H_n and H, against the covariance C.

Hence, it is possible now to study the excess-risk, using linear algebra, and concentration inequality in Hilbert space.

Bias-variance decomposition. Defining the following "regularized" optimal estimator

$$h^*_\lambda(x) = H_\lambda \phi(x) \quad \text{with} \quad H_\lambda = HC(C+\lambda I)^{-1}, \tag{2.40}$$

one can decompose the excess-risk as

$$\sqrt{\mathcal{R}(\hat{h}) - \mathcal{R}(h^*)} \leq \underbrace{\mathbb{E}_x[\|\hat{h}(x) - h^*_\lambda(x)\|^2_{\mathcal{Y}}]^{1/2}}_{\text{variance}} + \underbrace{\mathbb{E}_x[\|h^*_\lambda(x) - h^*(x)\|^2_{\mathcal{Y}}]^{1/2}}_{\text{bias}}. \tag{2.41}$$

Variance bound. The first term can be bounded with high probability, using concentration inequality for random Hilbert-Schmidt operators, as

$$\mathbb{E}_x[\|\hat{h}(x) - h^*_\lambda(x)\|^2_{\mathcal{Y}}]^{1/2} = \|(H_n - H_\lambda)C^{1/2}\|_{\text{HS}} \tag{2.42}$$

$$\lesssim \|(C+\lambda I)^{-1/2}C^{1/2}\|_{\text{HS}}\, n^{-1/2} \tag{2.43}$$

the proofs of (Ciliberto et al., 2016), with minor changes. For the sake of clarity, we only keep the dominant term in n and λ (when $n \to +\infty$, $\lambda \to 0$). This "variance term" decreases when the regularization increases (i.e. when λ decreases).

This gives rise to the definition of the following *capacity condition*.

Capacity condition. The capacity condition (Caponnetto and De Vito, 2007) measures the regularity of the features $\phi(x)$, and can defined as

$$\|(C+\lambda I)^{-1/2}C^{1/2}\|_{\text{HS}} \lesssim \lambda^{-u} \tag{2.44}$$

with $u \in [0, 1/2]$. The capacity condition is always verified for $u = 1/2$ because $\|C\|_{\text{HS}} < +\infty$. The faster the eigenvalue decay rate of C the smaller u. This is related to how much the regularization facilitates the statistical problem by reducing it. Roughly interpreting CC^{-1}_λ as the projection on the main components of C with eigenvalues greater than λ, we see that one hopes to have a fast eigenvalues decay for C. This corroborates with the intuition that a small intrinsic input dimensions makes a statistical learning problem easier.

Bias bound. The second term is deterministic and can be bounded as

$$\mathbb{E}_x[\|h^*_\lambda(x) - h^*(x)\|^2_{\mathcal{Y}}]^{1/2} = \|(H_\lambda - H)C^{1/2}\|_{\text{HS}} \tag{2.45}$$

$$= \|H(C(C+\lambda I)^{-1} - I)C^{1/2}\|_{\text{HS}} \tag{2.46}$$

$$= \lambda \|H(C+\lambda I)^{-1}C^{1/2}\|_{\text{HS}} \tag{2.47}$$

with $\beta > 0$. This "bias term" decreases when the regularization decreases (i.e. λ increases).

This gives rise to the definition of the following *source condition*.

Source condition. The source condition (Caponnetto and De Vito, 2007) measures the regularity of the target map h^*, and can defined as

$$\|H(C+\lambda I)^{-1}C^{1/2}\|_{\text{HS}} \lesssim \lambda^{-v} \tag{2.48}$$

with $v \in [0, 1/2]$. The source condition is always verified for $v = 1/2$ because $\|H\|_{\mathrm{HS}} < +\infty$. The more the right eigenvectors of H are aligned with the eigenvectors of C the smaller v. This is related to how much the target h^* respect the regularity defined by the RKHS norm. Namely, how much the RKHS norm regularization induces bias when λ increases, making the RKHS norm regularization more or less relevant for the learning problem at hand.

Trade-off bias-variance. Finally, the best learning rate (dependency in the number of training data n) possible is obtained by taking the λ obtaining the best bias-variance trade-off, which is $\lambda = n^{-\frac{1}{2(1-v+u)}}$, leading to:

$$\mathbb{E}_x[\|\hat{h}(x) - h^*(x)\|_{\mathcal{Y}}^2] \lesssim n^{-\frac{1-v}{2(1-v+u)}}. \tag{2.49}$$

The stronger the capacity and the source condition, the faster the learning rate, from $n^{-1/4}$ to $n^{-1/2}$.

Sharpness of the learning rates. One may ask: Are the obtained bounds sharp? Can one find an estimator with better bounds? The optimally of the kernel ridge estimator and the rates given above have been proved in Caponnetto and De Vito (2007), in terms of minimax lower rates over the suitable class of priors.

2.5 Structured prediction

Structured prediction is the supervised prediction setting that interests us in this thesis. The goal of this section is to define this setting and highlight its challenges.

What is structured output prediction? The most studied settings of supervised learning deal with *high-dimensional inputs*, and predicts *low-dimensional outputs*, as for example, real numbers in the case of regression, and the values zero or one in the case of binary classification. In structured (output) prediction, one deals also with *high-dimensional outputs*. Examples of structured objects include sequences, graphs, sets, positive definite matrices, probability distributions, and permutations. A methodological characterization of structured prediction problems is that predicting the components of the outputs independently would be detrimental in terms of statistical performance.

Example of structured prediction problems. Being able to predict complex outputs makes it possible to address a much broader range of practical tasks. We give some examples.

- In computational biology: molecule structure prediction (Brouard et al., 2016a), find global alignments of related DNA strings, recognize functional portions of a genome.
- In natural language processing: handwritting recognition (LeCun et al., 2015), language translation (Bahdanau et al., 2015), part-of-speech tagging and parsing (Collins, 2002).

- In computer vision: image segmentation (Nowozin et al., 2011), reconstruction of images (Weston et al., 2003), and 3D human pose estimation (Li and Chan, 2014), and scene graph prediction (Chen et al., 2019).
- Learning to rank (Korba et al., 2018).
- Prediction of probability distributions (Frogner et al., 2015; Luise et al., 2018).
- Manifold regression (Steinke et al., 2010; Rudi et al., 2018).

Challenge of lacking a linear structure. Vector-valued regression methods leverage the linear structure of the output space. The setting of structured prediction can be characterized as the supervised setting where the output space does not benefit naturally from a linear structure (Ciliberto et al., 2020). Nevertheless, $\mathcal{Y}$ does verify another kind of non-linear structure. This has important implications when building supervised learning methods. In particular, it leads to serious challenges both in terms of modeling and computational complexity. An important consequence of the non-linear structure is that linear interpolation (computation of weighted averages) of objects in $\mathcal{Y}$ becomes not relevant. Finding an appropriate way of interpolating objects in $\mathcal{Y}$ is key to constructing an interpolant function $\hat{f} : \mathcal{X} \to \mathcal{Y}$, that is to choose the values of $\hat{f}$ out of the training set $(x_i, y_i)_{i=1}^n$. On the opposite, linear interpolation is relevant in the regression setting, and the classification setting (using one-hot encoded classes).

About the size of $\mathcal{Y}$. In structured prediction, the number of possible outputs $|\mathcal{Y}|$ is very big. Indeed, if $\mathcal{Y}$ is not infinite (e.g. in manifold regression), then $\mathcal{Y}$ size is exponential in the dimension d of $\mathcal{Y}$. For example, in the case of multi-label prediction $|\mathcal{Y}| = 2^d$, in the case of ranking $|\mathcal{Y}| = d!$. From statistical and computational perspectives, this observation about the size of $\mathcal{Y}$ in structured prediction, makes clear that one requires to make maximum use of the structure of the outputs in order to somehow reduce the dimension of the learning problem. On the opposite, one-hot encoding all the objects in $\mathcal{Y}$, as in classification, would correspond to not considering at all the structure of $\mathcal{Y}$. This would give very poor statistical performance, and high computational costs.

Let's illustrate the previous remarks on two subfamilies of structured prediction problems: graph prediction, and multi-label prediction.

Graph prediction. We build a toy target map $f^*(x) : [1, 5] \to \mathcal{Y}$ where $\mathcal{Y}$ is the space of labeled graphs[1] (see Figure 2.4). This example allows us to focus on the problem of handling the non-linear structure of the output space, as the input space is just one-dimensional. In this example, it is clear that linear interpolation of the representations adjacency/feature matrices $C \in \mathbb{R}^{N \times N}$ would not give a good estimation of the true map. Put another way, the map $x \to \|C(x))\|^2$ is not smooth (because there is no canonical ordering of the node), and so, standard regression methods would fail in learning such map, as relying on this smoothness assumption. A real-world application of graph prediction is the prediction of metabolites from mass spectra (see Figure 2.5). All methods proposed in this thesis will be tested on the metabolite identification problem.

[1] We consider a continuous version of the block stochastic model with x blocks. The values of f^* for not integer inputs are defined as linear interpolation between *aligned* adjacency and feature matrices.

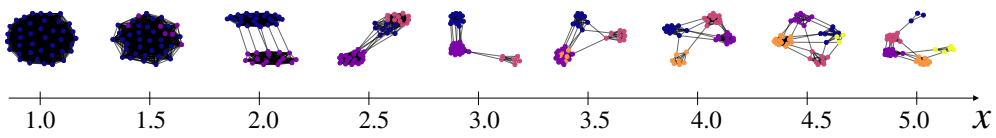

Figure 2.4: Graph prediction problem $f^* : \mathcal{X} = [1, 5] \to \mathcal{Y}$ where $\mathcal{Y}$ is the labeled graphs space. Plot of the values of $f^*(x)$ for $x \in \{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$.

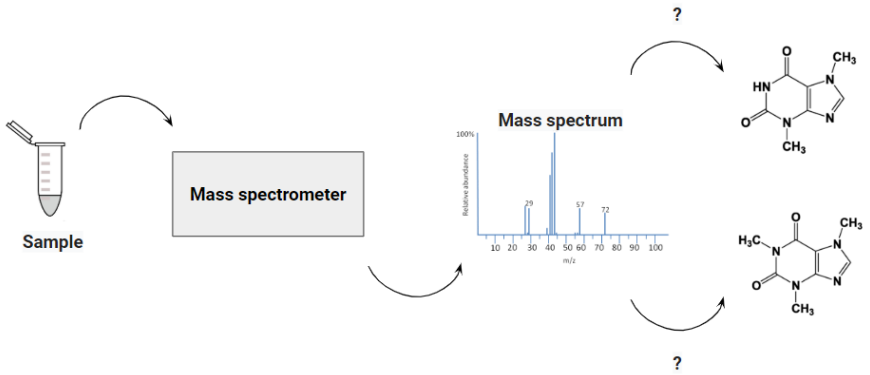

Figure 2.5: The metabolite identification problem. The goal is to be able to identify the structure of a molecule from a mass spectrum. Inputs $\mathcal{X}$ are mass spectra, outputs $\mathcal{Y}$ are molecular graphs of metabolites. A training set of couples (mass spectrum, molecule's structure) is available. Outputs are graphs with atoms as nodes.

Example of multi-label prediction. In this setting $\mathcal{Y} = \{0, 1\}^d$ with $d \in \mathbb{N}^*$. If d is not too big in comparison to the quantity of data n, one may just predict each of the d labels independently, which corresponds to linearly interpolation in $\mathcal{Y}$. When d increases, one needs to use the potential correlations between labels, e.g. by estimating a subset $S \subset \mathcal{Y}$ such that $f^*(x) \in S$ for any $x \in \mathcal{X}$. This is the goal of the methods proposed in Chapter 4 and Chapter 5.

The pre-image problem in structured Prediction. Structured predictions models write as $f(x; W) = \arg\min_{y \in \mathcal{Y}} g(x, y; W)$. Without assumptions on g the computational cost of one evaluation $f(x; W)$ is $\mathcal{O}(|\mathcal{Y}|)$ if $|\mathcal{Y}| < +\infty$, and can not be computed exactly, nor approximately, if $|\mathcal{Y}| = +\infty$. This inference step in structured prediction is called *decoding or pre-image computation*. Existing structured prediction algorithms comes with approximation methods to compute efficiently the pre-image (Nowozin et al., 2011). This is possible by exploiting specific structures of g. One contribution of this thesis is to provide methods to compute efficiently the pre-image in the case of surrogate methods.

2.6 Overview of structured prediction methods

The goal of this section is to give a brief background on existing methods for structured prediction. We refer the reader to Nowozin et al. (2011) and Bakir et al. (2007) for more details. Structured prediction methods can be presented in three main categories: Conditional Random Fields, Structured SVMs, and Surrogate methods. We

present these three families of methods in the following Sections 2.6.1, 2.6.2, and 2.6.3.

2.6.1 Conditional random fields

CRFs generalize logistic regression classifiers to structured output prediction (Lafferty et al., 2001).

Model. The conditional probability $p(y|x)$ is modeled choosing a parameterized graphical model:

$$p_w(y|x) = \frac{1}{Z(x,w)} \exp(-E_w(x,y)), \quad \text{with} \quad w \in \mathbb{R}^d \tag{2.50}$$

where $Z(x,w)$ is defined to ensure $\sum_{y\in\mathcal{Y}} p_w(y|x) = 1$. For example, one can choose $E_w(x,y) = \langle w, \phi(x,y)\rangle$ with ϕ a joint embedding map. For a given input x, a prediction y is computed using the estimated $p_w(y|x)$, e.g. through maximum a posteriori (MAP) inference $\hat{y} = \arg\max_y p_w(y|x)$, or using $\hat{y}_i = \arg\max_{y_i} p_w(y_i|x)$.

Training. Then, w is estimated to make $p_w(y|x)$ close to the true distribution $p(y|x)$ by maximizing the regularized conditional log-likelihood, for $\lambda > 0$:

$$w_n = \arg\min_{w\in\mathbb{R}^d} \sum_{i=1}^{n} \langle w, \phi(x_i, y_i)\rangle + \sum_{i=1}^{n} Z(x_i, w) + \lambda\|w\|^2. \tag{2.51}$$

The optimization problem defined in Equation (2.51) is a smooth convex optimization problem. It is solved using gradient descent.

A difficulty comes from the NP-hard computations of the normalization term Z for general graphical models: as this is a sum over $\mathcal{Y}$ whose size is exponential in the dimension of the outputs $y \in \mathcal{Y}$. As a result, inference and training are NP-hard for CRFs. Nevertheless, by exploiting the structure of the graphical model, it is possible to obtain computationally efficient approximate inference and training algorithms (e.g. using the belief propagation algorithm (Nowozin et al., 2011)).

2.6.2 Structured support vector machines

SSVMs generalize SVM classifiers to structured output prediction (Tsochantaridis et al., 2005; Taskar et al., 2005).

Model. The prediction function is modeled as:

$$f(x) = \arg\min_{y\in\mathcal{Y}} g(x,y,w) \tag{2.52}$$

where g is called energy function in the literature of energy-based methods (LeCun et al., 2006; Belanger and McCallum, 2016). Equivalently, the function $s(x,y,w) = -E(x,y,w)$ can be called score function, or compatibility function. For example, one can choose $g(x,y,w) = \langle w, \phi(x,y)\rangle$ (Tsochantaridis et al., 2005), or used a neural network for g (Belanger and McCallum, 2016; Belanger et al., 2017). An interesting property of SSVMs is that they can be kernelized as standard SVMs.

Training. Then, w is estimated to make $f(x)$ close to the true output y associated to the input x. To do so, for a given loss Δ a straightforward idea is to perform the following ERM, with $\lambda > 0$:

$$w_n = \operatorname*{arg\,min}_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \Delta(f(x_i), y_i) + \lambda \|w\|^2. \tag{2.53}$$

Nevertheless, solving this optimization problem with gradient descent is not possible as $\Delta(f(x_i), y_i)$ is piece-wise constant with respect to w. Therefore, SVMs consider instead minimizing a convex upper bound

$$w_n = \operatorname*{arg\,min}_{w \in \mathbb{R}^d} \sum_{i=1}^{n} l(x_i, y_i, w) + \lambda \|w\|^2 \tag{2.54}$$

with $l(x, y, w) = \max_{y'} \Delta(y, y') - g(x, y, w) + g(x, y', w) \geq \Delta(f(x), y)$. Notice that l generalizes the Hinge loss for binary classification to structured output space.

Similarly to CRFs, the size of $\mathcal{Y}$ creates computational difficulties when aiming to compute w_n (because of $\max_{y \in \mathcal{Y}}$). Various algorithms have been proposed to compute w_n efficiently. We refer the reader to Nowozin et al. (2011) for more details.

Max-margin Markov ($\mathbf{M^3}$) networks. M^3 networks (Taskar et al., 2003) is a family of structured prediction methods that can be considered as a combination of SSVMs and CRFs. Indeed g is defined as a graphical model, but then g is trained via a max margin-based optimization problem. Knowing such structure on g allows obtaining more efficient training procedures.

2.6.3 Structured prediction with least-squares surrogate regression

This family of methods generalizes least-squares surrogate classifier to structured output prediction (Weston et al., 2003; Cortes et al., 2005; Brouard et al., 2016b; Ciliberto et al., 2020). In this section, we start by presenting the least-squares surrogate method proposed in Brouard et al. (2016b), then we present the one proposed in Ciliberto et al. (2016, 2020).

Surrogate methods consist in transforming the structured prediction problems into a surrogate regression problem by embedding the output in a Hilbert space $\mathcal{H}_y$ thanks to an embedding map $\psi : \mathcal{Y} \to \mathcal{H}_y$, then using standard methods for regression. In particular, *least-squares surrogate methods* consider solving the following least-squares surrogate regression problem:

$$h^* = \operatorname*{arg\,min}_{h} \mathbb{E}[\|h(x) - \psi(y)\|^2]. \tag{2.55}$$

Then, a structured predictor $\hat{f} : \mathcal{X} \to \mathcal{Y}$ is built from an estimate $\hat{h}$ of h^* using a decoding function $d : \mathcal{H}_y \to \mathcal{Y}$ as:

$$\hat{f} = d \circ \hat{h}. \tag{2.56}$$

One can illustrate the construction of surrogate methods as in Figure 2.6.

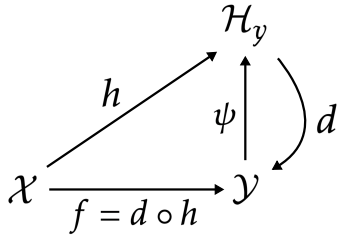

Figure 2.6: Surrogate methods for structured prediction.

Output Kernel Regression (OKR). Weston et al. (2003); Cortes et al. (2005); Brouard et al. (2016b) propose to define $\psi : \mathcal{Y} \to \mathcal{H}_y$ as the canonical map associated to a positive definite kernel k_y on $\mathcal{Y}$. Such kernel should be chosen with respect to the available information on the geometry of $\mathcal{Y}$.

Kernels on input spaces $\mathcal{X}$ are used to go from linear models to non-linear models, while keeping a linear parametrization of the hypothesis space. A kernel on the output space $\mathcal{Y}$ is used to go from euclidean loss to non-linear loss, while still solving a least-squares regression problem.

Even if $\psi(y)$ is infinite dimensional, by using kernel ridge regression with an operator-valued kernel (Brouard et al., 2016b), it is still possible to build estimator $\hat{h}$ as

$$\hat{h}(x) = \sum_{i=1}^{n} \alpha_i(x)\psi(y_i), \quad \text{with } \alpha(x) = (K + n\lambda)^{-1} k_x(x) \tag{2.57}$$

where $K = (k(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n\times n}$, $k_x(x) = (k_x(x, x_i))_{i=1}^n \in \mathbb{R}^n$, k_x is a kernel over $\mathcal{X}$. At training time only $M = (K + n\lambda)^{-1}\mathbb{R}^{n\times n}$ can be computed.

Then, Brouard et al. (2016b) propose to obtain a structured estimator via

$$\hat{f}(x) = \operatorname*{arg\,min}_{y\in\mathcal{Y}} \|\hat{h}(x) - \psi(y)\|^2_{\mathcal{H}_y}. \tag{2.58}$$

Even if $\psi(y)$ is infinite dimensional, the $\|\hat{h}(x) - \psi(y)\|^2_{\mathcal{H}_y}$ can be computed because one is able to compute the scalar products $\langle \psi(y_i), \psi(y)\rangle = k_y(y_i, y)$:

$$\operatorname*{arg\,min}_{y} \|\hat{h}(x) - \psi(y)\|^2_{\mathcal{H}_y} = \operatorname*{arg\,min}_{y} k_y(y, y) - 2\alpha(x)^T k_y(y) \tag{2.59}$$

with $k_y(y) = (k_y(y, y_i))_{i=1}^n \in \mathbb{R}^n$.

Remark 2.1 (Beyond least-squares)**.** *In this thesis, we only consider least-squares problems as surrogate regression problems, but one may consider other kinds of surrogate problems (Brouard et al., 2016b; Nowak-Vila et al., 2019; Laforgue et al., 2020). In particular, Brouard et al. (2016b) proposes to use other convex losses than the squared norm, as the hinge loss.*

Implicit Loss Embeddings (ILE) . Ciliberto et al. (2016, 2020) generalizes the method to a wide variety of losses as follows.

First, Ciliberto et al. (2020) proves that most losses admit an *Implicit Loss Embedding*, that is it can be written in the following form:

$$\Delta(y, y') = \langle \chi(y), \psi(y')\rangle_{\mathcal{H}_y} \tag{2.60}$$

where $\mathcal{H}_y$ is a Hilbert space, and $\chi : \mathcal{Y} \to \mathcal{H}_y$, $\psi : \mathcal{Y} \to \mathcal{H}_y$ are two bounded maps.

Characterization of f^*. Then, from the characterization of f^* (see Chapter 2), the following holds:

$$f^*(x) = \arg\min_{\hat{y} \in \mathcal{Y}} \mathbb{E}_{y|x}[\Delta(\hat{y}, y)] \tag{2.61}$$

$$= \arg\min_{\hat{y} \in \mathcal{Y}} \mathbb{E}_{y|x}[\langle \chi(\hat{y}), \psi(y) \rangle_{\mathcal{H}_y}] \tag{2.62}$$

$$= \arg\min_{\hat{y} \in \mathcal{Y}} \langle \chi(\hat{y}), \mathbb{E}_{y|x}[\psi(y)] \rangle_{\mathcal{H}_y} \tag{2.63}$$

$$= \arg\min_{\hat{y} \in \mathcal{Y}} \langle \chi(\hat{y}), h^*(x) \rangle_{\mathcal{H}_y} \tag{2.64}$$

where $h^*(x) = \mathbb{E}_{y|x}[\psi(y)]$ is the solution to the LS problem of predicting $\psi(y)$ from x, using the characterization of the solutions of LS problems.

From there, they naturally proposed the following structured estimator.

Proposed structured predictor. Ciliberto et al. (2020) proposed the estimator

$$\hat{f}(x) = \arg\min_{\hat{y} \in \mathcal{Y}} \langle \chi(\hat{y}), \hat{h}(x) \rangle_{\mathcal{H}_y} \tag{2.65}$$

$$= \arg\min_{\hat{y} \in \mathcal{Y}} \langle \chi(\hat{y}), \sum_{i=1}^{n} \alpha_i(x) \psi(y_i) \rangle_{\mathcal{H}_y} \tag{2.66}$$

$$= \arg\min_{y \in \mathcal{Y}} \sum_{i=1}^{n} \alpha_i(x) \Delta(\hat{y}, y_i) \tag{2.67}$$

where $\alpha(x)$ is the weight function of a kernel ridge estimator $\hat{h}$, or the weights function of other regression estimators such as $\hat{h}(x) = \sum_{i=1}^{n} \alpha_i(x) \psi(y_i)$ solves the LS regression problem of predicting $\psi(y)$ from x.

The ILE estimator can be understood as an estimate of f^* that is defined from the characterization of f^* and estimating the scalar-valued maps $x \to \mathbb{E}_{y|x}[\Delta(y, \hat{y})]$ for any $\hat{y} \in \mathcal{Y}$.

2.7 Theoretical guarantees for least-squares surrogates

This section focuses on theoretical guarantees for least-squares surrogate structured prediction estimators. As we saw in the previous section, one can prove learning rates for the KRR estimator. Can we provide similar theoretical guarantees for LS surrogate structured prediction estimators? We present here the main idea of the analysis of Ciliberto et al. (2020). For the theoretical study of other structured prediction methods such as SSVMs, CRFs, and M^{3}N, we refer the reader to Nowak-Vila et al. (2019); Nowak et al. (2020).

The construction of the ILE estimator, presented in the previous section, is based on the implicit estimation of h^* using $\hat{h}$. Hence, intuitively, the quality of the estimator $\hat{f}$ should relate to the quality of the estimator $\hat{h}$. The following result makes it clear.

Comparison inequality (Ciliberto et al., 2020). The estimation $\hat{f}$ relates to the estimation $\hat{h}$ through the following inequality:

$$\mathcal{R}_\Delta(\hat{f}) - \mathcal{R}_\Delta(f^*) \leq c_\chi \, \mathbb{E}[\|\hat{h}(x) - h^*(x)\|^2_{\mathcal{H}_y}]^{1/2} \tag{2.68}$$

with $c_\chi = \sup_{y \in \mathcal{Y}} \|\chi(y)\|_{\mathcal{H}_y}$.

Sketch of the proof. The proof is based on the fact that $h^*(x) = \mathbb{E}_{y|x}[\psi(y)]$, the linearity of the inner product, the Jensen inequality, and the inequality $|\inf_{y \in \mathcal{Y}} u(y) - \inf_{y \in \mathcal{Y}} v(y)| \leq \sup_{y \in \mathcal{Y}} |u(y) - v(y)|$ for any functions $u, v : \mathcal{Y} \to \mathbb{R}$.

Therefore, obtaining excess-risk bounds for the structured estimator $\hat{f}$ can be done by deriving excess-risk bounds for the regression estimator $\hat{h}$. For instance, defining the weight function α from kernel ridge weights, we can use the learning bounds of Section 2.4.3, and we obtain

$$\mathcal{R}_\Delta(\hat{f}) - \mathcal{R}_\Delta(f^*) \lesssim c_\chi n^{-1/4}. \tag{2.69}$$

Obtaining learning bounds for OKR. When choosing the loss $\Delta(y, y') = \|\psi(y) - \psi(y')\|^2_{\mathcal{H}_y}$ to define the ILE estimator, the OKR and the ILE approaches differ as follows

$$\hat{f}_{OKR}(x) = \arg\min_y \|\hat{h}(x) - \psi(y)\|^2_{\mathcal{H}_y} \tag{2.70}$$

$$= \arg\min_y \|\psi(y)\|^2_{\mathcal{H}_y} - 2\langle \hat{h}(x), \psi(y) \rangle_{\mathcal{H}_y}, \tag{2.71}$$

and

$$\hat{f}_{ILE}(x) = \arg\min_y \sum_i \alpha_i(x) \|\psi(y_i) - \psi(y)\|^2_{\mathcal{H}_y} \tag{2.72}$$

$$= \arg\min_y \left(\sum_i \alpha_i(x) \right) \|\psi(y)\|^2_{\mathcal{H}_y} - 2\langle \hat{h}(x), \psi(y) \rangle_{\mathcal{H}_y}. \tag{2.73}$$

That is the estimators differ because of the multiplicative constant $\sum_i \alpha_i(x)$ which can be understood as the ridge estimator of the regression problem with input x and constant output 1. Additional derivations allow to show that $\hat{f}_{OKR}$ benefits from the same guarantees than $\hat{f}_{ILE}(x)$ (using the inequality $|\inf_{y \in \mathcal{Y}} u(y) - \inf_{y \in \mathcal{Y}} v(y)| \leq \sup_{y \in \mathcal{Y}} |u(y) - v(y)|$ for any functions $u, v : \mathcal{Y} \to \mathbb{R}$, and the bound $\mathbb{E}_x[|\sum_i \alpha_i(x) - 1|] \leq \mathbb{E}_x[(\sum_i \alpha_i(x) - 1)^2]^{1/2}$).

2.8 On the role of the output structure in structured prediction

This section aims at discussing, through the lens of surrogate methods, the role of the structure of the output space when solving supervised learning problems. We will point out the importance of exploiting a priori information about the output structure for obtaining statistical and computational efficiency. Then, in light of this discussion, we will comment on the contributions presented in this manuscript.

Let us start by recalling the following definition, proposed in the introduction.

Structured space. A structured space $(\mathcal{Y}, \psi)$ is a set equipped with an embedding map taking values in a Hilbert space. As shown above, ψ can be chosen explicitly, or implicitly via the choice of a kernel, or a loss.

Questions. We would like to answer the following questions: What makes a structured problem harder to solve than a regression or a classification one? How the difficulty of supervised learning is related to the output space? What makes a structured prediction learnable? From the above definition of structured space, we can properly formulate these questions as: *How does the output structured space $(\mathcal{Y}, \psi)$ affect the learning bounds? Which assumptions on $(\mathcal{Y}, \psi)$ have been made to obtain these bounds?*

Before giving the learning bounds, we need to introduce the following operator.

Output variance or noise. We define the following covariance operator:

$$E = \mathbb{E}[\epsilon \otimes \epsilon] \tag{2.74}$$

with $\epsilon = \psi(y) - h^*(x) = \psi(y) - \mathbb{E}_{y|x}[\psi(y)]$.

Assuming that $\epsilon \neq 0$ allows us to consider problems where there is variability in the output y given the input x. This allows modeling labeling mistakes occurring when building data sets, or modeling the variability arising from omitted explanatory variables. Notice that this setting is especially relevant when one deals with high-dimensional outputs.

Learning bounds. From the comparison inequality Eq. (2.68), and keeping track of the dependency in $(\mathcal{Y}, \psi)$ when deriving the learning bounds for the KRR estimator, following the sketch of proofs given in Section 2.4.3, one obtains the following learning bounds. The structured predictor $\hat{f}$ verifies with high probability:

$$\mathcal{R}_\Delta(\hat{f}) - \mathcal{R}_\Delta(f^*) \lesssim c_\chi \quad \|H\|_{\mathrm{HS}}^{1/2} \quad \|E^{1/2}\|_{\mathrm{HS}}^{1/2} \quad n^{-\frac{1-v}{2(1-v+u)}} \tag{2.75}$$

keeping only the dominant terms in n (when $n \to +\infty$).

Notice that the inequality is dimensionally homogeneous. In the following, we comment on the dependency of the bound in ψ by structuring it in two insights (A) and (B).

(A) Intrinsic output dimension of the learning problem. The bound depends on the total output variance

$$\mathrm{Tr}(E) = \mathbb{E}[\|\epsilon\|^2] = \mathbb{E}[\|\psi(y)\|^2] - \mathbb{E}[\|h^*(x)\|^2] \leq \sup_y \|\psi(y)\|^2 := c_\psi^2. \tag{2.76}$$

Moreover, the bound also depends on the target's RKHS norm $\|H\|_{\mathrm{HS}}$, this term is studied in Nowak et al. (2019), and bounded as $\|H\|_{\mathrm{HS}}^2 \lesssim r$ where $r \in \mathbb{N}^*$ is called the *affine dimension*. r can be understood as the *intrinsic output dimension* of the learning problem, and can also be intuitively thought of as the "number of scalar values to be predicted".

Remark 2.2 (Non-unicity of the ILE representation)**.** *When considering ψ implicitly induced by a kernel or a loss, there is non-unicity of ψ for a given kernel or loss. The bound applies for all valid ψ. Hence, one may ask, for a given loss, which valid embedding leads to*

the best affine dimension. (Nowak et al., 2019) prove sharp affine dimension for the most standard losses. In particular, it is shown that for the 0-1 loss and the Hamming loss, the constants $r = 2^d$ and $r = d$ above are optimal.

(B) Regularity of f^*. The bounds have been obtained under two conditions: the capacity and source conditions. The capacity condition measures the regularity of the input features $\phi(x)$, and does not depend on ψ. The source condition does depend on ψ. Indeed, it quantities the regularity of the target $h^*(x) = \mathbb{E}_{y|x}[\psi(y)]$ (in the sense explained in Section 2.4.3). Now, considering the noiseless setting $y = f^*(x)$, getting rid of the problem of dealing with the output variance, this simply implies regularity of the map

$$x \to \psi(f^*(x)) \tag{2.77}$$

and also consequently of all the maps

$$x \to \Delta(\hat{y}, f^*(x)) \quad \text{for} \quad \hat{y} \in \mathcal{Y}. \tag{2.78}$$

That is, by choosing ψ, one should arrange the points in $\mathcal{Y}$ at certain distances from each other such that the map $x \to \mathbb{E}_{y|x}[\psi(y)]$ ($= x \to \psi(f^*(x))$ in the noiseless setting) respects the regularity of the chosen hypothesis space. Notice that the minimum source condition ($v = 1/2$) corresponds to the assumption that h^* belongs to the hypothesis space, making the problem at hand learnable, as the target belongs to a PAC learnable class of functions.

Example of $\mathcal{Y} = \{0,1\}^d$. Let's consider, for instance, the case of multi-label classification, i.e. $\mathcal{Y} = \{0,1\}^d$. When choosing the 0-1 loss, it leads to an affine dimension $r = |\mathcal{Y}| = 2^d$. Intuitively, this stems from the fact that it corresponds to solving a multiclass classification problem with $|\mathcal{Y}|$ classes, by choosing $\psi(y) = (\mathbb{1}_{y=y'})_{y' \in \mathcal{Y}}$. All the points in $\mathcal{Y}$ are arranged at equal distances. Even if the map $x \to \mathbb{E}_{y|x}[\psi(y)] = (p(y|x))_{y \in \mathcal{Y}}$ may strongly verify the source condition, when d increases the multiplicative constant increases exponentially making the problem quickly statistically untractable (Osokin et al., 2017). If one uses instead the Hamming loss $\Delta(y, y') = \sum_{j=1}^{d} \mathbb{1}_{y_j \neq y'_j}$, it leads to $r = d$. Intuitively, this stems from the fact that it corresponds to choosing $\psi(y) = (2\mathbb{1}_{y_j = y'_j} - 1)_{j=1}^{d}$, and to solve d binary classification problems. In this case, the points in $\mathcal{Y}$ are arranged at distances corresponding to the quantity of differing labels, providing a more informative geometry. The regularity condition is on $x \to \mathbb{E}_{y|x}[\psi(y)] = (2p(y_j|x) - 1)_{j=1}^{d}$.

Take-home message. The discussion above points out the importance in structured prediction of exploiting as much *a priori information on the geometry of* $\mathcal{Y}$ as possible, for obtaining good learnability, i.e. fast learning rate, and not too large constants in the bounds. In particular, it is preferable to use representations $\psi(y)$ with low intrinsic dimension (A), and whose values can be efficiently interpolated from the input space thanks to the chosen hypothesis space (B).

(I) A general output geometry for graph prediction. In practice, finding an embedding ψ that satisfies well the criteria described above, is not always obvious. The main proposal of Chapter 3 is to consider the Gromov-Wasserstein distance as a loss

(implicitly defining ψ) in the case of graph prediction. We discuss this choice in light of the criteria raised above.

If $f^* : x \to C(x)$ is a map taking values in the space of adjacency matrices, because there is no canonical ordering of the nodes, the following maps are not likely to be regular:

$$x \to \|\hat{y} - f^*(x)\|_2^2 \quad \text{for} \quad \hat{y} \in \mathcal{Y}. \tag{2.79}$$

The Gromov-Wasserstein distance (Peyré et al., 2016) can be thought of as the euclidean distance on adjacency matrices after "realignment of the nodes", in the hope of obtaining regularity of the following maps (criterion (A))

$$x \to \mathrm{GW}(\hat{y}, f^*(x)) \quad \text{for} \quad \hat{y} \in \mathcal{Y}. \tag{2.80}$$

The neural-network version of the method proposed in this work offers a way of controlling the expressiveness of the model with respect to the output space. This can be intuitively understood as aiming to control the output dimension of the learning problem (criterion (B)).

Now, let's consider the output embedding ψ fixed. Chapters 4 and 5 present two methods exploiting the output structure in order to obtain a computational and statistical gain. The statistical gain results from a reduction of the output variance, namely a reduction of the constant $\|E^{1/2}\|_{\mathrm{HS}}$.

(II) Reduced-rank regression for non-linear output space (Chapter 4). Coming back to the example of multi-label classification and the Hamming loss, the least-squares surrogate estimator corresponds to predicting each label with an independent least-squares surrogate binary classifier. If the labels are linearly dependent it is a good idea to perform reduced-rank regression. Indeed, projecting the $\hat{h}(x)$ on the main components of $h^*(x)$, will lead to a small bias, while can allow to substantially reduce the output variance $\epsilon = \psi(y) - h^*(x)$. The contribution of Chapter 4 can be understood as a generalization of reduced-rank regression for non-linear output space. $\mathcal{Y}$ is not a linear space but the embedding map provides a linear structure to $\mathcal{Y}$ by embedding it in the Hilbert space $\mathcal{H}_y$. Hence, we propose to perform reduced-rank regression of ψ. We will prove that reduced-rank regression allows indeed to significantly reduce the constant $\|E^{1/2}\|_{\mathrm{HS}}$, under output regularity conditions that we will refer to as output capacity condition, and output source condition. Notice that we will consider the general case where $\mathcal{H}_y$ can be infinite dimensional. Finally, the resulting structured predictor will benefit from the same statistical gain, and also from an alleviation of the pre-image computational complexity.

(III) Calibrated structured prediction with loss regularization. (Chapter 5). The method proposed in Chapter 4, exploits the low-rank structure of the $h^*_\psi(x) = \mathbb{E}_{y|x}[\psi(y)]$, and is shown to be calibrated with the loss $\Delta(y, y') = \|\psi(y) - \psi(y')\|^2_{\mathcal{H}_y}$. The contribution of Chapter 5 is to propose a method that separates the target embedding $\psi : \mathcal{Y} \mapsto \mathcal{H}_y$ from an alternate embedding $\tilde{\psi} : \mathcal{Y} \mapsto \tilde{\mathcal{H}}_y$ whose structure is exploited. More precisely, we propose to exploit the low-rank structure of the LS solution $h^*_{\tilde{\psi}}(x) = \mathbb{E}_{y|x}[\tilde{\psi}(y)]$, but we keep the regression estimator calibrated with the estimation of $h^*_\psi(x) = \mathbb{E}_{y|x}[\psi(y)]$. This allows for instance to exploit the structure provided by a Gaussian kernel over $\mathcal{Y}$, while being calibrated with the euclidean loss $\Delta(y, y') = \|y - y'\|^2$ when $\mathcal{Y} = \mathbb{R}^d$, or

the geodesic loss over when $\mathcal{Y}$ is a manifold. Interestingly, the proposed structured estimator can be thought of as the ILE estimator but with a regularized loss Δ with respect to the regularity defined by $\tilde{\psi}$. This leads to the following intuitive interpretation of the statistical gain: one had better make more or less fine-grained predictions, depending on the quantity of training data, in order to deal with the output variance (or noise).

3

Learning to Predict Graphs with Fused Gromov-Wasserstein Barycenters

Contents

3.1 Introduction

Graphs allow to represent entities and their interactions. They are ubiquitous in real-world: social networks, molecular structures, biological protein-protein networks, recommender systems, are naturally represented as graphs. Nevertheless, graphs structured data can be challenging to process. An important effort has been made to design well-tailored machine learning methods for graphs. For example, many kernels for graphs have been proposed allowing to perform graph classification, graph clustering, graph regression (Kriege et al., 2020). Many deep learning architecture have also been developed (Zhang et al., 2022), including Graph Convolutional Networks (GCNs) that are powerful models for learning with graphs as inputs. Most of these existing works in machine learning consider graphs as inputs, but predicting a graph as output given an input has received much less attention.

In this work, we consider the difficult problem of supervised learning of graph-valued functions. Some works address this learning problem in various settings. Gómez-Bombarelli et al. (2018) try to obtain a continuous representation of molecules using a variational autoencoding (VAE) of text representations of molecules (SMILES). Kusner et al. (2017) incorporates in the VAE architecture knowledge about the structure of SMILES thanks to its available grammar. Li et al. (2018); Olivecrona et al. (2017); Liu et al. (2017); You et al. (2018); Shi et al. (2020) propose models that generate graphs using a sequential process generating one node/edge at a time, and train it by maximizing the likelihood.

In this work we consider the supervised graph prediction problem as a structured prediction problem. The abundant literature on the topic of structured prediction has mainly explored three directions: energy-based models, surrogate approaches and end-to-end learning. In energy-based models (Tsochantaridis et al., 2005; Chen et al., 2015; Belanger and McCallum, 2016), predictions are obtained by maximizing a score function for input-output pairs over the output space. In surrogate approaches (Cortes et al., 2005; Geurts et al., 2006; Brouard et al., 2016b; Ciliberto et al., 2016), a feature map is used to embed the structured outputs. After minimizing a surrogate loss a decoding procedure is used to map back the surrogate solution. End-to-end learning attempt to solve structured prediction by directly learning to generate a structured object (Belanger et al., 2017; Silver et al., 2017) and leverage differentiable and relaxed definition of energy-based methods (see for instance Pillutla et al. (2018); Mensch and Blondel (2018)). In the case of supervised graph prediction, major challenges come from the fact that the number of possible outputs can be extremely large and that the graphs have generally different sizes. Finding a good loss and output representation is therefore particularly crucial. Typical graph representations usually rely on graph kernels leveraging fingerprint representation, i.e. a bag of motifs approach (Ralaivola et al., 2005) or more involved kernels such the Weisfeiler-Lehman kernel (Shervashidze et al., 2011). In this work, we propose to exploit another kind of graph representation, opening the door to the use of an Optimal Transport loss, and derive an end-to-end learning approach that constrasts to energy-based learning and surrogate methods.

Successful applications of optimal transport (OT) in machine learning are becoming increasingly numerous thanks to the advent of numerical optimal transport (Cuturi, 2013; Peyré and Cuturi, 2019; Altschuler et al., 2017). Examples include domain adaptation (Courty et al., 2016), unsupervised learning (Arjovsky et al., 2017), multi-label classification (Frogner et al., 2015), natural language processing (Kusner et al., 2015), fair classification (Gordaliza et al., 2019), supervised representation learning (Flamary et al., 2018). Optimal transport provide meaningful distances between probability distributions, by leveraging the geometry of the underlying metric spaces.
Supervised learning with optimal transport losses has been considered in Frogner et al. (2015); Bonneel et al. (2016); Luise et al. (2018); Mensch et al. (2019) for predicting histograms. But traditional OT loss can be applied only between distributions lying in the same space, preventing their use on structured data such as graphs. Mémoli (2011) proposed the Gromov-Wasserstein distance that can measure similarity between metric measure space and has been used as a distance between graphs in several applications such as computing graph barycenters (Peyré et al., 2016) or for performing graph node embedding (Xu et al., 2019b) and graph partitioning (Xu et al., 2019a). This distance has been extended to the Fused Gromov-Wasserstein distance (FGW) in Vayer et al. (2019, 2020) with applications to attributed graphs classification, barycenter estimation and more recently dictionary learning (Vincent-Cuaz et al., 2021). Those novel divergences that can be used on graphs are a natural fit, first as a loss term in graph prediction but also as a way to model the space of graphs for instance using FGW barycenters.

Contributions. In this dissertation we present the following novel contributions. First we propose a novel and and general framework in Sec. 3.3 for graph prediction building on FGW as a loss and FGW barycenter as a way to interpolate in the target space. The framework is studied theoretically in Sec. 3.4 in the non-parametric case

for which we provide consistency and excess risk bounds. Then a parametric version of the model building on deep neural network and learning of the template graphs is proposed in Sec. 3.5 with a simple stochastic gradient algorithm. Finally we provide some numerical experiments in Sec. 3.6 on synthetic and real life metabolite prediction datasets.

3.2 Background on OT for graphs

We begin by introducing how to represent graphs and define distances between graph by leveraging the Fused Gromov-Wasserstein distance.

Notations. $\mathbb{1}_p$ is the all-ones vector with size p. δ_x denotes the Dirac measure in x for x in a measurable space. Identity matrix in $\mathbb{R}^{p\times p}$ is noted I_p. $\mathcal{L}(\mathcal{A})$ the set of bounded linear operator from $\mathcal{A}$ to $\mathcal{A}$. $\mathcal{M}(\mathcal{A},\mathcal{B})$ the set of measurable functions from $\mathcal{A}$ to $\mathcal{B}$.

Graph represented as metric measure spaces. Denote $p_{max} \in \mathbb{N}^*$ the maximal number of nodes (vertices) in the graphs we consider in this dissertation. We define $\mathcal{F} \subset \mathbb{R}^d$ a finite feature space of size $|\mathcal{F}| < \infty$. A labeled graph y of $p \leq p_{max}$ nodes is represented by a triplet $y = (C, F, h)$ where $C = C^T \in \{0,1\}^{p\times p}$ is the adjacency matrix, and $F = (F_i)_{i=1}^p$ is a p-tuple composed of feature vectors $F_i \in \mathcal{F} \subset \mathbb{R}^d$ labeling each node indexed by i. The space of labeled graphs is thus defined as $\mathcal{Y}_{dis} = \{(C,F,h) | p \leq p_{max}, C \in \{0,1\}^{p\times p}, C^T = C, F = (F_i)_{i=1}^p \in \mathcal{F}^p, h = \frac{1}{p}\mathbb{1}_p\}$. Observe that we equipped all graphs with a uniform discrete probability distributions over the nodes $\mu = \sum_{i=1}^p h_i \delta_{u_i}$ where $u_i = (v_i, F_i)$ represents the structure v_i (encoded only through $C(i,j), \forall j$) and the feature information F_j attached to a vertex i (Vayer et al., 2019). These weights indicate the relative importance of the vertices in the graph. In absence of this information, we simply fix uniform weights $h_i = \frac{1}{p}$ for a graph of size p. Now, let us introduce the space of continuous relaxed graphs with **fixed size** p: $\mathcal{Y}_p = \{(C,F,h) | C \in [0,1]^{p\times p}, C^T = C, F \in \mathrm{Conv}(\mathcal{F})^p\}, h = p^{-1}\mathbb{1}_p\}$. $\mathrm{Conv}(\mathcal{F})$ denotes the convex hull of $\mathcal{F}$ in $\mathbb{R}^{p\times d}$. We call $\mathcal{Y} = \bigcup (\mathcal{Y}_i)_{i=1}^{p_{max}}$ and want to emphasize that $\mathcal{Y}_{dis} \subset \mathcal{Y}$.

Gromov-Wasserstein (GW) distance. The Gromov-Wassertein distance between metric measure space has been introduced by Mémoli (2011) for object matching. The GW distance defines an OT problem to compare these objects, with the key property that it defines a strict metric on the collection of isomorphism classes of metric measure spaces. In this dissertation, we adopt this angle to address graph representation and graph comparison, opening the door to define a loss for supervised graph prediction. Let $y_1 = (C_1, p_1^{-1}\mathbb{1}_{p_1})$ and $y_2 = (C_2, p_2^{-1}\mathbb{1}_{p_2})$ be the representation of two graphs with respectively $p_1 \in \mathbb{N}^*$ and $p_2 \in \mathbb{N}^*$ nodes, the Gromov-Wasserstein (GW) distance between y_1 and y_2, is defined as follows:

$$\mathrm{GW}_2^2(y_1, y_2) = \min_{\pi \in \mathcal{P}_{p_1,p_2}} \sum_{i,k=1}^{p_1} \sum_{j,l=1}^{p_2} (C_1(i,k) - C_2(j,l))^2 \pi_{i,j} \pi_{k,l}, \tag{3.1}$$

where $\mathcal{P}_{p_1,p_2} = \{\pi \in \mathbb{R}_+^{p_1\times p_2} | \pi \mathbb{1}_{p_2} = p_1^{-1}\mathbb{1}_{p_1}, \pi^T \mathbb{1}_{p_1} = p_2^{-1}\mathbb{1}_{p_2}\}$. GW_2 can be used to compare unlabeled graphs with potentially different numbers of nodes, it is symmetric, positive and satisfies the triangle inequality. Furthermore, it is equal to zero when y_1 and y_2 are isomorphic, namely when there exist a bijection $\phi : [\![1, p_1]\!] \to [\![1, p_2]\!]$

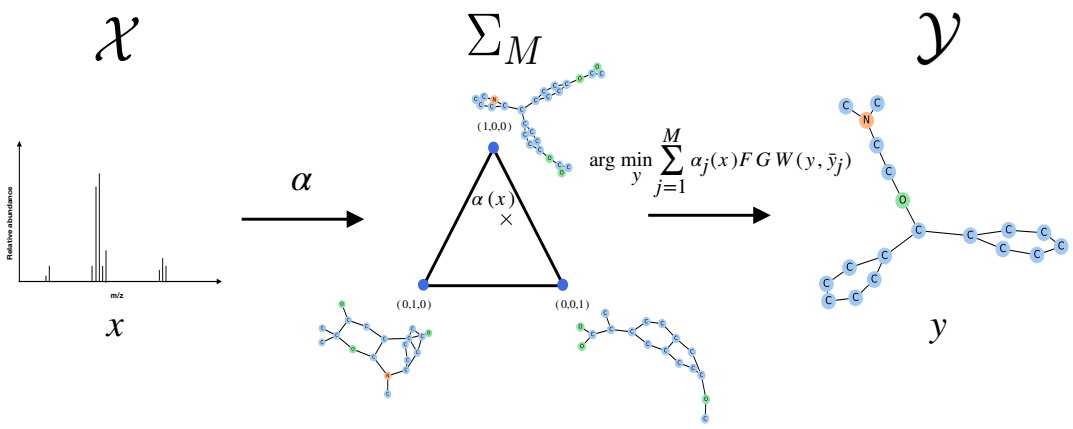

Figure 3.1: Proposed supervised graph prediction model. The input x (left) is mapped with $\boldsymbol{\alpha}(x)$ onto the simplex (center) where the weights are used for computing the prediction as a FGW barycenter (right).

such that $C_2(\phi(i),\phi(j)) = C_1(i,j)$ for all $i,j \in [\![1,p_1]\!]$. GW provides a distance on the unlabeled graph quotiented by the isomorphism, making it a natural metric when comparing graphs.

Fused Gromov-Wasserstein (FGW) distance. The FGW distance has been proposed recently as an extension of GW that can be used to measure the similarity between attributed graphs (Vayer et al., 2020). For a given $0 \leq \beta \leq 1$, the FGW distance between two labeled weighted graphs represented as $y_1 = (C_1, F_1, p_1^{-1}\mathbb{1}_{p_1})$ and $y_2 = (C_2, F_2, p_2^{-1}\mathbb{1}_{p_2})$ is defined as follows (Vayer et al., 2020):

$$\mathrm{FGW}_2^2(y_1,y_2) = \min_{\pi \in \mathcal{P}_{p_1,p_2}} \sum_{i,k,j,l} \Big[(1-\beta)\|F_1(i) - F_2(j)\|_{\mathbb{R}^d}^2 + \beta(C_1(i,k) - C_2(j,l))^2\Big]\pi_{i,j}\pi_{k,l}.$$

The optimal transport plan matches the vertices of the two graphs by minimizing the discrepancy between the labels, while preserving the pairwise similarities between the nodes. Parameter β governs the trade-off between structure and label information. Its choice is typically driven by the application.

3.3 Graph prediction with Fused Gromov-Wasserstein

Relaxed Supervised Graph Prediction. In this work, we consider labeled graph prediction as a *relaxed* structured output prediction problem. We assume that $\mathcal{X}$ is the input space and that the predictions belong to the space $\mathcal{Y}_p$ defined in Section 3.2, for a given value of p, while we observe training data in the finite set $\mathcal{Y}_{dis}$. We define an asymmetric partially relaxed structured loss function $\Delta : \mathcal{Y}_p \times \mathcal{Y}_{dis} \to \mathbb{R}^+$. Given a finite sample $(x_i, y_i)_{i=1}^n$ independently drawn from an unknown distribution ρ on $\mathcal{X} \times \mathcal{Y}_{dis}$, we consider the problem of estimating a target function $f^* : \mathcal{X} \to \mathcal{Y}_p$ with values in the structured objects $\mathcal{Y}_p$ that minimizes the expected risk:

$$\mathcal{R}_\Delta^p(f) = \mathbb{E}_\rho[\Delta(f(X), Y)], \tag{3.2}$$

by an estimate $\hat{f}$ obtained by minimizing the empirical counterpart of the true risk, namely the empirical risk:

$$\hat{\mathcal{R}}^p_\Delta(f) = \sum_{i=1}^{n} \Delta(f(x_i), y_i), \tag{3.3}$$

over the hypothesis space $\mathcal{G}^p \subset \mathcal{M}(\mathcal{X}, \mathcal{Y}_p)$. The goal of this work is to provide a whole framework to address this family of problems instantiated by $p \leq p_{max}$. Note that the complexity of the task depends primarily on p.

FGW as training loss. We propose in this work to use the FGW distance as the loss. More precisely, we define:

$$\forall (y, y') \in \mathcal{Y}_p \times \mathcal{Y}_{dis},\ \Delta_{\mathrm{FGW}}(y, y') := \mathrm{FGW}_2^2(y, y'). \tag{3.4}$$

As FGW is defined for graphs of different sizes, the expression in Eq. (3.4) is well posed. Accordingly, for all $i = 1, \dots n$, we denote $y_i \in \mathcal{Y}_{p_i}$ the relaxed version of $y_i \in \mathcal{Y}_{dis}$ with number of nodes p_i.

Supervised Graph Prediction with FGW. Having fixed a value for p and following these definitions, the empirical risk minimization problem now writes as follows. Given the training sample $\{(x_i, y_i)_{i=1}^n\}$, we want to find a minimizer over $\mathcal{G}^p \subset \mathcal{M}(\mathcal{X}, \mathcal{Y}_p)$ of the following problem:

$$\min_{f \in \mathcal{G}^p} \sum_{i=1}^{n} \mathrm{FGW}_2^2(f(x_i), y_i). \tag{3.5}$$

Remark 3.1. *Using FGW yields an interesting property for the family of problems defined by* $\mathcal{R}^p_\Delta(f) := \mathbb{E}_\rho[FGW_2^2(f(X), Y)]$. *Assume we have different values for* p, *say* $p_1 < p_2 < \dots p_k \leq p_{max}$. *Denote* $r_1^*, \dots, r_k^*$, *the corresponding minima of the respective true risks*

$$\mathcal{R}^{p_1}_\Delta(f), \dots, \mathcal{R}^{p_k}_\Delta(f), \tag{3.6}$$

obtained respectively in $\mathcal{M}(\mathcal{X}, \mathcal{Y}_{p_1}), \dots, \mathcal{M}(\mathcal{X}, \mathcal{Y}_{p_k})$. *In the case of the FGW distance well defined on* $\mathcal{Y} \times \mathcal{Y}$, *all these minimal risks* r_i^* *are comparable, and thus there is a best value* $\tilde{p}$ *among* $p_1, \dots, p_k$ *that corresponds to the best target* $\tilde{f}^*$ *that achieves the minimum of the FGW risk. Hence, in principle, we should also tackle the problem of finding the best value* $\tilde{p}$ *that allows to come closer to the solution in* $\mathcal{Y}_{dis}$ *in expectation. We leave this bilevel optimization problem as future work.*

Structured prediction model. To address this structured regression problem, we propose a generic model $f_\theta : \mathcal{X} \to \mathcal{Y}_p$ expressed as a **conditional FGW barycenter** computed over M template graphs $\bar{y}_j \in \mathcal{Y}$ (See Figure 3.1):

$$f_\theta(x) = \operatorname*{arg\,min}_{y \in \mathcal{Y}_p} \sum_{j=1}^{M} \alpha_j(x; W) \mathrm{FGW}_2^2(y, \bar{y}_j), \tag{3.7}$$

where the weights $\alpha_i(x; W) : \mathcal{X} \to \mathbb{R}^+$ are functions that can be understood as similarity scores between x and x_j. We include in a single parameter $\theta = (M, (\bar{y}_j)_{j=1}^M, W)$ all model's parameters.

A key feature of the proposed model f_θ is that it interpolates in the graph space $\mathcal{Y}$ by using the Fréchet mean with respect to the FGW distance. Therefore, it inherits the good properties of FGW, especially including the invariance under isomorphism (two isomorphic graphs have equal scores in Eq. (3.7)). Moreover, in terms of computations, the proposed model leverages the recent advances in computational optimal transport such as Conditional Gradient descent (Vayer et al., 2019) or Mirror descent for (F)GW with entropic regularization (Peyré et al., 2016).

Properties of f_θ. Relying on recent works that studied in a large extent GW and FGW barycenters, we now discuss the shape of the recovered objects (Peyré et al., 2016; Vayer et al., 2020, Eq. 14). Let us call p the number of nodes of the graph represented by $f_\theta(x)$. The evaluation of f_θ on input x writes as follows: $f_\theta(x) = (C(x;\theta), F(x;\theta), p^{-1}\mathbb{1}_p)$, where the structure and feature barycenters are:

$$C(x;\theta) = p^2 \sum_{j=1}^{M} \alpha_j(x;W) \bar{\pi}_j^T \bar{C}_j \bar{\pi}_j \ \in [0,1]^{p\times p}, \tag{3.8}$$

$$F(x;\theta) = p \sum_{j=1}^{M} \alpha_j(x;W) \bar{F}_j \bar{\pi}_j^T \ \in \mathbb{R}^{p\times d}. \tag{3.9}$$

The $(\bar{\pi}_k)_k$ are the optimal transport plans from $(\bar{C}_k, \bar{F}_k)_k$ to the barycenter $(C(x;\theta), F(x;\theta))$ (Cuturi and Doucet, 2014, Eq. 8) , and thus depend on θ. Note that a very appealing property of using FGW barycenter is that the order p (that fixes the prediction space $\mathcal{Y}_p$) of the prediction does not depend on the parameters θ. This means that a unique trained model can predict several objects with a different resolution p allowing better interpretation at small resolution and finer modeling at higher resolution. This will be illustrated in the experimental section.
In the next sections, we propose two different approaches to learn and define the conditional barycenter. The first one in Section 3.4 leads to a purely nonparametric estimator with $M = n$ and $\bar{y}_j = y_j$ and the second one proposed in Section 3.5 relies on a deep neural network for the weight functions α_js' while the template graphs $(\bar{y}_j)_{j=1}^M$ are learned as well.

3.4 Nonparametric conditional Gromov-Wasserstein barycenter

Non-parametric estimator with kernels. Before addressing the general problem of learning both the template graphs and the weight function α, we adopt a nonparametric point of view to address the structured regression problem. Under some conditions we recover a FGW conditional barycenter estimator of the following form:

$$f_W(x) = \operatorname*{arg\,min}_{y\in\mathcal{Y}_p} \sum_{j=1}^{n} \alpha_j(x;W) \mathrm{FGW}_2^2(y, y_j), \tag{3.10}$$

where $\theta = W$ is now the single parameter to learn and the template graphs $\bar{y}_j$ are not estimated but set as all the training samples y_j. Similarly to scalar or vector-valued regression, one can find many different ways to define the weight functions α_i in the large family of nonparametric estimators (Geurts et al., 2006; Ciliberto et al., 2020). We propose here a kernel approach that leverages kernel ridge regression.

Defining a positive definite kernel on the input space $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, one can consider the coefficients of kernel ridge estimation as in Brouard et al. (2016b); Ciliberto et al. (2020) to define the weight function $\alpha : \mathcal{X} \to \mathbb{R}^n$:

$$\alpha(x) = (K + \lambda I_n)^{-1} k_x \tag{3.11}$$

with the Gram matrix $K = (k(x_i, x_j))_{ij} \in \mathbb{R}^{n \times n}$ and the vector $k_x^T = (k(x, x_1), \ldots, k(x, x_n))$. Such a model leverages learning in vector-valued Reproducing Kernel Hilbert Spaces and is rooted in the Implicit Loss Embedding (ILE) framework proposed and studied by Ciliberto et al. (2020).

Example 3.2. *In the metabolite identification problem (see Section 3.6), the input takes the form of tandem mass spectra. A typical relevant kernel k for such data is the probability product kernel (PPK) (Heinonen et al., 2012).*

3.4.1 Theoretical justification for the proposed model

The framework SELF (Ciliberto et al., 2016) and its extension ILE (Ciliberto et al., 2020) concerns general regression problems defined by an asymmetric loss $\Delta : \mathcal{W} \times \mathcal{Y} \to \mathbb{R}$ that can be written using output embeddings, allowing to solve a surrogate regression problem in the output embedding space. We recall the ILE property and the resulting benefits, especially when working in vector-valued Reproducing Kernel Hilbert Space.

Definition 3.3 (ILE)**.** *For given spaces $\mathcal{W}, \mathcal{Y}$, a map $\Delta : \mathcal{W} \times \mathcal{Y} \to \mathbb{R}$ is said to admit an Implicit Loss Embedding (ILE) if there exists a separable Hilbert space $\mathcal{U}$ and two measurable bounded maps $\chi : \mathcal{W} \to \mathcal{U}$ and $\psi : \mathcal{Y} \to \mathcal{U}$, such that for any $w \in \mathcal{W}, y \in \mathcal{Y}_{dis}$: $\Delta(w, y) = \langle \chi(w), \psi(y) \rangle_{\mathcal{U}}$.*

Note that this definition highlights an asymmetry between the processing of w and y. A regression problem based on a loss satisfying the ILE condition enjoys interesting properties. The following true risk minimization problem: $\min_f \mathbb{E}_\rho[\Delta(f(X), Y)] := \mathbb{E}_\rho[\langle \chi(f(X)), \psi(Y) \rangle_{\mathcal{U}}$, can be converted into i) a surrogate (intermediate) and simpler least square regression problem into the implicit embedding space $\mathcal{U}$, i.e. $\min_{h:\mathcal{X}\to\mathcal{U}} \mathbb{E}_\rho[\|h(X) - \psi(Y)\|_{\mathcal{U}}^2]$, and ii) a decoding phase: $f^*(x) := \arg\min_w \langle \chi(w), h^*(x) \rangle_{\mathcal{U}}$, where h^* is solution of problem i), i.e. $h^*(x) = \mathbb{E}[\psi(Y)|x]$. A nice property proven by Ciliberto et al. (2020) is the one of Fisher consistency, f^* is exactly the minimizer of problem in Eq. (3.2), justifying the surrogate approaches.

Structured prediction with implicit embedding and kernels. Assuming the loss Δ is ILE, when relying on a i.i.d. training sample $\{(x_i, y_i)_{i=1}^n\}$, one gets $\hat{h}$ an estimator of h^* by minimizing the corresponding (regularized) empirical risk and then builds $\hat{f}$.

If we choose to search $\hat{h}$ in the vector-valued Reproducing Kernel Hilbert Space $\mathcal{H}_\mathcal{K}$ associated to the decomposable operator-valued kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{U})$ of the form $\mathcal{K}(x, x') = I_\mathcal{U} k(x, x')$ where k is the positive definite kernel defined in Section 3.4 and $I_\mathcal{U}$ is the identity operator on the Hilbert space $\mathcal{U}$, then the solution to the problem:

$$\min_{h \in \mathcal{H}_\mathcal{K}} \sum_{i=1}^n \|h(x_i) - \psi(y_i)\|_{\mathcal{U}}^2 + \lambda \|h\|_{\mathcal{H}_\mathcal{K}^2},$$

for $\lambda > 0$, writes as $\hat{h}(x) = \sum_{i=1}^{n} \alpha_i(x)\psi(y_i)$ with $\alpha(x)$ verifying Eq. (3.11). Then, $\hat{f}(x)$ can be expressed as

$$\hat{f}(x) = \arg\min_{w \in \mathcal{Z}} \left\{ \langle \chi(w), \sum_{i=1}^{n} \alpha_i(x)\psi(y_i) \rangle = \sum_{i=1}^{n} \alpha_i(x)\Delta(w, y_i) \right\}$$

We show in the following proposition that Δ_{FGW} admits an ILE. This allows us to obtain theoretical guarantees from Ciliberto et al. (2020) for our estimator.

Proposition 3.4. *Δ_{FGW} admits an ILE.*

Proof $\mathcal{Y}_{dis}$ is a finite space by definition. $\mathcal{Y}_p$ is a compact space as $[0,1]^{p \times p}$ and $\mathrm{Conv}(\mathcal{F})^p$ are compact ($\mathcal{F}$ is finite). Moreover, $\forall y' \in \mathcal{Y}_{dis}, y \to \Delta_{\mathrm{FGW}}(y, y')$ is a continuous map (See Lemma 6.1). Therefore, according to Theorem 7 from Ciliberto et al. (2020) $\Delta_{\mathrm{FGW}} : \mathcal{Y}_p \times \mathcal{Y}_{dis} \to \mathbb{R}$ admits an ILE. ■

3.4.2 Excess-risk bounds

Since Δ_{FGW} is ILE, the proposed estimator enjoys consistency (See Theorem 6.2 in Appendix). Moreover, under an additional technical assumption (Assumption 6.3 in Appendix), it verifies the following excess-risk-bound.

Theorem 3.5 (Excess-risk bounds)**.** *Let k be a bounded continuous reproducing kernel such that $\kappa^2 := \sup_{x \in \mathcal{X}} k(x,x) < +\infty$. Let ρ be a distribution on $\mathcal{X} \times \mathcal{Y}_{dis}$. Let $\delta \in (0,1]$ and n_0 sufficiently large such that $n_0^{-1/2} \geq \frac{9\kappa^2}{n_0} \log \frac{n_0}{\delta}$. Under Assumption 6.3, for any $n \geq n_0$, if f_W is the proposed estimator built from n independent couples $(x_i, y_i)_{i=1}^n$ drawn from ρ, then, with probability at least $1 - \delta$*

$$\mathcal{R}_{\Delta}^{p}(f_W) - \mathcal{R}_{\Delta}^{p}(f^*) \leq c \log(4/\delta)\, n^{-1/4}, \tag{3.12}$$

with c a constant independent of n and δ.

Note that $n^{-1/4}$ is the typical rate for structured prediction problems without further assumptions on the problem (Ciliberto et al., 2016, 2020). Theorem 3.5 relies on the attainability assumption 6.3. This can be interpreted as the fact that the proposed GW barycentric model defined an hypothesis space which is able to deal with graph prediction problems that are smooth with respect to the FGW metric. This corroborates with the intuition that for such problems FGW interpolation will obtain good prediction results. We illustrate this theoretical insight on a synthetic dataset in the experimental section. Furthermore, both theorems are valid for any $\mathcal{Y}_p, p \in \mathbb{N}^*$, that is, they provide guarantees for all regression problems defined in Eq. (3.2) for all $p \in \mathbb{N}^*$.

3.5 Neural network-based conditional Gromov-Wasserstein barycenter

In this section, we discuss how to train a neural network model estimator as defined in Equation (3.7) where the template graphs $\bar{y}_i$ are learned simultaneously with the weight function α. This provides a very generic model that inherits the flexibility of deep neural networks and their ability to learn input data representation.

Parameters of the model. First we recap the different parameters that we want to optimize. First, the weights $\alpha(x, W)$ of the barycenter are modeled by a deep neural network with parameters W. Next the templates M graphs $\bar{y}_j$ are also estimated allowing the model to better adapt to the prediction task. It is important to note that M is also a parameter of the model that will tune the complexity of the model and will need to be validated in practice. Note that this parametric formulation is better suited to large scale datasets since the complexity of the predictor will be fixed by M instead of increasing with the number of training data n as in non-parametric models.

Stochastic optimization of the model. We optimize the parameters of the model using a classical ADAM (Kingma and Ba, 2014) stochastic optimization procedure where the gradients are taken over samples or minibatches of the full empirical distribution. We now discuss the computation of the stochastic gradient on a training sample (x_i, y_i). First note that the gradient of $\text{FGW}(f_\theta(x_i), y_i)$ *w.r.t.* θ is actually the gradient of a bi-level optimization problem since f_θ is the solution of a FGW barycenter. The barycenter solutions expressed in Equations (3.8) and (3.9) actually depends on the optimal OT plans $(\bar{\pi}_j)_j$ of the barycenter that depends themselves on θ. But in practice the OT plans $(\bar{\pi}_j)_j$ are solutions of a non-convex and non-smooth quadratic program and are with high probability on a border of the polytope (Maron and Lipman, 2018). This means that we can assume that a small change in θ will not change their value and a reasonable differential of $(\bar{\pi}_j)_j$ *w.r.t.* θ is the null vector. This actually corresponds in Pytorch (Paszke et al., 2019) notation to "detach" the OT plan with respect to the input which is done by default in POT toolbox (Flamary et al., 2021). The gradient of the outer FGW loss can be easily computed as the gradient of the loss with the fixed optimal plan π_i using the theorem from Bonnans and Shapiro (1998). Computing a sub-gradient of the loss $\text{FGW}(f_\theta(x_i), y_i)$ can then be done with the following steps:

1. $(\bar{\pi}_j)_j \leftarrow$ Compute the barycenter $f_\theta(x_i)$.
2. $\pi_i \leftarrow$ Compute the loss $\text{FGW}(f_\theta(x_i), y_i)$.
3. $\nabla_\theta \leftarrow$ Compute the gradient of $\text{FGW}(f_\theta(x_i), y_i)$ with fixed OT plans $(\bar{\pi}_j)_j$ and π_i.

Note that for the matrices $\bar{C}_j$ in the templates, the stochastic update is actually a projected gradient step onto the set of matrices with components belonging to $[0, 1]$.

3.6 Numerical experiments

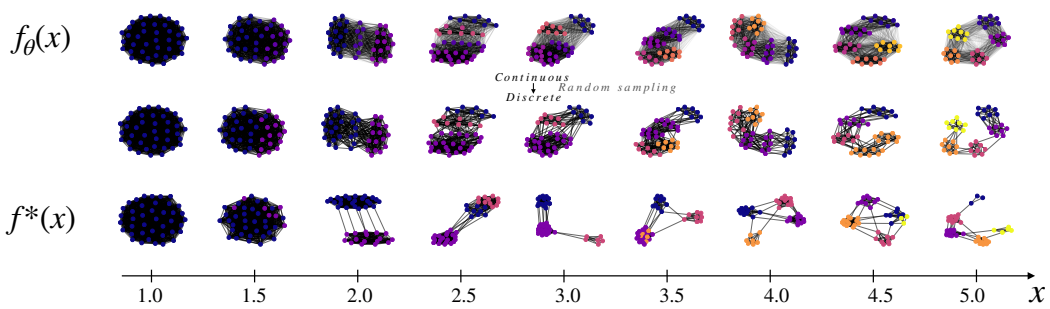

Figure 3.2: Graph prediction on the synthetic dataset as a function of the 1D input x. (top) estimated continuous prediction $f_\theta(x)$, (middle) discrete realizations following the continuous prediction, (bottom) true graph prediction function $f^\star(x)$.

In this section, we evaluate the proposed method on a synthetic problem and the metabolite identification problem.

3.6.1 Synthetic graph prediction problem

Problem and dataset. We consider the following graph prediction problem. Given an input x drawn uniformly in $[1, 6]$, y is drawn using a Stochastic block model with $\lfloor x \rfloor$ blocks, such that the biggest block smoothly splits into two blocks when x is between two integers (see Figure 3.2, bottom line). Each node has a label, which is an integer indicating the block the node is belonging to. More precisely, we take randomly from 40 to 45 nodes for each graph (uniformly in $[\![40, 45]\!]$. There is a probability 0.9 of connection between nodes belonging to the same block, and a probability 0.01 of connection between nodes belonging to different blocks. The probability of connection between nodes belonging to the splitting blocks is $p(x) = 0.889(x - \lfloor x \rfloor) + 0.01$. When a node belongs to the new appearing block its label is the new block's label with probability $(x - \lfloor x \rfloor)$, and the splitting block's label otherwise. We generate a training set of $n = 50$ couples $(x_i, y_i)_{i=1}^n$. Notice that the considered learning problem is highly difficult as one want to predict a graph from a continuous value in $[1, 6]$.

Experimental setting. We test the parametric version of the proposed method with learning of the templates. We use $M = 10$ templates, with 5 nodes, and initialize them drawing $\bar{C}_i \in \mathbb{R}^{5\times5}, \bar{F}_i \in \mathbb{R}^{5\times1}$ uniformly in $[0, 1]^{5\times5}$ and $[0, 1]^{5\times1}$. The weights $\alpha(x; W) \in \mathbb{R}^M$ are implemented using a three-layer (100 neurons in each hidden layer) fully connected neural network with ReLU activation functions, and a final softmax layer. We use $\beta = 1/2$ as FGW's balancing parameter and a prediction size of $n = 40$ during training. During training, we optimize the parameters θ of the model using the continuous relaxed graph prediction model. Interestingly this prediction provides us with continuous versions of the adjacency matrices so we can generate discrete graphs by randomly sampling each edge with a Bernouilli distribution of parameter given by $C(x, \theta)$.

Supervised learning result. The estimated graph prediction model on the synthetic dataset is illustrated in Figure 3.2. We can see that the learned map is indeed recovering the evolution of the graphs as a function of x. This shows, as suggested by the theoretical results in Section 3.4, that the FGW metric is a a good data fitting term and that FGW barycenters are a good way to interpolate continuously between discrete objects. This is particularly true on this problem where a small change w.r.t x induces small change in the output of $f^*(x)$ according to the FGW metric.

Interpretability and flexibility of the proposed model. We now illustrate how interpretable is the estimated model. First we recall that the prediction is actually a Fréchet mean w.r.t the FGW distance, according to the weights $\alpha_j(x)$ and the templates $(\bar{y}_j)_{j=1}^m$. In practice it means that we can plot the template graphs $(\bar{y}_j)_{j=1}^m$ to check that the learned templates are indeed similar (with less nodes) to training data. But on this synthetic dataset we can also plot the trajectory of the barycenter weights α_j on the simplex as a function of x which we did in Figure 3.3. We can see in the figure that in practice the weights $\alpha_j(x)$ are sparse concentrated on the templates on the left of the Figure starting with a graph with one connected cluster and ending with a graph with 5 clusters following the true model $f^\star$.

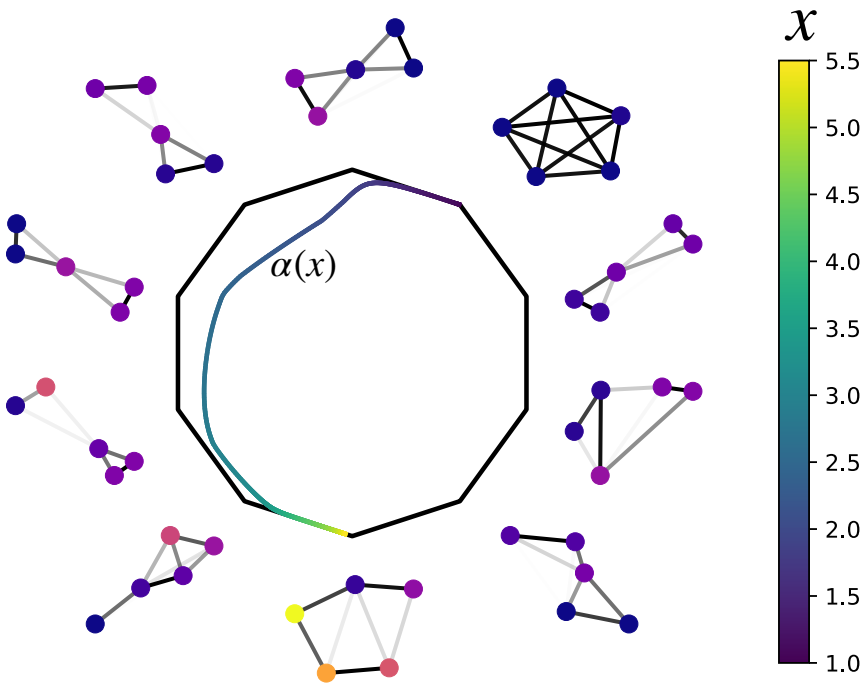

Figure 3.3: Learned templates $(\bar{y}_j)_{j=1}^m$ on the synthetic dataset and trajectory of the weights $\alpha(x)$ on the simplex as a function of x.

We now illustrate one very interesting property of our model: the ability to predict graphs with a varying number of nodes p for a given input x. An example of the predicted graphs for $x = 5$ is provided in Figure 3.4. It is interesting to note that even with small templates of 5 nodes, the proposed barycentric graph prediction model is able to predict big graphs while preserving their global structure. This is particularly true for Stochastic Block Models graphs that can by construction be factorized with a small number of clusters. Note that the number of nodes in the templates $(\bar{y}_j)_{j=1}^m$ can be seen as a regularization parameter. The model is also very flexible in the sens that the FGW barycenter modeling allows for templates with different number of nodes allowing for a coarse to fine modeling of the data.

3.6.2 Metabolite identification problem

Problem and dataset. An important problem in metabolomics is to identify the small molecules, called metabolites, that are present in a biological sample. Mass spectrometry is a widespread method to extract distinctive features from a biological sample in the form of a tandem mass (MS/MS) spectrum. The goal of this problem is to predict the molecular structure of a metabolite given its tandem mass spectrum. Labeled data are expensive to obtain, and despite the problem complexity not many labeled data are available in datasets. Here we consider a set of 4138 labeled data, that have been extracted and processed in Dührkop et al. (2015), from the GNPS public spectral library (Wang et al., 2016).

Experimental setting. We test the nonparametric version of the proposed method, using a probability product kernel on the mass spectra, as it has been shown to be a good choice on this problem (Brouard et al., 2016a). We use $\beta = 0.5$ as FGW balancing parameter. We split the dataset into a training set of size $n = 3000$ and a test set of

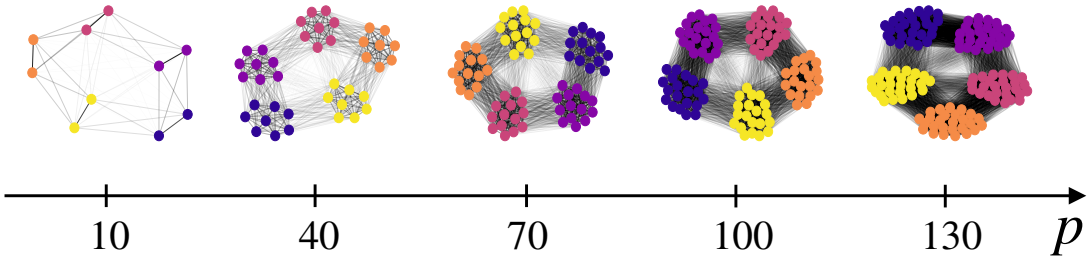

Figure 3.4: Predicted graphs with the estimated model $f_\theta(x)$ with a varying number of nodes p for $x = 5$.

size $n_{te} = 1138$. On this problem, structured prediction approaches that have been proposed fall back on the availability of a known candidate set of output graphs for each input spectrum (Brouard et al., 2016a). This means that in practice for prediction on new data, we will not solve the FGW barycenter in (3.7) but search among the possible candidates in $\mathcal{Y}$ the one minimizing the barycenter loss.
In a first experiment, we evaluate the performance of FGW as a graph metric. To this end we compare the performance of various graph metrics $D : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ used in the model: $\arg\min_{y \in \mathcal{Y}} \sum_{j=1}^{n} \alpha_j(x; W) D(y, y_j)$. We consider the metric induced by the standard Weisfeiler–Lehman (WL) graph kernel that consists in embedding graphs as a bag of neighbourhood configurations (Shervashidze et al., 2011). The FGW one-hot distance corresponds to the FGW distance and using a one-hot encoding of the atoms. The FGW fine distance corresponds to the one-hot distance concatenated with additional atom features: number of attached hydrogens, number of heavy neighbours, formal charge, is in a ring, is in an aromatic ring. Additional features are normalize by their maximum values in the molecule at hand. The FGW diffuse distance corresponds to the FGW distance and using a one-hot encoding of the atoms which has been diffused, namely: $F_{\text{diff}} = e^{-\tau \text{Lap}(C)} F$, where $\tau > 0$, $\text{Lap}(C)$ denotes the normalized Laplacian of C as proposed in Barbe et al. (2020). Fingerprints are molecule representations, well engineered by experts, that are binary vectors. Each value of the fingerprint indicates the presence or absence of a certain molecular property (generally a molecular substructure). Several machine learning approaches using fingerprints as output representations have obtained very good performances for metabolite identification (Dührkop et al., 2015; Brouard et al., 2016a; Nguyen et al., 2018) or other tasks, such as metabolite structural annotation (Hoffmann et al., 2021). In the last two Casmi challenges (Schymanski et al., 2017), such approaches have obtained the best performances for the best automatic structural identification category. Here we consider the metrics induced by linear and gaussian kernels between fingerprints of length $d = 2765$. We compute the test predictions using the test spectra with less than 300 candidates for faster computation: 286 test points. For the FGW metrics, we compute them using the 5 greatest weights $\alpha_i(x)$. We evaluate the results in terms of top-k accuracy: percentage of true output among the k outputs given by the k greatest scores in the model. The two hyperparameters (ridge regularization parameter λ and the output metric's parameter) are selected using a validation set (1/5 of the training set) and top-1 accuracy.

Graph metrics comparison. The results given in Table 3.1 shows that gaussian fingerprints is state-of-the-arts on this dataset when a candidate set is available. We see that the FGW greatly benefits from the improved fine and diffuse metrics showing the

Table 3.1: Top-k accuracies for various graph kernels on the metabolite identification dataset.

	Top-1	Top-10	Top-20
WL kernel	28.7%	57.2%	71.2%
Linear fingerprint	33.6%	76.2%	80.1%
Gaussian fingeprint	48.7%	81.0%	86.0%
FGW one-hot	24.6%	64.2%	75.4%
FGW fine	31.2%	64.9%	76.1%
FGW diffuse	40.0%	72.3%	82.5%

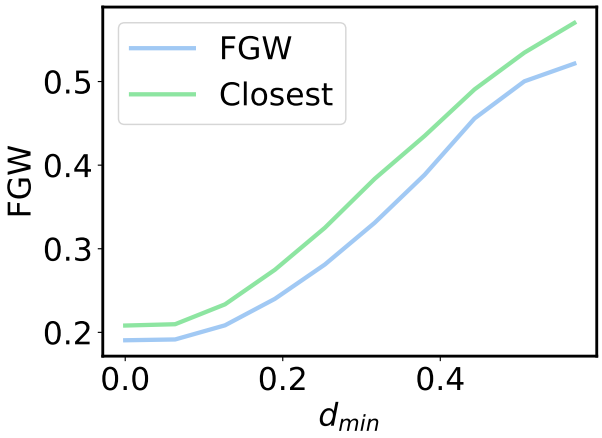

Figure 3.5: No candidate set setting. In average, the FGW barycenter (blue) using the 10 molecules with the greatest weights $\alpha_j(x)$ is closer to the true molecule, than the molecule with the greatest weight $\alpha_j(x)$: closest template prediction (green).

adaptation potential of the FGW metric to the graph space at hand reaching competitive performance against baselines and even beating Fingerprints with linear kernel and WL kernels.

Predicting novel molecules. Being able to interpolate novel graphs without using predefined candidate sets is a great advantage of the proposed method. Such computation is in general intractable (e.g. with WL and fingerprints metrics). In this experiment, we evaluate the performance of the estimator when computing the barycenter over $\mathcal{Y}_p$, and not over the candidate sets. For a given test input x, let us define $d_0(x)$ the FGW (one-hot) distance of the training molecule with the greatest $\alpha_j(x)$ to the true molecule. $d_0(x)$ measures the level of interpolation difficulty: very small d_0 means that the true molecule is close to a training molecule and no interpolation is required. We compute, over 1000 test data, the mean $d_0(x)$ and the mean FGW (one-hot) distance between the predicted barycenter (using the 10 largest $\alpha_j(x)$) and the true test molecule. In Figure 3.5, we plot the two mean distances, with respect to a filtering threshold d_{min} such that only the test point with $d_0(x) > d_{min}$ are used when computing these means. We can see that the FGW interpolation allows to become closer to the true output than only predicting the output with the greatest weight $\alpha_j(x)$, even more when the interpolation is required ($d_0(x)$ big). This validates the choice of FGW as a way to interpolate between real-world graphs.

3.7 Conclusion

We proposed in this work a novel framework for graph prediction using optimal transport barycenters to interpolate continuously in the output space. We discussed both a non-parametric estimator with theoretical guarantees and a parametric one based on neural network models that can be estimated with stochastic gradient methods. The method was illustrated on synthetic and real life data showing the interest of the continuous relaxation especially when targets are not available.
Future works include estimation of the target number of nodes $p(x)$ and supervised learning of complementary feature on the templates that can guide the FGW barycenters.

4

Vector-valued Least-Squares Regression under Output Regularity Assumptions

4.1 Introduction

Learning vector-valued functions plays a key role in a large variety of fields such as economics (Lütkepohl, 2013), physics, computational biology, where multiple variables have to be predicted simultaneously. As opposed to solving multiple single regression problems, the interest of vector-valued regression lies on the ability to take into account the dependence structure among the output variables by appropriate regularization (see for instance Micchelli and Pontil, 2005; Baldassarre et al., 2012; Álvarez et al., 2012; Lim et al., 2015) or by imposing a low-rank assumption (Anderson, 1951; Izenman, 1975; Velu and Reinsel, 2013). Regarding the infinite dimensional output case, besides functional output regression (Kadri et al., 2016), the motivation for vector-valued regression mainly comes from the application of surrogate approaches in Structured Output Prediction (Weston et al., 2003; Geurts et al., 2006; Kadri et al., 2013; Brouard et al., 2016b; Ciliberto et al., 2020). In order to learn a model to predict an output with some discrete structure, surrogate approaches embed the structured output variable into a Hilbert space and thus boil down to vector-valued regression with a potentially infinite dimensional output space. At prediction time, decoding allows to return a prediction in the original structured output space. Image completion (Weston et al., 2003), label ranking (Korba et al., 2018) and graph prediction (Brouard et al., 2016a) are all examples of structured prediction tasks that can be handled by surrogate approaches.

One way to implement infinite dimensional output regression consists in learning in vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHS) (Micchelli and Pontil, 2005). In particular, regularized least-squares estimators in vv-RKHS enjoy strong theoretical guarantees (see Caponnetto and De Vito, 2007). However complex tasks such as structure prediction very often involve a limited amount of training data compared to the complexity of the input and output data. To overcome this issue, the structure of the target output can be leveraged. This is typically the goal of reduced-rank approaches (Mukherjee and Zhu, 2011; Luise et al., 2019).

In this chapter, our aim is to improve upon the regularized least-squares estimators by imposing a rank constraint on the least-squares estimator. Our contributions are three-fold.

As a first contribution, we introduce a novel reduced-rank estimator for vector-valued least-squares regression in the general case of infinite dimensional outputs. Denoting $\mathcal{Z}$ a Hilbert space and $\mathcal{X}$ a Polish space, we consider the following relationship between the input variable and the output variable:

$$z = h^*(x) + \epsilon, \tag{4.1}$$

where the pair of random vectors (x, z) takes its values in $\mathcal{X} \times \mathcal{Z}$, $\epsilon \in \mathcal{Z}$ is a random noise independent of x with expectation $\mathbb{E}[\epsilon] = 0$ and $h^* : \mathcal{X} \to \mathcal{Z}$ is a measurable function. Assuming we have already an estimator $\hat{h} : \mathcal{X} \to \mathcal{Z}$ of h^* built from a training i.i.d. sample $(x_i, z_i)_{i=1}^n$, we propose to learn a linear operator $\hat{P}$ of rank p, for $p \in \mathbb{N}^*$ allowing to project $\hat{h}(x)$ onto $\mathcal{Z}_0 \subset \mathcal{Z}$ with $\dim(\mathcal{Z}_0) \leq p$ giving rise to the following new estimator:

$$x \mapsto \hat{P}\hat{h}(x).$$

This novel estimator generalizes the reduced-rank kernel ridge regression estimator proposed by Mukherjee and Zhu (2011) to the infinite dimensional case.

The second contribution of this work is to study the proposed least-squares estimator under output regularity assumptions and provide excess-risk bounds. We assume that h^* belongs to a vector-valued reproducing kernel Hilbert Space, namely $h^* = H\phi(.)$ with $H \in \mathcal{Z} \otimes \mathcal{H}_x, \|H\|_{\mathrm{HS}} < +\infty$, and $\phi : \mathcal{X} \to \mathcal{H}_x$ is a canonical map associated to a scalar-valued kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The difficulty of the learning problem in Eq. (4.1) can be characterized by standard complexity measures. For instance, the capacity condition measures the regularity of the features in terms of eigenvalue decay rate of the covariance operator $C = \mathbb{E}[\phi(x) \otimes \phi(x)]$, and the source condition measures the regularity of H in terms of alignment of H^*H with C (Caponnetto and De Vito, 2007; Ciliberto et al., 2020; Varre et al., 2021). The more regular the problem is, the better are the statistical guarantees. In this work, we consider regularity assumptions on the outputs of the learning problem. We measure the eigenvalue decay rates of the covariance operator $\mathbb{E}[h^*(x) \otimes h^*(x)]$, and $\mathbb{E}[\epsilon \otimes \epsilon]$, and also the alignment of HH^* with HCH^*.

The third contribution of this work is a novel structured prediction method, which leverages our reduced-rank estimator in the surrogate regression problem. The proposed approach makes use of both an input and an output kernel. In this case, the resulting surrogate regression problem's output space is thus a reproducing kernel Hilbert space. The least-squares analysis allows to prove the statistical and computational interest of the structured prediction method. In particular, consistency and learning rates for our structured prediction method are given. Moreover, we show by an extensive empirical study on different real world structured prediction tasks that the proposed approach improves upon full rank and state-of-the art structured prediction approaches.

Outline. The chapter is organized as follows. In Section 4.2, we provide a novel reduced-rank method for solving vector-valued least-squares problems. In Section 4.3, we give learning bounds for the proposed least-squares estimator. Then, we study under which setting this method improves the statistical and computational performance. In particular, our analysis includes and extends the interest of reduced-rank regression beyond the standard setting of reduced-rank regression where the optimum is assumed to be low-rank, and the noise homogeneous in $\mathcal{Z}$. In Section 4.4, we show how the proposed estimator can be advantageously used in structured prediction with surrogate methods. We give an excess-risk bound for the resulting structured predictor, inherited from our least-squares theoretical analysis. In Section 4.5, we illustrate our theoretical analysis on synthetic least-squares problems. We empirically show the benefit of the method in structured prediction on three different problems: image reconstruction, multi-label classification, and metabolite identification.

4.2 Problem setting and proposed estimator

In this section, we introduce the learning setting of vector-valued least-squares regression. Then, we give background on kernel ridge regression. Finally, we present the reduced-rank least-squares estimator proposed in this work.

Vector-valued least-squares regression. We consider the problem of estimating a function $h : \mathcal{X} \to \mathcal{Z}$ with values in a separable Hilbert space $\mathcal{Z}$ with norm $\|.\|_{\mathcal{Z}}$, given a finite set $\{(x_i, z_i)_{i=1}^n\}$ independently drawn from an unknown distribution ρ on $\mathcal{X} \times \mathcal{Z}$, minimizing the expected risk

$$\mathcal{R}(h) = \mathbb{E}_\rho[\|h(x) - z\|_{\mathcal{Z}}^2]. \tag{4.2}$$

The solution is given by $h^*(x) := \mathbb{E}_{\rho(z|x)}[z]$. We define the noise ϵ as the random variable defined by the following equation

$$z = h^*(x) + \epsilon. \tag{4.3}$$

In practice, solving (4.2) requires the choice of an hypothesis space $\mathcal{H}$. In this work, we consider reproducing kernel Hilbert space (RKHS).

Reproducing kernel Hilbert spaces. Given a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, one can build a Hilbert space $\mathcal{H}_x$ of scalar-valued functions $\mathcal{H}_x$, called the associated RKHS of k, defined by the completion $\mathcal{H}_x = \overline{\text{span}\{k(x,.)|x \in \mathcal{X}\}}$ according to the norm induced by the scalar product $\langle k(x,.), k(x',.)\rangle_{\mathcal{H}_x} := k(x,x')$. There is a one-to-one relation between a kernel k and its associated RKHS (Aronszajn, 1950). A crucial tool is the representer theorem which allows to solve in practice regularized empirical risk minimization problems over RKHS (Wahba, 1990; Schölkopf et al., 2001).

Vector-valued reproducing kernel Hilbert spaces. The theory of vector-valued RKHSs (vv-RKHSs) extends the theory of real-valued RKHS by enabling to build Hilbert spaces of vector-valued functions (Senkene and Tempel'man, 1973; Micchelli and Pontil, 2005; Carmeli et al., 2010). We note A^* the adjoint of any operator A. An operator-valued kernel is an application $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Z})$ with values in the set of bounded linear operator on $\mathcal{Z}$, satisfying the two following properties: $K(x,x') = K(x',x)^*$ and $\sum_{i,j=1}^n \langle K(x_i,x_j')z_i, z_j\rangle_{\mathcal{Z}} \geq 0$ for any $n \in \mathbb{N}^*$, $(x_1,z_1),\ldots,(x_n,z_n) \in \mathcal{X} \times \mathcal{Z}$. Then, akin to scalar-valued kernel, one can build a Hilbert space $\mathcal{H}$ of vector-valued function from $\mathcal{X}$ to $\mathcal{Z}$, called the associated RKHS of K, defined by the completion $\mathcal{H} = \overline{\text{span}\{K(x,.)z|(x,z) \in \mathcal{X} \times \mathcal{Z}\}}$ according to the norm induced by the scalar product $\langle K(x,.)z, K(x',.)z'\rangle_{\mathcal{H}} := \langle K(x,x')z, z'\rangle_{\mathcal{Z}}$. There is a one-to-one relation between a kernel K and its associated vv-RKHS. Learning with operator-valued kernels is also possible thanks to representer theorems (Micchelli and Pontil, 2005).

Kernel ridge regression. The kernel ridge regression method (KRR) considers the estimator minimizing the following empirical objective

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \|h(x_i) - z_i\|_{\mathcal{Z}}^2 + \lambda \|h\|_{\mathcal{H}}^2 \tag{4.4}$$

where $\mathcal{H}$ is the RKHS associated to an operator-valued kernel K. In this work, we consider kernel of the form $K(x,x') = k(x,x')I_{\mathcal{Z}}$, where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite scalar-valued kernel on $\mathcal{X}$. In this case, the solution of the problem above can be computed in closed-form as follows:

$$\hat{h}(x) = \sum_{i=1}^{n} \alpha_i(x) z_i, \quad \text{with } \alpha(x) = (K + n\lambda)^{-1} k_x \tag{4.5}$$

where $K = (k(x_i,x_j))_{i,j=1}^n \in \mathbb{R}^{n\times n}$, and $k_x = (k(x,x_i))_{i=1}^n \in \mathbb{R}^n$.

Related works in reduced-rank regression. Reduced-rank (or low-rank) estimators are estimators whose predictions $\hat{z} \in \mathcal{Z}$ lie in a linear subspace $\mathcal{Z}_0 \subset \mathcal{Z}$, estimated from the data. Reduced-rank regression methods have been proposed for both linear models (Izenman, 1975) and non parametric models (Mukherjee and Zhu, 2011; Foygel et al., 2012; Rabusseau and Kadri, 2016; Luise et al., 2019). Two ways of building reduced-rank estimators have been proposed so far. A first way consists in imposing small rank constraints on the estimated linear operator (Izenman, 1975; Mukherjee and Zhu, 2011; Rabusseau and Kadri, 2016): on other words, the obtained estimators can be written as full-rank estimators that has been projected with estimated projection operators for a chosen rank p. Among those works devoted to finite dimensional vector-valued regression, the contribution of Rabusseau and Kadri (2016) differs in many ways. They consider a tensor output (the constraint is thus a multilinear rank constraint) and also provide learning bounds. Another way to address reduced-rank regression is to use nuclear norm (or trace norm) penalization as a convex relaxation to rank penalization as developed in (Romera-Paredes et al., 2013; Foygel et al., 2012; Luise et al., 2019). It is worth mentioning that only Luise et al. (2019) tackle an infinite dimensional vector valued-regression problem and provide a statistical study. More precisely, in terms of statistical guarantees, Rabusseau and Kadri (2016) and Luise et al. (2019) show improved constants in learning bounds when using reduced-rank regression, in comparison with full-rank, in their respective settings.

Proposed least-squares estimator. We introduce a non-parametric estimator belonging to the family of reduced-rank estimators. Let $\lambda_1, \lambda_2 > 0$ and $p \in \mathbb{N}^*$. Let $\mathcal{P}_p$ be the set of the orthogonal projections from $\mathcal{Z}$ to $\mathcal{Z}$ of rank p. We note $\hat{h}_\lambda$ a KRR estimator defined using with the training sample $(x_i,z_i)_{i=1}^n$ and a regularization parameter $\lambda > 0$.
Ideally, we would propose the reduced-rank estimator $x \mapsto P\hat{h}_{\lambda_2}(x)$ where P is the operator defined as follows:

$$P := \operatorname*{arg\,min}_{\mathsf{P}\in\mathcal{P}_p} \mathbb{E}[\|\mathsf{P}h^*(x) - h^*(x)\|_{\mathcal{Z}}^2]. \tag{4.6}$$

Nevertheless, P is unknown, so we replace it by the following empirical estimator

$$\hat{P}_{\lambda_1} := \operatorname*{arg\,min}_{\mathsf{P}\in\mathcal{P}_p} \frac{1}{n}\sum_{i=1}^{n} \|\mathsf{P}\hat{h}_{\lambda_1}(x_i) - \hat{h}_{\lambda_1}(x_i)\|_{\mathcal{Z}}^2, \tag{4.7}$$

based on a KKR estimator $\hat{h}_{\lambda_1}$ of h^*, with possibly $\lambda_1 \neq \lambda_2$. Eventually, this approximation gives rise to the following proposition for our reduced-rank estimator with hyperparameters (p,λ_1,λ_2):

$$x \mapsto \hat{P}_{\lambda_1}\hat{h}_{\lambda_2}(x). \tag{4.8}$$

$\mathcal{X}$	input space
$\mathcal{Y}$	structured output space
$\mathcal{Z}$	regression output Hilbert space
$\|.\|_{\mathcal{Z}}$	norm of the Hilbert space $\mathcal{Z}$
n/n_{te}	number of training data/test data
h^*	least-squares optimum $x \to \mathbb{E}_{\rho(z\|x)}[z]$
Δ	structured loss $\Delta : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^+$
f^*	structured prediction optimum $x \to \arg\min_{\hat{z}\in\mathcal{Z}} \mathbb{E}_{\rho(z\|x)}[\Delta(z,\hat{z})]$
k	positive definite kernel on $\mathcal{X}$
$\mathcal{H}_x$	RKHS associated to k
$\mathcal{H}$	vv-RKHS associated to $K(x,x') = k(x,x')I_{\mathcal{Z}}$
$\mathcal{P}_p$	space of orthogonal projections from $\mathcal{Z}$ to $\mathcal{Z}$ with rank p
P	$\arg\min_{\mathsf{P}\in\mathcal{P}_p} \mathbb{E}[\|\|\mathsf{P}h^*(x) - h^*(x)\|\|^2_{\mathcal{Z}}]$
A^*	adjoint of A
$A \preceq B$	$\forall u, \langle u, Au\rangle \leq \langle u, Bu\rangle$
$\mu_p(A)$	p-th eigenvalue of A sorted in decreasing order
$\|.\|_{\mathrm{HS}}$	Hilbert-Schmidt norm
$\|.\|_{\infty}$	operator norm
$a \otimes b$	defined such as $\forall x, a \otimes bx = \langle b, x\rangle a$
$S_p(A)$	$\sum_{k=1}^{p} \mu_k(A)$

Table 4.1: Notations

Remark 4.2.1. *Note that P is the projection onto the span of the p eigenvectors of the covariance operator $\mathbb{E}[h^*(x) \otimes h^*(x)]$ corresponding to the p greatest eigenvalues. Similarly, $\hat{P}_{\lambda_1}$ is the projection onto the span of the p eigenvectors of the empirical covariance operator $\frac{1}{n}\sum_{i=1}^{n} \hat{h}_{\lambda_1}(x_i) \otimes \hat{h}_{\lambda_1}(x_i)$ corresponding to the p greatest eigenvalues.*

The proposed estimator allows to cope with any separable Hilbert output space $\mathcal{Z}$ (potentially infinite dimensional), which is of practical interest (See Section 4.4). Furthermore, efficient and theoretically grounded approximation methods for KRR and kernel principal component analysis (Rudi et al., 2015; Rudi and Rosasco, 2017; Sterge et al., 2020) can be straightforwardly leveraged to alleviate the computation of this estimator. For sake of simplicity, in the remainder of the chapter, except when it is necessary, we omit the dependency in λ_1 and λ_2 and use notations $\hat{h}$ and $\hat{P}$.

Remark 4.2.2. *The proposed estimator can be seen as a generalization of the reduced-rank estimator defined in (Mukherjee and Zhu, 2011) for finite dimensional vector-valued to the infinite dimensional output case and when λ_1 and λ_2 are not necessarily equal. In this work, we additionally provide learning bounds by leveraging the linear structure of the noise ϵ and those of the outputs $h^*(x)$.*

Notations are gathered in Table 4.1.

4.3 Theoretical analysis

In this section, we present a statistical analysis of the proposed estimator. We start, in Section 4.3.1, by giving the assumptions on the learning problem that we considered. Then, in Section 4.3.2, we provide learning bounds. Finally, in Section 4.3.3, we study

under which setting reduced-rank regression is statistically and computationally beneficial.

4.3.1 Assumptions

Here, we introduce and discuss the main assumptions that we need in order to prove our results.

Assumption 4.1 (attainable case)**.** *We assume that the solution h^* belongs to the RKHS associated to the kernel $K(x,x') = k(x,x')I_{\mathcal{Z}}$, i.e. there exists a linear operator H from $\mathcal{H}_x$ to $\mathcal{Z}$ with $\|H\|_{\mathrm{HS}} < +\infty$ such that:*

$$h^*(x) = H\phi(x). \tag{4.9}$$

This assumption states that the solution h^* indeed belongs to the chosen hypothesis space $\mathcal{H}$. It is a standard assumption in the learning theory (Ciliberto et al., 2020).

Assumption 4.2 (regularity of target's outputs)**.** *The operator $M = \mathbb{E}[h^*(x) \otimes h^*(x)]$ satisfies the following property. There exists $\alpha \in [0,1]$ such that:*

$$c_1 := \mathrm{Tr}(M^\alpha) < +\infty. \tag{4.10}$$

Assumption 4.2 is always verified for $\alpha = 1$ (as $\mathrm{Tr}(M) \leq \|H\|^2_{\mathrm{HS}}\kappa^2$), and the smaller the α the faster is the eigenvalue decay of M. It quantifies the regularity of the target's outputs $h^*(x) \in \mathcal{Z}$. As a limiting case, when M is finite rank $\alpha = 0$. The capacity condition is a standard assumption for least-squares problems, which can be written $\mathrm{Tr}(C^r) < +\infty$ with $r \in [0,1]$, and that characterises instead the regularity of the features $\phi(x) \in \mathcal{H}_x$. Remark that it implies the Assumption 4.2 to hold with at least $\alpha \leq r$, but $\alpha \ll r$ is possible.

Assumption 4.3 (output source condition)**.** *The operators H and $C = \mathbb{E}[\phi(x) \otimes \phi(x)]$ satisfy the following property. There exists $\beta \in [0,1]$, $c_2 > 0$ such that:*

$$HH^* \preceq c_2 M^{1-\beta}. \tag{4.11}$$

Assumption 4.3 is always verified for $\beta = 1$ (as $\|H\|_\infty < +\infty$), and the smaller the β the stricter the assumption is. It quantifies the alignment of the left-singular vectors of H with the main components of M. The source condition is a standard assumption for least-squares problems, which can be written $H^*H \preceq aC^{1-r}$ with $r \in [0,1], a > 0$, and that quantifies instead the alignment of the right-singular vectors of H with the main components of C (See, e.g. Ciliberto et al., 2020; Caponnetto and De Vito, 2007). The Assumption 4.3 allows to show a fast convergence rate of $\hat{P}$. In general, Assumption 4.3 can be maximum ($\beta = 0$) while the source condition is arbitrarily weak ($r = 1$).

Assumption 4.4 (diffuse noise and concentrated signal)**.** *The operators M and $E = \mathbb{E}[\epsilon \otimes \epsilon]$ satisfy the following property. There exists $\gamma \in [0,1]$, $c_3 > 0$ such that*

$$c_3 M^{1-\gamma} \preceq E. \tag{4.12}$$

Assumption 4.4 quantifies the alignment of the main components of E and M, and the greater the γ the more the noise is diffuse in comparison to the signal. As a limiting case, when $\gamma \to 1$, then $\sigma^2 I_{\mathcal{Z}} \preceq E$ with a certain $\sigma^2 > 0$, which is only possible in finite dimension (e.g. $E = \sigma^2 I_{\mathcal{Z}}$, homogeneous noise commonly assumed in low-rank regression).

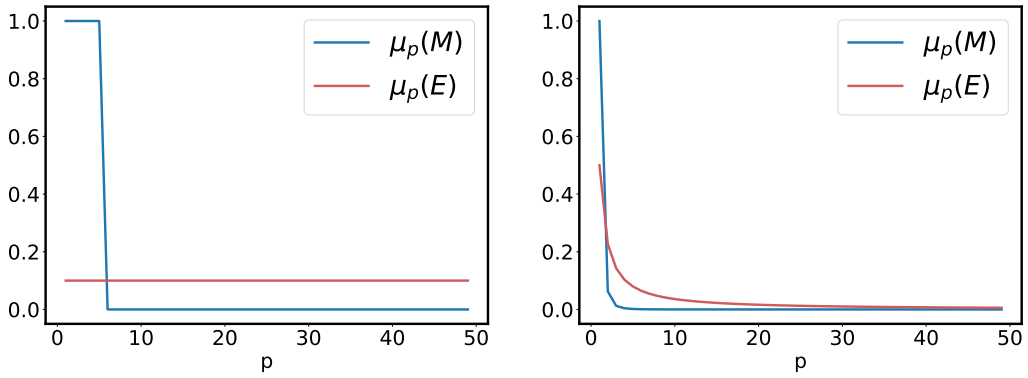

Figure 4.1: Illustration of finite-rank setting with $r = 5$, $\sigma_c^2 = 1, \sigma_\epsilon^2 = 0.1$ (Left) and polynomial setting with $r_c = 3/2, r_h = 5/4, r_e = 8/7$ (Right). We plot $p \to \mu_p(M) = \langle v_p, M v_p \rangle_{\mathcal{Z}}$ and $p \to \mu_p(E) = \langle v_p, E v_p \rangle_{\mathcal{Z}}$.

Example 4.5 (finite-rank example)**.** *The standard low-rank regression setting (See Figure 4.1 left) corresponds to* $\mathcal{Z} = \mathbb{R}^d$, $C = \sigma_c^2 I_{\mathcal{H}_x}$ *with* $\sigma_c^2 > 0$, $H = \sum_{i=1}^r v_i \otimes u_i$ *with* $r \in \mathbb{N}^*$, $E = \sigma_\epsilon^2 I_{\mathcal{Z}}$ *with* $\sigma_\epsilon^2 > 0$, $(u_i)_i$, $(v_i)_i$ *being orthonormal bases (ONB) of respectively* $\mathcal{H}_x$ *and* $\mathcal{Z}$*. In this case, the assumptions are verified with* $\alpha = 0, \beta = 0, \gamma = 1$.

Example 4.6 (polynomial example)**.** *In this dissertation, we study reduced-rank regression beyond low-rank setting. For instance, we can consider polynomial forms (See Figure 4.1 right) for* $C = \sum_{i=1}^{+\infty} i^{-r_c} u_i \otimes u_i$, $H = \sum_{i=1}^{+\infty} i^{-r_h} v_i \otimes u_i$, $E = 0.5 \times \sum_{i=1}^{+\infty} i^{-r_e} v_i \otimes v_i$, *with* $(u_i)_i$ *and* $(v_i)_i$ *being (ONB) of* $\mathcal{H}_x$ *and* $\mathcal{Z}$*, respectively. In this case, the assumptions are verified with* $\alpha = \frac{2}{2r_h + r_c}$, $c_1 = \mathrm{Tr}(M^\alpha) < 2$, $\beta = \frac{r_c}{2r_h + r_c}$, $\gamma = 1 - \frac{r_e}{2r_h + r_c}$.

4.3.2 Main Result

Now, we present the main result of this work which is Theorem 4.7. Under Assumptions 4.1, 4.2, 4.3, 4.4, it provides a bound on the proposed estimator's excess-risk for a chosen $p = \mathrm{rank}(\hat{P})$.

Theorem 4.7 (Learning bounds)**.** *Let* $\hat{P}\hat{h}$ *be the proposed estimator in Eq.* (4.8) *with* $\mathrm{rank}(\hat{P}) = p$*, built from* n *independent couples* $(x_i, z_i)_{i=1}^n$ *drawn from* ρ*. Let* $\delta \in [0,1]$*. Under the Assumptions 4.1, 4.2, 4.3, 4.4, there exists constants* $c_4, c_5, c_8 > 0$, $n_0 \in \mathbb{N}^*$ *defined in the proof, and independent of* p, n, δ*, such that, if* $\mu_{p+1}(M) \geq c_8 \log^8(\frac{8}{\delta}) n^{-\frac{1}{\beta+1}}$ *and* $n \geq n_0$*, then with probability at least* $1 - 3\delta$*,*

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \left(c_4 \sqrt{p} n^{-1/4} + c_5 S_p(E)^{1/4} \right) n^{-1/4} \log(n/\delta) + \sqrt{3c_1} \mu_{p+1}(M)^{1/2(1-\alpha)} \tag{4.13}$$

with $S_p(E) = \sum_{i=1}^p \mu_i(E)$.

The bound is the sum of two terms: the first one increases with p, the second one decreases with p. When $p = o(\sqrt{n})$, the first term is dominated by a term proportional to $S_p(E)^{1/4} \log(n/\delta) n^{-1/4}$, which should be compared to the dominating term of the kernel ridge estimator's bound $\mathrm{Tr}(E)^{1/4} n^{-1/4}$ (cf. Lemma 6.10): instead of the total

amount of noise $\mathrm{Tr}(E)$, the reduced-rank estimator only incurs the quantity within the p main components of E, plus a logarithmic term in n. The second term of the sum decays w.r.t p at the speed of the eigenvalue decay rates of $\mathbb{E}_x[h^*(x) \otimes h^*(x)]$, modulo an exponent $1-\alpha$. Finally, the condition $\mu_{p+1}(M) \geq c_8 n^{-\frac{1}{\beta+1}}$ stems from the estimation error of P, and can translate into the existence of a plateau threshold p^* from which the second term cannot decrease anymore (See Rudi et al. (2013)). Hence, the stronger is Assumption 4.3, the faster is the estimation of $\hat{P}$ and the divergence rate of the plateau threshold. We give here a sketch of the proof for the Theorem 4.7. The complete proof is detailed in Appendix 6.2.

Sketch of the proof. The proof consists in decomposing the excess-risk of the estimator $\hat{P}\hat{h}$ as follows.

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \underbrace{\mathbb{E}_x[\|\hat{P}\hat{h}(x) - \hat{P}h^*(x)\|_{\mathcal{Z}}^2]^{1/2}}_{\text{regression error on a subspace}} + \underbrace{\mathbb{E}_x[\|\hat{P}h^*(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2}}_{\text{reconstruction error}}. \quad (4.14)$$

Then each right-hand term is bounded using a dedicated lemma given in the Appendix 6.2. Lemma 6.5 bounds the regression error on the subspace defined by $\hat{P}$ (akin to a variance). Lemma 6.9 bounds the reconstruction error (akin to a bias). We exploit techniques and schemes similar to those used in (Rudi et al., 2013; Rudi and Rosasco, 2017; Ciliberto et al., 2016, 2020; Luise et al., 2019) in order to prove these lemmas. Namely, L^2-norms of functions in $\mathcal{H}$ are expressed as Hilbert-Schmidt norms of Hilbert-Schmidt operators in $\mathcal{Z} \otimes \mathcal{H}_x$. Relevant norms decompositions lead to study the deviation of the sample operators from the true operators $\mathbb{E}[z \otimes \phi(x)]$ and $\mathbb{E}[\phi(x) \otimes \phi(x)]$. For this purpose, Bernstein's inequalities for the operator norm, or the Hilbert-Schmidt norm, of random operators between separable Hilbert spaces are applied (Tropp, 2012). The previously introduced assumptions of Section 4.3.1 play an important role in the proof of Lemma 6.9, allowing to obtain faster learning rate for $\hat{P}$.

Remark 4.3.1 (Independence assumption on $\phi(x)$ and ϵ)**.** *In this work, we assume that $\phi(x)$ is independent of ϵ. This allows to keep a clear exposition of the proofs, by performing lighter mathematical derivations. Nevertheless, such assumptions is not exploited by the proposed method, and similar results hold without this assumption as we discuss in Appendix 6.2.7.*

4.3.3 Polynomial Eigenvalue Decay Rates

In this subsection, we discuss under which setting reduced-rank ridge regression can be statistically and computationally advantageous in comparison to standard full-rank ridge regression. For this purpose, we apply Theorem 4.7 considering polynomial eigenvalue decay rates for M and E.

Assumption 4.8 (polynomial eigenvalue decay rates)**.** *M and E have polynomial eigenvalue decay rates with parameter $s > 1$ and $e > 1$, if there exist constants $a, A, b, B > 0$ such that:*

$$ap^{-s} \leq \mu_p(M) \leq Ap^{-s}, \quad (4.15)$$

$$bp^{-e} \leq \mu_p(E) \leq Bp^{-e}. \quad (4.16)$$

Parameters s and e characterize the shapes of the signal's and noise's distributions in $\mathcal{Z}$, and provide information complementary to the total amounts of variance $\mathrm{Tr}(M)$ and $\mathrm{Tr}(E)$. Moreover, notice that Assumption 4.8 does not require an exact polynomial decay of the eigenvalues $\mu_k \propto k^{-r}$. In particular, one can define a measure of distortion of $\mu_k(M)$ and $\mu_k(E)$ from exact polynomial decays as the values $\frac{A}{a}$ and $\frac{B}{b}$, respectively. The greater are these ratios the greater are the distortions.

Remark 4.3.2 (Assumptions relationship)**.** *Assumption 4.8 implies that Assumption 4.2 holds with* $c_1 = \mathrm{Tr}(M^{\frac{2}{s}})$*, and Assumption 4.4 holds with* $\gamma = 1 - \frac{e}{s}$ *and* $c_3 = A^{e/s} b^{-1}$.

Under the Assumptions 4.1, 4.3, and 4.8 we derive the following corollary from Theorem 4.7 in the special case of polynomial eigenvalue decay rates.

Corollary 4.9 (Learning bounds (polynomial decay rates))**.** *Let* $\delta \in]0,1]$, $n \geq n_0$. *Under Assumptions 4.1, 4.3, and 4.8, assuming* $\frac{B}{b} \leq \theta$ *with* $\theta \geq 1$, *then by taking only*

$$p = c_9 (\log^8(\frac{8}{\delta}))^{-\frac{1}{s}} n^{\frac{1}{(\beta+1)s}}, \tag{4.17}$$

we have with probability at least $1 - 3\delta$*:*

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq c_{10}(s,e) \log^{5/4}(\frac{n}{\delta}) n^{-1/4} + c_{11}(e)\, n^{-\frac{1}{2}\frac{1-2/s}{1+\beta}} \log^8(\frac{8}{\delta}), \tag{4.18}$$

where $c_{10}(s,e) = \tilde{c}_{10}\left(\frac{e(e-1)}{s}\right)^{1/4}\left(1 + \log\left(\frac{e}{e-1}\right)\right)$, $c_{11}(e) = \tilde{c}_{11}\left(1 + \log\left(\frac{e}{e-1}\right)\right)$. $\tilde{c}_{10}$, $\tilde{c}_{11}$, n_0, *are constants independent of* n, δ, s, e, *and* c_9 *is a constant independent of* n, δ, *defined in the proofs.*

As a first remark, note that the chosen components number p of order $\mathcal{O}(n^{\frac{1}{(\beta+1)s}})$ is significantly smaller than n when s is big (concentrated signal). For instance, $s = 2$ yields at most to $p = O(\sqrt{n})$. Then, notice that the bound is the sum of two terms. The first term is decaying in $O(n^{-1/4})$ modulo a logarithm term in n, and its multiplicative constant can be arbitrarily small when e is small (spread noise), as $c_{10}(s,e) \xrightarrow[e \to 1^+]{} 0$. The decreasing rate of the second term varies within the open interval $]0, 1/2[$. The greater is s and the smaller is β, the better is the rate.

Comparison with full-rank estimator's bound. The bound provided in Eq. (4.18) sheds light on the role of M and E's shapes, flat ($s, e \to 1^+$) or concentrated ($s, e \to +\infty$), in the performance of the reduced-rank estimator. At the opposite, remark that the full-rank ridge estimator's bound is dominated by a term of the form $c(\kappa + \|H\|_{\mathrm{HS}})\mathrm{Tr}(E) n^{-1/4} \log(\frac{4}{\delta})$ with $c > 0$ a constant independent of n, δ, s, e (See Lemma 6.10). So, the ridge estimator is not impacted by the shapes of M and E, but is only affected by the total amounts of signal $\|H\|_{\mathrm{HS}}$, and noise $\mathrm{Tr}(E)$.

Favorable settings for reduced-rank. Which situations are favorable to the proposed reduced-rank method? To simplify the discussion, let us not consider the terms $(1 + \log(e/(e-1)))$ appearing in c_{10}, c_{11}. If s is big enough and β small enough then the right term of (4.18) is $o(n^{-1/4})$ (e.g. $s = 6$, $\beta = 0$ gives $\mathcal{O}(n^{-1/3})$). So, for n big enough, it remains to compare the left term of the bound with the dominating term of the ridge bound. When e becomes close to 1^+ the left term can be arbitrarily smaller than the

ridge bound, because $c_{10}(s,e) \to 0$, while $c\,\mathrm{Tr}(E)$ is unchanged. Let be $q \in \mathbb{N}^*$. For the following family of settings:

$$\beta < 1 - \frac{4}{s}, \qquad e \in]1, e^*(n,q)] \tag{4.19}$$

with $e^*(n,q) = \sup\{e / c_{10}(s,e) < \frac{c\,\mathrm{Tr}(E)^{1/4}}{q \log^{5/4}(n)}\}$, the reduced-rank bound is q times smaller than the full-rank one, when n is big enough.
This gain is obtained because the projection yields to an important noise reduction and a small increase in bias. This can be think as a direct generalization of the low-rank regression setting.

In the following corollary, we duly show that, despite the $(1 + \log(e/(e-1)))$ terms, one can find settings $(n,s,e) \in \mathbb{N}^* \times \mathbb{R}^+ \times \mathbb{R}^+$ such that the learning bound (4.18) is arbitrarily smaller than the kernel ridge estimator's one under the same assumptions on the learning problem.

Corollary 4.10 (Statistical gain of reduced-rank regression)**.** *Let $\delta \in]0,1]$ and $\epsilon > 0$. If $\beta < 1$, then there exists a setting $s, e > 1$, $n \in \mathbb{N}^*$, such that, under the assumptions of Corollary 4.9, with probability at least $1 - 3\delta$,*

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{\frac{1}{2}} \leq \epsilon \times \mathrm{Tr}(E)^{1/4} \times n^{-1/4}. \tag{4.20}$$

Proof We exhibit such a setting (n,s,e). We choose (s,β) such that $\beta < 1 - \frac{4}{s}$. One can check that in this case $c_{11} n^{-\frac{1}{2}\frac{1-2/s}{1+\beta}} \log(n/\delta) = o(n^{-1/4})$, and also $c_{10}\left(\frac{e\theta}{\zeta(e)s} \times \log^5(\frac{n}{\delta})\right)^{1/4} n^{-1/4} = o(n^{-1/4})$ (when $e \to 1^+, n \to +\infty$, with $e \geq 1 + \frac{1}{n^a}$ for any $a > 0$). So, taking n big enough we obtain the desired inequality. ∎

Corollary 4.10 shows that a significant statistical gain is possible using reduced-rank regression, even if the support of $h^*(x)$ covers the entire output space $\mathcal{Z}$, i.e. beyond the standard low-rank setting. Besides the statistical gain, reducing the rank of the predictions' space is of interest for reducing the computational complexity at prediction time.

As it will be presented in the application to structured prediction (See Section 4.4), decoding predictions in surrogate approaches or simply computing mean squared errors require to calculate inner products between the predictions provided by the regression estimator and elements of the output space. In the following lemma, we analyze the complexity in time of such computations. Note that the same complexity holds for computing distances between predictions and elements of the output space. We consider the setting where the dimension of $\mathcal{Z}$ is bigger than n (e.g. infinite).

Corollary 4.11 (Computational gain of reduced-rank regression)**.** *Let $\hat{h} : \mathcal{X} \to \mathcal{Z}$ be a kernel ridge estimator trained on n points. Let $\hat{P} : \mathcal{Z} \to \mathcal{Z}$ be a projection operator of rank p. Given N output points $(z_i)_{i=1}^N$, computing the inner products $\left(\langle \hat{P}\hat{h}(x), z_i \rangle_{\mathcal{Z}}\right)_{i=1}^N$ has a time and space complexity of order $\mathcal{O}(p(N+n))$ while computing the inner products $\left(\langle \hat{h}(x), z_i \rangle_{\mathcal{Z}}\right)_{i=1}^N$ has a time complexity $\mathcal{O}(nN)$.*

Proof In order to compute $\left(\langle \hat{P}\hat{h}(x), z_i\rangle_{\mathcal{Z}}\right)_{i=1}^{N}$ one needs to compute

$$\underbrace{\alpha(x)^T}_{(1,n)}\underbrace{(UZ_{tr})^T}_{(n,p)}\underbrace{UZ}_{(p,N)} \tag{4.21}$$

with $\alpha(x) = (K + n\lambda I)^{-1}k_x$, $k_x = (k(x, x_1), \ldots, k(x, x_n))$, $U = \sum_{i=1}^{p} e_i \otimes u_i$, where $(u_i)_{i=1}^{p}$ is an orthogonal basis of the range of $\hat{P}$, $(e_i)_{i=1}^{p}$ an orthogonal basis of $\mathbb{R}^p$, and Z_{tr} is the operator with the n training output points as columns, z the operator with the N output points as columns. This costs $p(N + n)$ in time and space complexity. In order to compute the $\left(\langle \hat{h}(x), z_i\rangle_{\mathcal{Z}}\right)_{i=1}^{N}$ one needs to compute

$$\underbrace{\alpha(x)^T}_{(1,n)}\underbrace{K^z}_{(n,N)} \tag{4.22}$$

with K^z the gram matrix between the n training points and N output points for the kernel $k_z(z, z') = \langle z, z'\rangle_{\mathcal{Z}}$. This costs nN in time and space complexity. ■

Corollary 4.11 shows that a significant computational gain is possible when $N \gg p$ and $n \gg p$, as in this case $p(N+n) \ll nN$. Combining this result with Corollary 4.10 we conclude that, under the output regularity assumptions made, the proposed method offers both statistical and computational gains by projecting the ridge estimator onto an estimated linear subspace.

Remark 4.3.3 (Consequences for finite dimensional $\mathcal{Z}$)**.** *The obtained results are not limited to the infinite dimensional setting and are still valuable when $\mathcal{Z} = \mathbb{R}^d$. One can notice that in the finite dimensional case Assumptions 4.2, 4.3, and 4.4 are always verified choosing the best exponents $\alpha = \beta = 0, \gamma = 1$ (if $M, E > 0$), but it is at the price of very large constants c_1, c_2 and very small c_3, which make the bounds very large. In fact, it amounts to using the rough inequalities $\mathrm{Tr}(A) \leq d \times \|A\|_\infty$ and $A \preceq \frac{\mu_1(B)}{\mu_d(B)}B$ for any bounded operators A, B, thereby loosing information on the shape of M and E. At the opposite, choosing α, β, γ such that the constants c_1, c_2, c_3 remain close to 1 allows to obtain finer bounds, taking into account the signal/noise configuration, closed to the observed behaviors.*

Take-home message. The proposed reduced-rank regression estimator enjoys a statistical gain under more general assumptions than standard low-rank assumptions. As parameter λ, the rank p acts as a regularization parameter whose impact should disappear when the size of the training sample increases, i.e. $p \xrightarrow[n\to+\infty]{} +\infty$. The settings where the proposed method performs better than the kernel ridge estimator require faster eigenvalue decay rates for $\mathbb{E}[h^*(x)\otimes h^*(x)]$ than for $\mathbb{E}[\epsilon\otimes\epsilon]$ (concentrated signal/diffuse noise). But this is not sufficient: Assumption 4.3 with a sufficiently small β ($\beta < 1 - \frac{4}{s}$) is also necessary to ensure a fast enough estimation of P. Last but not least, reducing the predicted outputs' dimension can also yield to substantial computational gains.

4.4 Application to structured prediction

In this section, we develop an application of the reduced-rank estimator to structured prediction. The novel method fits into the generic framework of surrogate approaches for structured prediction and exploits an infinite dimensional embedding by the mean of a kernel. We describe the algorithm and give learning bounds for the proposed structured prediction estimator.

4.4.1 Surrogate Reduced-Rank Estimator for Structured Prediction

Structured prediction consists in solving a supervised learning task where the output variable is a structured object. Denoting $\mathcal{Y}$ the structured output space, a structured loss $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ measures the discrepancy between a true output and a predicted output. The goal of structured prediction is to minimize the following expected risk:

$$\mathcal{R}_\Delta(f) = \mathbb{E}_\rho[\Delta(f(x), y)], \tag{4.23}$$

over a class of functions $f : \mathcal{X} \to \mathcal{Y}$, using a finite set $(x_i, y_i)_{i=1}^n$ independently drawn from an unknown distribution ρ on $\mathcal{X} \times \mathcal{Y}$. In other words, if we note $f^* : \mathcal{X} \to \mathcal{Y}$ the minimizer of $\mathcal{R}_\Delta(f)$, the aim of learning is therefore to get an estimator $\hat{f}$ of f^* based on the finite sample $(x_i, y_i)_{i=1}^n$ with the best possible statistical properties.

A surrogate approach: Output Kernel Regression We consider here the case when Δ is defined as a metric induced by a positive definite kernel k_y acting over the structured output space $\mathcal{Y}$:

$$\Delta(y, y') = \|\psi(y) - \psi(y')\|^2_{\mathcal{H}_y}. \tag{4.24}$$

This boils down to embedding objects of $\mathcal{Y}$ into the Reproducing Kernel Hilbert Space associated to k_y using the canonical feature map $\psi : \mathcal{Y} \to \mathcal{H}_y$ associated to k_y, and then consider the square loss over $\mathcal{H}_y$. Relying on the abundant literature about kernels on structured objects (Gärtner, 2003), this class of losses covers a wide variety of structured prediction problems.

However, learning directly f through ψ still raises an issue and a simple way to overcome it consists in seeking instead a **surrogate** model $h : \mathcal{X} \to \mathcal{H}_y$ able to predict the embedded objects in the infinite dimensional space $\mathcal{H}_y$ and leverage the kernel trick in the output space. This approach is referred as Output Kernel Regression (OKR) (Weston et al., 2003; Geurts et al., 2006; Brouard et al., 2016b). The original structured prediction problem is then replaced by the following surrogate vector-valued regression problem stated in terms of the surrogate true risk:

$$\min_{h:\mathcal{X}\to\mathcal{H}_y} \mathbb{E}_\rho[\|h(x) - \psi(y)\|^2_{\mathcal{H}_y}]. \tag{4.25}$$

Assume h^* is the function $x \to \mathbb{E}_y[\psi(y)|x]$ (solution of Eq. (4.25)). Then at prediction time, one can retrieve a prediction in the original space $\mathcal{Y}$ through an appropriate decoding function $d : \mathcal{H}_y \to \mathcal{Y}$:

$$y^* = f^{**}(x) := d \circ h^*(x) := \arg\min_{y\in\mathcal{Y}} \|h^*(x) - \psi(y)\|^2_{\mathcal{H}_y}. \tag{4.26}$$

The overall approach is illustrated on Fig. 4.2.

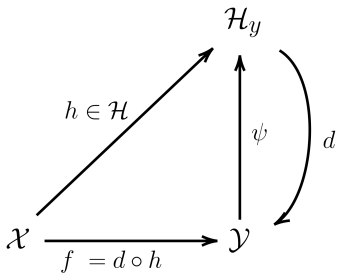

Figure 4.2: Schematic illustration of OKR.

Ciliberto et al. (2016) have proved that f^{**} solves exactly the original structured prediction problem, i.e. $f^{**} = f^*$. Fo that purpose, they have shown that $\Delta(y, y') = \|\psi(y) - \psi(y')\|^2_{\mathcal{H}_y}$ belongs to the wide family of Structure Encoding Loss Functions (SELF), as it can be written $y, y' \to \langle \gamma(y), \theta(y') \rangle_\nu$ with $\gamma(y) = (\sqrt{2}\psi(y), \|\psi(y)\|^2_{\mathcal{H}_y}, 1)$, $\gamma(y) = (-\sqrt{2}\psi(y'), 1, \|\psi(y')\|^2_{\mathcal{H}_y})$, and $\nu = \mathcal{H}_y \oplus \mathbb{R} \oplus \mathbb{R}$.

Moreover, when providing an estimator $\hat{h}$ of h^* using the training sample $(x_i, y_i)_{i=1}^n$, we benefit from the so called comparison inequality from Ciliberto et al. (2016)

$$\mathcal{R}_\Delta(\hat{f}) - \mathcal{R}_\Delta(f^*) \leq c \times \mathbb{E}_x[\|\hat{h}(x) - h^*(x)\|^2_{\mathcal{H}_y}]^{1/2}, \tag{4.27}$$

where $\hat{f} = d \circ \hat{h}$ and the constants c and Q are defined as: $c = 2\sqrt{2Q^2 + Q^4 + 1}$, and $Q = \sup_y \|\psi(y)\|_{\mathcal{H}_y}$.

Reduced-rank regression in structured prediction. The OKR problem depicted in Eq. (4.25) can be solved in various hypothesis spaces and trees-based approaches (Geurts et al., 2006) as well as kernel methods (Weston et al., 2003; Geurts et al., 2006; Brouard et al., 2011; Kadri et al., 2013; Laforgue et al., 2020) have been developed so far to tackle it. We focus here on Input Output Kernel Regression (IOKR), a method that exploits operator-valued kernels (Brouard et al., 2016b) and assumes that h belongs to a vv-RKHS. In particular, IOKR-ridge solves the kernel ridge regression problem in Eq. (4.4) with the following choice s: the output space is $\mathcal{Z} := \mathcal{H}_y$, the chosen operator-valued kernel writes as $K(x, x') = k(x, x')I_{\mathcal{H}_y}$,and the hypothesis space$\mathcal{H}$ is the vv-RKHS associated to K. Instantiating Eq. 4.5, the solution to IOKR-ridge writes as:

$$\hat{h}(x) = \sum_{i=1}^{n} \alpha_i(x)\psi(y_i), \tag{4.28}$$

where α_i's are defined according Eq. 4.5.

In this section, we propose to solve the surrogate problem in Eq. (4.25) using our reduced-rank estimator based on the IOKR-ridge estimator. This gives rise to the definition of a novel structured output prediction $\hat{f}$:

$$\hat{f}(x) := \underset{y \in \mathcal{Y}}{\arg\min} \|\hat{P}\hat{h}(x) - \psi(y)\|^2_{\mathcal{H}_y}. \tag{4.29}$$

Because of the comparison inequality Eq. (4.27), the resulting structured predictor directly benefits from the learning bound on the least-squares problem.

Theorem 4.12 (Excess-risk bound for the structured predictor)**.** *Let $\delta \in]0,1]$, $n \geq n_0$. Under Assumptions 4.1, 4.3, and 4.8, assuming $\frac{B}{b} \leq \theta$ with $\theta \geq 1$, then by taking only*

$$p = c_9(\log^8(\frac{8}{\delta}))^{-\frac{1}{s}} n^{\frac{1}{(\beta+1)s}} \tag{4.30}$$

then with probability at least $1-3\delta$

$$\mathcal{R}_\Delta(\hat{f}) - \mathcal{R}_\Delta(f^*) \leq c \times \left(c_{10}(s,e) \log^{5/4}(\frac{n}{\delta}) n^{-1/4} + c_{11}(e) n^{-\frac{1}{2}\frac{1-2/s}{1+\beta}} \log^8(\frac{8}{\delta}) \right) \tag{4.31}$$

where $c_{10}(s,e) = \tilde{c}_{10}\left(\frac{e(e-1)}{s}\right)^{1/4}\left(1+\log\left(\frac{e}{e-1}\right)\right)$, $c_{11}(e) = \tilde{c}_{11}\left(1+\log\left(\frac{e}{e-1}\right)\right)$. $\tilde{c}_{10}$, $\tilde{c}_{11}$, n_0, are constants independent of n, δ, s, e and c_9 is a constant independent of n, δ, defined in the proofs.

The bound provided in Theorem 4.12 is similar to the one of Corollary 4.9 modulo the multiplicative constant c, and thus the interpretation is the same. In particular, when s is sufficiently big and e, β sufficiently small, we can obtain a significant statistical gain in comparison to the not projected estimator, as shown in Corollary 4.10.

4.4.2 Algorithms and Complexity Analysis

To define the final reduced-rank IOKR-ridge estimator $\hat{f}$, one has to apply Algorithm 4.1 to compute all the parameters of $\hat{P}\hat{h}$ necessary to the decoding phase described in Algorithm 4.2.

Complexity in time At decoding/prediction time, one needs to compute n_{te} times the prediction $\hat{f}(x_i)$, for the testing data points $(x_i)_{i=1}^{n_{te}}$. Each prediction requires to calculate the distances in Eq. (4.26). This is made possible by using the kernel trick, avoiding to compute the infinite dimensional vectors $\hat{h}(x)$ and $\psi(y)$. These computations cost $\mathcal{O}(n_{te} n |\mathcal{Y}|)$ in time, where n and $|\mathcal{Y}| \in \mathbb{N}^*$ are the size of the training data set and the number of output candidates, respectively. Note that $|\mathcal{Y}|$ is typically very big in structured prediction. For instance, in multilabel classification with d labels $|\mathcal{Y}| = \{0,1\}^d = 2^d$. In practice, one often chooses a subset of $\mathcal{Y}$ as a candidate set. Hence, the decoding phase badly scales with n, and in general is computationally expensive. Because of the projection onto a finite dimensional space, the proposed method can significantly alleviate these computations. When using $\hat{P}\hat{h}$ with $\hat{P}$ of rank p, the decoding time complexity reduces to $\mathcal{O}(n_{te} p |\mathcal{Y}|)$ as shown in Corollary 4.11. Furthermore, the training phase consists in a matrix inversion for computing $\hat{h}$ plus a singular value decomposition for computing $\hat{P}$. Hence, the time complexity of the training algorithm without approximation is $\mathcal{O}(2n^3)$. It can still be reduced using efficient and theoretically grounded approximation methods for KRR and kernel principal component analysis developed in (Rudi et al., 2015; Rudi and Rosasco, 2017; Sterge et al., 2020).

Algorithm	IOKR	Reduced-rank IOKR
Training	$\mathcal{O}(n^3)$	$\mathcal{O}(2n^3)$
Decoding	$\mathcal{O}(n_{te} n \lvert\mathcal{Y}\rvert)$	$\mathcal{O}(n_{te} p \lvert\mathcal{Y}\rvert)$

Table 4.2: Time complexity of IOKR versus reduced-rank IOKR.

Algorithm 4.1 Reduced-rank IOKR-ridge - Training phase

Input: $K_x, K_y \in \mathbb{R}^{n\times n}$, $\lambda \geq 0$, $p \in \mathbb{N}^*$
KRR estimation: $W = (K_x + n\lambda I)^{-1} \in \mathbb{R}^{n\times n}$
Subspace estimation:
$K_h = WK_xK_yK_xW \in \mathbb{R}^{n\times n}$
$\beta = \begin{bmatrix} | & & | \\ \frac{u_1}{\sqrt{\mu_1}} & \cdots & \frac{u_p}{\sqrt{\mu_p}} \\ | & & | \end{bmatrix} \in \mathbb{R}^{n\times p} \leftarrow SVD(K_h) = \sum_{l=1}^{n} \mu_l u_l u_l^T$
Training outputs projection:
$K_{yh} = K_y W K_x \in \mathbb{R}^{n\times n}$
$UY = K_{yh}\beta \in \mathbb{R}^{n\times p}$
Return: W (KRR coefficients), β (projection coefficients), UY (projected training outputs)

Algorithm 4.2 Reduced-rank IOKR-ridge - Decoding phase

Input: $k_x^{te} \in \mathbb{R}^n$, $Y_{candidates} \in \mathbb{R}^{n_c\times d}$, $UY \in \mathbb{R}^{n\times p}$, $W \in \mathbb{R}^{n\times n}$
Output candidates projection:
$K_{yh} = WK_xK_y^{tr/c} \in \mathbb{R}^{n\times n_c}$
$UY_c = K_{yh}\beta \in \mathbb{R}^{n_c\times p}$
Distances computation:
$\alpha = Wk_x^{te} \in \mathbb{R}^n$
$Uh_{te} = UY^T\alpha \in \mathbb{R}^p$
$S := "\langle \hat{P}\hat{h}(x_{te}), \psi(Y_{candidates})\rangle_{\mathcal{H}_y}" = (Uh_{te})^T UY_c \in \mathbb{R}^{n_c}$
$N := "\|\psi(Y_{candidates})\|^2_{\mathcal{H}_y}" = \Big(K_y(y,y)\Big)_{y\in Y_{candidates}} \in \mathbb{R}^{n_c}$
$D = N - 2S$
1-NN prediction :
$\hat{i} = \arg\min_{i\in[1,n_c]} D_i$
$\hat{y} = Y_{candidates}[\hat{i}] \in \mathcal{Y}$
Return: $\hat{y}$ (prediction)

4.5 Numerical experiments

We now carry out experiments with the methods proposed in this work. In Section 4.5.1, we illustrate our theoretical insights on synthetic least-squares problems. In Section 4.5.2, we test the proposed structured prediction method on three different problems: image reconstruction, multi-label classification, and metabolite identification.

4.5.1 Reduced-rank regression: statistical gain and importance of Assumption 4.3

We illustrate, on synthetic least-squares problems, the theoretical insights, given in Subsection 4.3.3. For $d = 300$, $\mathcal{X} = \mathcal{H}_x = \mathcal{Z} = \mathbb{R}^d$, we choose $\mu_p(C) = \frac{1}{\sqrt{p}}$, $\mu_p(E) = \frac{0.2}{p^{1/10}}$. We draw randomly the eigenvector associated to each eigenvalue. We draw $H_0 \in \mathbb{R}^{d\times d}$ with independently drawn coefficients from the standard normal distribution. We consider two different optimums $H = H_0$ ($\beta = 1$) and $H = (H_0CH_0)H_0$ ($\beta = 1/3$). Then,

we generate $n \in [10^2, \ldots, 5 \times 10^3], n_{val} = 1000, n_{test} = 1000$ couples (x, z) such that $x \sim \mathcal{N}(0, C)$, $\epsilon \sim \mathcal{N}(0, E)$, and $z = Hx + \epsilon$. We select the hyper-parameters of the three estimators $\hat{h}$, $P\hat{h}$, and $\hat{P}\hat{h}$ in logarithmic grids, with the best validation MSE. On the Figure 4.3 we plot the test MSE obtain by the three estimators for various p and n, and for the two different optimums $H = H_0$ (left) and $H = (H_0 C H_0) H_0$ (right). There exists for both H (left/right) a minimum MSE w.r.t p for $P\hat{h}$ below the MSE of $\hat{h}$ when n is big enough: P offers a valuable regularization of $\hat{h}$. Moreover, we observe that the selected p increases when n increases with a decreasing gain, following the provided bounds' behavior. Furthermore, we observe that because of the estimation error of $\hat{P}$, there is no gain for $\hat{P}\hat{h}$ when $H = H_0$, while when $H = (H_0 C H_0) H_0$ there is a gain for n big enough. This illustrates the faster convergence rate of $\hat{P}$ when β is small.

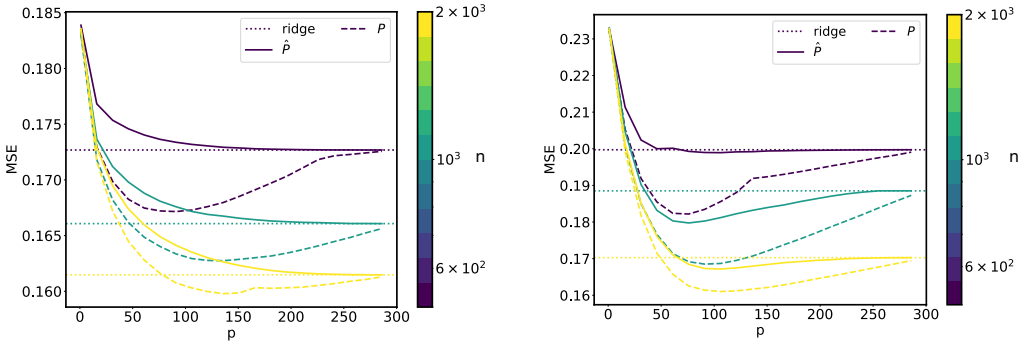

Figure 4.3: Test MSE w.r.t p (x axis) and the quantity of training data n (color bar), obtained with the optimal projection P and its estimation $\hat{P}$, for various output source condition. (Left) Output source condition $\beta = 1$, $H = H_0$. (Right) Output source condition $\beta = 1/3$, $H = (H_0 C H_0) H_0$.

4.5.2 Experiments on Structured Prediction

In this section, we assess the performance of the reduced-rank IOKR estimator calculated using Algorithms 4.1 and 4.2 proposed in Section 4.4 on three real-world structured prediction tasks: image reconstruction, multi-label classification, and metabolite identification. Our experiments show how reduced-rank regression can be advantageously used for surrogate methods in structured prediction in order to improve both statistical and computational aspects. In these experiments, we choose $\lambda_1 = \lambda_2$ in order to reduce the quantity of hyperparameters.

State of the art approaches For each task, we compared our reduced-rank method to relevant existing SOTA approaches. SPEN (Belanger and McCallum, 2016), a neural network learned by minimizing the structured hinge loss, is an Energy-Based Model (EBM), considered as a strong benchmark in the literature. Contrary to surrogate approaches, EBM involves the computation of the decoding phase during the training phase. Kernel Dependency Estimation (KDE) (Weston et al., 2003) shares with IOKR the use of kernels in the input and output space with the following differences: in KDE, Kernal PCA is used to decompose the output feature vectors into p orthogonal directions. Kernel ridge regression is then used for learning independently the mapping between the input feature vectors and each direction. By applying KPCA on the outputs KDE aims at estimating the linear subspace of the output embedding $\psi(y)$

while the proposed reduced-rank estimator aims at estimating the linear subspace of the $h^*(x)$. Additionally, for the multi-label classification problem, we choose the exact setting of previous benchmark experiments (See for instance, (Gygli et al., 2017; Lin et al., 2014)) and thus benefited from the collected results and comparison with other methods.

4.5.2.1 Image Reconstruction

Problem and data set. The goal of the image reconstruction problem provided by Weston et al. (2003) is to predict the bottom half of a USPS handwritten postal digit (16 x 16 pixels), given its top half. The data set contains 7291 training labeled images and 2007 test images.

Experimental setting. As in Weston et al. (2003) we used as target loss an RBF loss $\|\psi(y) - \psi(y')\|^2_{\mathcal{H}_y}$ induced by a Gaussian kernel k and visually chose the kernel's width $\sigma^2_{output} = 10$, looking at reconstructed images of the method using the ridge estimator (i.e. without reduced-rank estimation). We used a Gaussian input kernel of width σ^2_{input}. For the pre-image step, we used the same candidate set for all methods constituted with all the 7291 training bottom half digits. We considered $\lambda := \lambda_1 = \lambda_2$ for the proposed method. The hyper-parameters for all tested methods (including $\sigma^2_{input}, \lambda, p$, and SPEN layers' sizes) have been selected using logarithmic grids via 5 repeated random sub-sampling validation (80%/20%).

Reduced-rank estimator for surrogate problem. We start by evaluating the performance of the reduced-rank estimator in solving the Hilbert space valued least-squares problem described in Eq. (4.25). We plot on Figure 4.4 the test mean squared error of our estimator, and of the ridge estimator, w.r.t the quantity of training data n from $n = 500$ to $n = 7000$. We observe that the reduced-rank estimator ($p < +\infty$) always performs better than the kernel ridge estimator ($p = +\infty$). Nevertheless, we see that this gain is smaller for small n or big n. This is a typical behavior observed in our experiments, which can be interpreted as a difficulty in estimating $\hat{P}$ when n is small, and the diminishing usefulness of regularization when n increase. Indeed p can be thought of as a regularization parameter exploiting a different regularity assumption than λ, but whose action, similarly to λ, should decrease when n increases, such that $p \to +\infty$ when $n \to +\infty$.

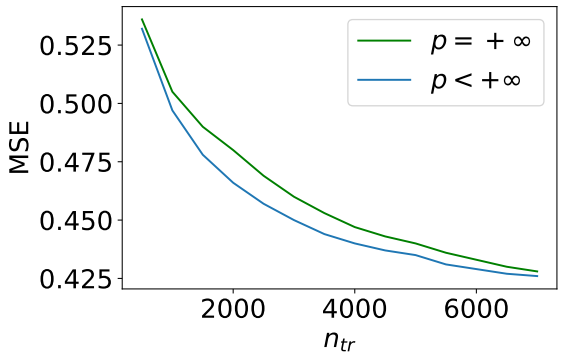

Figure 4.4: Test MSE of the proposed reduced-rank estimator ($p < +\infty$), and of the ridge estimator ($p = +\infty$) w.r.t n on the USPS problem.

Comparison with SOTA methods. Then, in a second experiment, we compare the structured predictor (see Eq. (4.29)) using reduced-rank estimation, to state-of-the-art methods: SPEN (Belanger and McCallum, 2016), IOKR (Brouard et al., 2016b), and Kernel Dependency Estimation (KDE) (Weston et al., 2003). We fix $n = 1000$ where the reduced-rank estimation seems helpful, according to Figure 4.4. For SPEN we employed the standard architecture and training method described in the corresponding article (cf. supplements for more details). We evaluated the results in term of RBF loss (e.g. Gaussian kernel loss), as in Weston et al. (2003). The obtained results are given in Table 4.3. Firstly, we see that SPEN obtains worse results than KDE, IOKR, and reduced-rank IOKR. Furthermore, note that the number of hyperparameters for SPEN (architecture and optimization) is usually larger than reduced-rank IOKR. Finally, notice that IOKR correspond to the proposed method with $p = +\infty$. Hence, this shows the benefit of exploiting output regularity thanks to reduced-rank estimation in structured prediction.

Method	RBF loss	p
SPEN	0.801 ± 0.011	128
KDE	0.764 ± 0.011	64
IOKR	0.751 ± 0.011	∞
Reduced-rank IOKR	$\mathbf{0.734} \pm 0.011$	64

Table 4.3: Test mean losses and standard errors for the proposed method, IOKR, KDE, and SPEN on the USPS digits reconstruction problem where $n = 1000$, and $n_{test} = 2007$.

4.5.2.2 Multi-label Classification

Problem and data set. Bibtex and Bookmarks (Katakis et al., 2008) are tag recommendation problems, in which the objective is to propose a relevant set of tags (e.g. url, description, journal volume) to users when they add a new Bookmark (webpage) or Bibtex entry to the social bookmarking system Bibsonomy. Corel5k is an image data set and the goal of this application is to annotate these images with keywords. Information on these data sets is given in Table 4.4.

data set	n	n_{te}	$n_{features}$	n_{labels}	$\bar{l}$
Bibtex	4880	2515	1836	159	2.40
Bookmarks	60000	27856	2150	208	2.03
Corel5k	4500	499	37152	260	3.52

Table 4.4: Multi-label data sets description. $\bar{l}$ denotes the averaged number of labels per point.

Experimental setting. For all multi-label experiments we used a Gaussian input and output kernels with widths σ^2_{input} and $\sigma^2_{output} = \bar{l}$, where $\bar{l}$ is the averaged number of labels per point. As candidate sets we used all the training output data. We measured the quality of predictions using example-based F1 score. We selected the hyper-parameters λ and p in logarithmic grids.

Comparison with SOTA methods. We compare our method with several multi-label and structured prediction approaches including IOKR (Brouard et al., 2016b), logistic regression (LR) trained independently for each label (Lin et al., 2014), a two-layer neural network with cross entropy loss (NN) by (Belanger and McCallum, 2016), the multi-label approach PRLR (Posterior-Regularized Low-Rank) (Lin et al., 2014), the energy-based model SPEN (Structured Prediction Energy Networks) (Belanger and McCallum, 2016) as well as DVN (Deep Value Networks) (Gygli et al., 2017). The results in Table 4.5 show that surrogate methods (first two lines) can compete with state-of-the-art dedicated multilabel methods on the standard data sets Bibtex and Bookmarks. With Bookmarks (n/n_{te} = 60000/27856) we used a Nyström approximation with 15000 anchors when computing $\hat{h}$ to reduce the training complexity, and we learned $\hat{P}$ only with a subset of 12000 training data. $\hat{h}$ decoding took about 56 minutes, and $\hat{P}\hat{h}$ decoding less than 4 minutes. With a drastically smaller amount of time, $\hat{P}\hat{h}$ (first line) achieves the same order of magnitude of F1 as $\hat{h}$ (line two) at a lower cost (see Table 4.6) and still has better performance than all other competitors.

Method	Bibtex	Bookmarks
Reduced-rank IOKR	43.8	39.1
IOKR	44.0	**39.3**
LR	37.2	30.7
NN	38.9	33.8
SPEN	42.2	34.4
PRLR	44.2	34.9
DVN	**44.7**	37.1

Table 4.5: Tag prediction from text data. F_1 score of reduced-rank IOKR compared to state-of-the-art methods. LR (Lin et al., 2014), NN (Belanger and McCallum, 2016), SPEN (Belanger and McCallum, 2016), PRLR (Lin et al., 2014), DVN (Gygli et al., 2017). Results are taken from the corresponding articles.

	IOKR	Reduced-rank IOKR
Bibtex	2s/13s	15s/4s
Bookmarks	465s/3371s	617s/214s
USPS	0.1s/9s	0.4s/1s

Table 4.6: Fitting/Decoding computation time of IOKR compared to our method (in seconds)

Small training data regime. We evaluate the reduced-rank structured predictor in a setting where only a small number of training examples is known. For this setting, we consider only the 2000 first couples (x_i, y_i) of each multi-label data set as training set. Hyper-parameters have been selected using 5 repeated random sub-sampling validation (80%/20%) and the same λ was used for IOKR. The results of this comparison are given in Table 4.7. We observe that the proposed reduced-rank structured predictor obtains higher F1 scores than the one using kernel ridge regression in this setup. This highlights the interest of our method in a setting where the data set is small in comparison to the difficulty of the task.

	Bibtex	Bookmarks	Corel5k
n	2000	2000	2000
n_{te}	2515	2500	499
IOKR	35.9	22.9	13.7
Reduced-rank IOKR	**39.7**	**25.9**	**16.1**

Table 4.7: Test F_1 score of reduced-rank IOKR and IOKR on different multi-label problems in a small training data regime.

About the selected rank p. We selected the rank p with integer logarithmic scales, ensuring that the selected dimensions were always smaller than the maximal one of the grids. From Table 4.7 to Table 4.5, the selected dimension p for Bibtex/Bookmarks are 80/30, then 130/200. In Table 4.7 recall that we used a reduced number of training couples. Interpreting p as a regularisation parameter, we see that when n increases then the p increases, i.e. the rank regularisation decreases.

4.5.2.3 Metabolite Identification

Problem and data set. An important problem in metabolomics is to identify the small molecules, called metabolites, that are present in a biological sample. Mass spectrometry is a widespread method to extract distinctive features from a biological sample in the form of a tandem mass (MS/MS) spectrum. The goal of this problem is to predict the molecular structure of a metabolite given its tandem mass spectrum. The molecular structures of the metabolites are represented by fingerprints, that are binary vectors of length $d = 7593$. Each value of the fingerprint indicates the presence or absence of a certain molecular property. Labeled data are expensive to obtain, and despite the problem complexity only $n = 6974$ labeled data are available. State-of-the-art results for this problem have been obtained with the IOKR method by Brouard et al. (2016a). The median size of the candidate sets is 292, and the biggest candidate set is of size 36918. Hence, the metabolite identification data set is characterized by high-dimensional complex outputs, a small training set, and a very large number of candidates.

Experimental setting. We adopt a similar numerical experimental protocol (5-CV Outer/4-CV Inner loops) than in Brouard et al. (2016a), probability product input kernel for mass spectra, and Gaussian-Tanimoto output kernel on the molecular fingerprints (with parameter $\sigma^2 = 1$). We selected the hyper-parameters λ, p in logarithmic grids using nested cross-validation with 5 outer folds and 4 inner folds.

Improved prediction with reduced-rank estimation . We compare the proposed reduced-rank structured predictor with SPEN, and with the state-of-the art method on this problem IOKR (which corresponds to our method with $p = +\infty$). The result are given in Table 4.8. We observe that reduced-rank IOKR improved upon plain IOKR, in this context of supervised learning with complex outputs and a small training data set.

Method	MSE	Tanimoto-Gaussian loss	Top-k accuracies $k=1 \mid k=5 \mid k=10$
SPEN	–	0.537 ± 0.008	25.9%|54.1%|64.3%
IOKR	0.781 ± 0.002	0.463 ± 0.009	29.6%|61.1%|71.0%
Reduced-rank IOKR	$\mathbf{0.766 \pm 0.003}$	$\mathbf{0.459 \pm 0.010}$	**30.0**%|**61.5**%|**71.4**%

Table 4.8: Test mean losses and standard errors for the metabolite identification problem. SPEN MSE in $\mathcal{H}_z$ is not defined as predictions are directly done in $\mathcal{Z}$.

4.6 Extension: leveraging unsupervised output data

In this section, we extend the proposed approach in order to leverage unsupervised output data.

4.6.1 Extended setting and proposed method

Setting. We consider that an additional data set $\mathcal{U}_m = (y_j^u)_{j=1}^m$ of m output data, independently draw from the marginal distribution ρ_y, is given, in addition to the supervised data set $\mathcal{S}_n = (x_i, y_i)_{i=1}^n$. Such data is generally easy to obtain for many structured output problems, including the metabolite identification task described in the experiments.

Extended method. We propose to extend the proposed reduced-rank method proposed above in order to leverage the data set $\mathcal{U}_m$ as follows. Instead of using the projection

$$\hat{P}_{\lambda_1} := \operatorname*{arg\,min}_{\mathsf{P} \in \mathcal{P}_p} \frac{1}{n} \sum_{i=1}^{n} \|\mathsf{P}\hat{h}_{\lambda_1}(x_i) - \hat{h}_{\lambda_1}(x_i)\|_{\mathcal{Y}}^2, \tag{4.32}$$

we use the projection

$$\hat{P}_{\lambda_1} := \operatorname*{arg\,min}_{\mathsf{P} \in \mathcal{P}_p} \frac{c}{n} \sum_{i=1}^{n} \|\mathsf{P}\hat{h}_{\lambda_1}(x_i) - \hat{h}_{\lambda_1}(x_i)\|_{\mathcal{Y}}^2 + \frac{(1-c)}{m} \sum_{j=1}^{m} \|\mathsf{P}\psi(y_j^u) - \psi(y_j^u)\|_{\mathcal{Y}}^2, \tag{4.33}$$

with a chosen $c \in [0, 1]$.

Then, as previously we define the following *extended reduced-rank estimator*:

$$x \to \hat{P}_{\lambda_1} \hat{h}_{\lambda_2} \tag{4.34}$$

The resulting Algorithm is provided just below.

[algo]

4.6.2 Extended theoretical analysis

We only provide a sketch of the proof for obtaining learning bounds for the extended reduced-rank estimator. We leave the derivation of the complete proof, and the analysis of the statistical behavior of the extension for future work.

First notice that the ideal counterpart of the objective Eq. (4.33) is

$$P := \operatorname*{arg\,min}_{\mathsf{P}\in\mathcal{P}_p} c\mathbb{E}[\|\mathsf{P}h^*(x) - h^*(x)\|_{\mathcal{Y}}^2] + (1-c)\mathbb{E}[\|\mathsf{P}\psi(y) - \psi(y)\|_{\mathcal{Y}}^2] \tag{4.35}$$

$$= \operatorname*{arg\,min}_{\mathsf{P}\in\mathcal{P}_p} c\,\mathrm{Tr}((\mathsf{P}-I)M) + (1-c)\,\mathrm{Tr}((\mathsf{P}-I)C_\psi) \tag{4.36}$$

$$= \operatorname*{arg\,min}_{\mathsf{P}\in\mathcal{P}_p} \|(\mathsf{P}-I)M_c^{1/2}\|_{\mathrm{HS}}^2. \tag{4.37}$$

with $M = \mathbb{E}[h^*(x)\otimes h^*(x)]$, $C_\psi = \mathbb{E}[\psi(y)\otimes\psi(y)]$, and $M_c = cM + (1-c)C_\psi$.

Then, one can carry out similar proof than done for the non extended reduced-rank estimator, by measuring the alignments of M_c with M, and with $E = \mathbb{E}[\epsilon\otimes\epsilon]$, via assumptions of the form

$$aM^{1-u} \preceq M_c \qquad bM_c^{1-v} \preceq E \tag{4.38}$$

with $u, v \in [0,1]$, $a, b > 0$.

4.6.3 Numerical experiments

We carry out an experimental study of the proposed extension, by considering the three same problems : image reconstruction, multi-label classification, metabolite identification.

4.6.3.1 Image reconstruction

Experimental setting. We consider the exact same setting as in Table 4.3, but exploiting $m = 6000$ additional outputs as unsupervised data set. In particular, we also select the hyper-parameters (λ, c, p) using logarithmic grids via 5 repeated random sub-sampling validation (80%/20%).

Improved statistical performance. When leveraging the $m = 6000$ unsupervised output data, we obtain an improvement of the test mean loss (See Table 4.9.

Method	RBF loss	p
Reduced-rank IOKR	0.734 ± 0.011	64
Extended Reduced-rank IOKR	$\mathbf{0.725 \pm 0.011}$	98

Table 4.9: Test mean losses and standard errors for Reduced-rank IOKR, and its extended version leveraging $m = 6000$ unsupervised output data.

4.6.3.2 Multi-label classification

Experimental setting. We consider the exact same setting as in Table 4.5, but exploiting additional outputs as unsupervised data set ($m = 2880$ for Bibtex, $m = 4000$ for Bookmarks, $m = 2500$ for Corel5k). In particular, we also select the hyper-parameters (λ, c, p) using logarithmic grids via 5 repeated random sub-sampling validation (80%/20%).

Improved statistical performance. When leveraging the m unsupervised output data, we obtain an improvement of the test mean F1 score (See Table 4.10).

	Bibtex	Bookmarks	Corel5k
n	2000	2000	2000
m	2880	4000	2500
n_{te}	2515	2500	499
Reduced-rank IOKR	**39.7**	25.9	16.1
Extended Reduced-rank IOKR	**39.7**	**27.1**	**19.0**

Table 4.10: Test F_1 score of Reduced-rank IOKR, and its extended version leveraging m unsupervised output data, on different multi-label problems in a small training data regime.

Study of the number of unsupervised data m. We further show the impact of additional unsupervised data on the Bookmarks dataset by training the KRR with only $n = 2000$ data, and training the extended reduced-rank IOKR method with various numbers of unexploited data from 0 to 50000 randomly selected. Figure 4.5 shows that adding unsupervised output data through the right term of Equation (4.33) allows to improve the results up to a certain level.

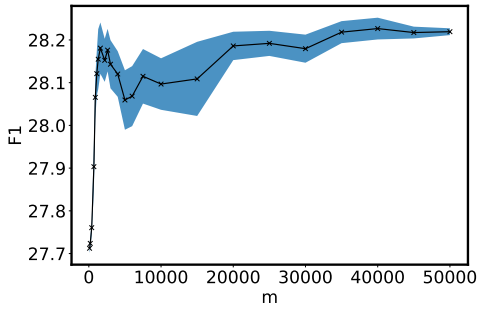

Figure 4.5: Test F1 of extended Reduced-rank IOKR on Bookmarks dataset with $n/n_{te} = 2000/27856$ w.r.t the quantity of randomly selected unsupervised data $m \in [0, 50000]$ used.

4.6.3.3 Metabolite identification

Experimental setting. We consider the exact same setting as in Table 4.8, but exploiting additional outputs as unsupervised data set ($m = 2880$ for Bibtex, $m = 4000$ for Bookmarks, $m = 2500$ for Corel5k). Using randomized singular value decomposition we trained the extended Reduced-rank IOKR with $m = 10^5$ molecular fingerprints, which are not exploited with Reduced-rank and Full-rank IOKR, as the corresponding inputs (spectra) are not known.

Improved statistical performance. When leveraging the $m = 10^5$ unsupervised output data, we obtain an improvement of the test Top-k accuracies (See Table 4.11) upon reduced-rank and full-rank IOKR. The selected balancing parameter by a inner cross-validation on training set is $\hat{c} = 0.75$ in average on the outer splits, imposing a balance between the influence of the small size labeled dataset and the large unsupervised output set.

Method	Gaussian -Tanimoto loss	Top-k accuracies $k = 1 \mid k = 5 \mid k = 10$
IOKR	0.463 ± 0.009	29.6%|61.1%|71.0%
Reduced-rank IOKR	0.459 ± 0.010	30.0%|61.5%|71.4%
Extended Reduced-rank IOKR	$\mathbf{0.441 \pm 0.009}$	**31**.2%|**63**.**5**%|**72**.**7**%

Table 4.11: Test mean losses and standard errors of Reduced-rank IOKR, and its extended version to leverage $m = 10^5$ unsupervised output data, on the metabolite identification problem.

4.6.4 Conclusion

These experiments empirically show that the proposed extension can take advantage of additional output data.

5

Structured Prediction with Loss Regularization

Contents

5.1 Introduction

Structured Output Prediction (SOP) consists in learning a function whose outputs are structured objects. The key difficulty of SOP usually comes from the discrete nature of the output space itself that does not enjoy the wishable properties of Euclidean spaces. To overcome this issue, various approaches have been proposed in the literature to relax both the inference and learning problems. The wide and emblematic family of energy-based methods rely on an energy function that measures how much an output structured object is fitted to a given input. Inference is processed by solving an "arg max" problem over the output candidate set. Learning then boils down to determine this energy function with the additional price of inference. Probabilistic graphical models with or without deep neural networks are certainly the most well known instances of these methods together with Structured Support Vector Machines. Recently, efforts to improve upon these approaches and overcome the inference cost at training time have led to two distinct lines of research. Some recent works have focused on relaxing the "arg max" problem into a differentiable problem opening the door to end-to-end learning while others have explored the so-called surrogate approaches that embed structured outputs into a Hilbert space and solve consequently a vector-valued regression problem instead of the original structured output prediction task. Notice that this allows to considerably alleviate the computational burden of training but still does not avoid to pay the price of the "arg max" at testing time.

Surrogate methods benefit from strong theoretical guarantees (Bartlett et al., 2006; Mroueh et al., 2012; Ciliberto et al., 2020). In particular, Ciliberto et al. (2020) propose an estimator that is universally consistent, and provide learning rates under standard assumptions. Namely, the excess-risk bounds associated to a loss $\Delta(y, y')$ are of the form $c\, n^{-r}$ where c and r are positive constants independent of the number n of training data. The chosen loss Δ carries a geometry on the output space. In particular, it affects the regularity of the target optimal predictor, and thus the learning rate (through the so-called source condition (Caponnetto and De Vito, 2007)). Moreover, it also impacts the constant c. We note it as $c_{\mathcal{Y}}^{\Delta}$ to make explicit this dependence. This constant plays an important role: it can be very big (potentially of the same order than $|\mathcal{Y}|^{1/2}$), making then the bounds very large (Osokin et al., 2017; Nowak et al., 2019).

Objectives of this work. In this chapter, we aim at proposing a general method, with theoretical guarantees, allowing to exploit the structure of the output space, thanks to an *unsupervised data set* $\mathcal{U}_m = (y_j)_{j=1}^m$, in order to obtain a computational and statistical gain in structured prediction.

Contributions. First, we show that in structured prediction the structure of the output space can be formulated as regularity conditions on the loss function. It allows us to propose a principle of loss regularization exploiting such regularity in order to obtain computationally and statistically efficient structured prediction methods. Then, we study under which setting the approach leads to a computational and statistical gain. Finally, we assess the method experimentally on synthetic and real-world problems.

Related works. The following works are related to this one as they also aim at exploiting non-linear structure of the output space in structured prediction. Luise et al. (2019) propose to leverage the structure of the output space by using trace norm regularized regression estimators. (Ciliberto et al., 2017) propose a method for multi-task learning, and prove improved constants in the learning bounds when leveraging the relations between the tasks rather than treating them independently. (Ciliberto et al., 2019) propose a method for exploiting local structure (input and output data made by parts), showing also improved constants in the learning bounds when exploiting this structure. The first main difference with this work is the formulation of the structure of the output space. We will show that the formulation proposed in this chapter is general, in the sense that most space structures can be expressed within our formulation. The second important difference is that, in this work, we make use of an unsupervised data set of outputs $(y_j)_{j=1}^m$. Finally, another specificity of this chapter is to address the problem of alleviating the computational complexity of the pre-image.

Structure of the chapter. In Section 5.2, we present a background on the ILE estimator. In Section 5.3, we present the loss regularization principle. In Section 5.4, we carry out its theoretical analysis. In Section 5.5, we test the method numerically.

5.2 Background

In this section, we recall standard results on Least-squares regression. Then, we introduce the framework of Implicit Loss Embeddings (Ciliberto et al., 2016, 2020): a general method for structured prediction benefiting from strong theoretical guaran-

tees. It will provide us a sound framework for instantiating our principle in the next section.

5.2.1 Non-parametric estimator for least-squares regression

Least-Squares (LS) regression. Given a Hilbert space $\mathcal{Z}$, a probability distribution ρ over $\mathcal{X} \times \mathcal{Z}$, and a measurable function $h : \mathcal{X} \to \mathcal{Z}$, the expected square loss of h is defined as:

$$\mathcal{R}_{LS}(h) = \mathbb{E}_\rho[\|h(x) - z\|_{\mathcal{Z}}^2]^{1/2}. \tag{5.1}$$

and its minimizer over the space of measurable functions from $\mathcal{X}$ to $\mathcal{Z}$ (Bayes predictor) is:

$$h_z^* : \; x \mapsto \mathbb{E}_{z|x}[z]. \tag{5.2}$$

Solving a regression problem consists in estimating $h_z^* : \mathcal{X} \to \mathcal{Z}$ from a given training sample $(x_i, z_i)_{i=1}^n$, independently drawn from ρ.

Non-parametric regression estimators. Well-known non-parametric estimators for LS regression (trees, random forests, L2-boosting, k-nearest neighbors) with scalar outputs can be extended to vectorial outputs as shown in several works, enjoying the following general form:

$$\hat{h}_z(x) = \sum_{i=1}^{n} \alpha_i(x) z_i, \tag{5.3}$$

where the *weight function* $\alpha : \mathcal{X} \to \mathbb{R}^n$ provides the coefficients of a linear combination of output training data.

These estimators come along with theoretical guarantees of the form $\mathcal{R}_{LS}(\hat{h}_z) - \mathcal{R}_{LS}(h_z^*) \leq \mathcal{O}(n^{-r})$ with $r > 0$ depending on regularity assumptions on the learning problems (Micchelli and Pontil, 2005; Caponnetto and De Vito, 2007; Ciliberto et al., 2020; Cabannes et al., 2021b). For example, the *ridge regression* approach builds an estimator of h_z^* by minimizing the empirical counterpart of the true risk $\mathcal{R}_{LS}(h)$ plus a ℓ_2 regularization term weighted by some positive parameter $\lambda > 0$. In this work, we consider Kernel Ridge Regression (KRR) where the functional space to search the minimizer $\hat{h}_z$ is a vector-valued Reproducing Kernel Hilbert Space (vv-RKHS). (Micchelli and Pontil, 2005) showed that similarly to the scalar case, i.e. $\mathcal{Z} := \mathbb{R}$, the function $\alpha : \mathcal{X} \to \mathbb{R}^n$ enjoys a close-form.

Remark 5.1 (About the weight function α)**.** *Note that KRR in vv-RKHS enjoys an appealing property when the operator-valued kernel is chosen as the decomposable identity kernel: $K(x, x') = k(x, x')I_{\mathcal{Z}}$ with $I_{\mathcal{Z}}$ the identity operator on $\mathcal{Z}$, and k is a positive definite (scalar-valued) kernel over $\mathcal{X}$: the weight function α is the same than in the scalar regression case. As for k-nearest neighbors estimators, their weight functions are the same whether it be in the case of scalar outputs or vectorial outputs.*

5.2.2 Structured prediction

Structured space. Let $\mathcal{Y}$ be a set of structured objects. In this chapter, we call a *structured space* the couple $(\mathcal{Y}, \Delta)$ where $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a dissimilarity on $\mathcal{Y}$ that in practice, takes into account the structure of the objects in $\mathcal{Y}$. Having no prior information about the structure of the objects in $\mathcal{Y}$ would correspond to use $\Delta(y, y') = \mathbb{1}_{y \neq y'}$.

Conversely, for example, a dissimilarity that compares two objects y and y' using a dictionary of substructures brings much more information. Furthermore, note that in the following Δ is not necessarily symmetric.

Structured output prediction. Given a structured space $(\mathcal{Y}, \Delta)$, a probability distribution ρ over $\mathcal{X} \times \mathcal{Y}$ and a function $f : \mathcal{X} \to \mathcal{Y}$, we define the expected structured risk over $\mathcal{Y} \times \mathcal{Y}$ as follows:

$$\mathcal{R}_\Delta(f) = \mathbb{E}_p[\Delta(f(x), y)]. \tag{5.4}$$

It can be shown that the function minimizing the expected risk (5.4) is defined as:

$$f^*(x) = \arg\min_{\hat{y} \in \mathcal{Y}} \mathbb{E}_{y|x}[\Delta(\hat{y}, y)]. \tag{5.5}$$

Structured output prediction refers to the problem of estimating f^* using a training dataset $\mathcal{S}_n = (x_i, y_i)_{i=1}^n$ of n samples independently drawn from the probability distribution ρ over $\mathcal{X} \times \mathcal{Y}$.

In this work, we consider the general family of surrogates approaches that allows to avoid direct minimization of the empirical counterpart of $\mathcal{R}_\Delta(f)$ by solving instead a surrogate LS regression problem. In particular, we anchor our contribution within the Implicit Loss Embedding framework introduced by Ciliberto et al. (2020).

Implicit output embeddings and surrogate estimation. Surrogate approaches for Structured Prediction leverage the notion of output representation: the structured objects of $\mathcal{Y}$ are implicitely embedded into a Hilbert space, in that way, the structured prediction problem is turned into a (surrogate) regression problem. While several works have emphasized the role of a so-called "output kernel" and vector-valued RKHS to implement in practice this approach, an important line of research (Ciliberto et al., 2016, 2020) has focused on the definition of a general framework for surrogate approaches benefiting from strong theoretical guarantees. The cornerstone of this framework called Implicit Loss Embedding is an assumption on the loss function Δ.

Assumption 5.2 (Implicit Loss Embedding (ILE) condition)**.** *There exists a separable Hilbert space $\mathcal{H}_y$ and two measurable bounded maps $\chi, \psi : \mathcal{Y} \to \mathcal{H}_y$, such that for any $y, y' \in \mathcal{Y}$ we have:*

$$\Delta(y, y') = \langle \chi(y), \psi(y') \rangle_{\mathcal{H}_y}. \tag{5.6}$$

Ciliberto et al. (2020) showed that Assumption 5.2 is very mild, and is verified in practice by most couples $(\mathcal{Y}, \Delta)$. In particular, it is verified for any finite output space $\mathcal{Y}$.

Surrogate regression. Instead of solving directly the structured prediction problem, one can solve a surrogate vector-valued regression problem with target $h^*_\psi : x \mapsto \mathbb{E}_{y|x}[\psi(y)]$. Using the training sample $\mathcal{S}_n$, a non-parametric estimator of h^*_ψ can be defined:

$$\hat{h}_\psi(x) = \sum_{i=1}^n \alpha_i(x) \psi(y_i), \tag{5.7}$$

where α is the associated weight function.

ILE estimator. Given Assumption 5.2, defining the estimator of f^* as:

$$\hat{f}(x) := \operatorname*{arg\,min}_{y \in \mathcal{Y}} \langle \chi(y), \hat{h}_\psi(x) \rangle_{\mathcal{H}_y}, \tag{5.8}$$

yields to the ILE estimator proposed by Ciliberto et al. (2020):

$$\hat{f}(x) = \operatorname*{arg\,min}_{y \in \mathcal{Y}} \sum_{i=1}^{n} \alpha_i(x) \Delta(y, y_i). \tag{5.9}$$

Remark 5.3 (Links with KDE and IOKR)**.** *When Δ is defined from a positive definite kernel normalized k_y over $\mathcal{Y}$ such as $\Delta(y,y') = \|k_y(y,\cdot) - k_y(y',\cdot)\|^2_{\mathcal{H}_y} = -2k_y(y,y')$, one retrieve the framework of Kernel Dependency Estimation (Cortes et al., 2005) and ridge Input Output Kernel Regression (ridge-IOKR) (Brouard et al., 2016b).*

Curse of the pre-image An important limitation of the ILE estimator is the computational complexity of computing $\hat{f}(x)$ because of the optimization problem over $\mathcal{Y}$ which need to be solved. The first contribution of this work, is to provide a method to alleviate the computational complexity of the pre-image with theoretical guarantees.

5.2.3 Theoretical guarantees

In the following, we recall excess-risk results in the case of kernel-based methods. Let k_x be a positive definite kernel over $\mathcal{X}$ and $\mathcal{H}_x$ be the RKHS associated to k_x. It can be shown that the vector-valued RKHS associated to the operator-valued kernel $K(x,x') = k(x,x')I_{\mathcal{H}_y}$ is isometric to $\mathcal{H}_y \otimes \mathcal{H}_x$. In the following, we note $h \in \mathcal{H}_y \otimes \mathcal{H}_x$ when $h : \mathcal{X} \to \mathcal{H}_y$ belongs to the RKHS associated to K.

In order to obtain finite sample bounds, the following assumption is required.

Assumption 5.4 (attainability assumption)**.** *We have*

$$x \mapsto \mathbb{E}_{y|x}[\psi(y)] \in \mathcal{H}_y \otimes \mathcal{H}_x. \tag{5.10}$$

This assumption says that the target is in the hypothesis space considered, which is a standard condition in statistical learning (Caponnetto and De Vito, 2007).

Learning bounds. Under the Assumptions 5.2 and 5.4, the ILE estimator (5.9) benefits from the following excess risk bounds:

$$\mathcal{R}_\Delta(f_n) - \mathcal{R}_\Delta(f^*) \leq c^\Delta_{\mathcal{Y}} n^{-1/4} \log(4/\delta) \tag{5.11}$$

where $c^\Delta_{\mathcal{Y}}$ is a constant that depends on the structured space at hand $(\mathcal{Y}, \Delta)$.

Choice of Δ. The constant $c^\Delta_{\mathcal{Y}}$ can be understood as a measure of the size or dimension of the output space $\mathcal{Y}$. For a given output set $\mathcal{Y}$, $c^\Delta_{\mathcal{Y}}$ depends on the choice of the loss Δ. Being able to choose a loss Δ that leads to a small $c^\Delta_{\mathcal{Y}}$ is directly linked to the available a priori information on the geometry of $\mathcal{Y}$. The study of this constant for the most common couples $(\mathcal{Y}, \Delta)$ has been done in (Osokin et al., 2017; Nowak et al., 2019). As an example, if $\mathcal{Y} = \{0,1\}^d$ and $\Delta(y,y') = \mathbb{1}_{y \neq y'}$, corresponding to do not have

any a priori on the structure of $\mathcal{Y}$ (all the $y \in \mathcal{Y}$ lie at the same "distance"), leads to the very large constant $|\mathcal{Y}|^{1/2} = 2^{d/2}$. That is in order to obtain an excess-risk smaller than a desired quantity, the quantity of data required depends exponentialy on the output dimension d. Choosing instead the Hamming loss $\Delta(y, y') = \sum_{i=1}^{d} \mathbb{1}_{y_i \neq y'_i}$, leads to the constant d (linear dependence in the output dimension).

5.3 Structured prediction with loss regularization

In this section, we introduce the structured prediction with additional output training data and leverage this information to define a novel estimator based on output regularization, or equivalently loss regularization. We leave to Section 5.4 the formal assumptions required to back up theoretically this estimator.

Structured prediction with additional output training data. All along this chapter, we assume that in addition to the training sample $\mathcal{S}_n$, we also have access to an additional sample $\mathcal{U}_m = (y_j^u)_{j=1}^m$ of m output data independently drawn from the marginal distribution ρ_y. Our goal is then to build an estimator of f^* defined in Eq. (5.5) using both $\mathcal{S}_n$ and $\mathcal{U}_m$ within the ILE context augmented with novel assumptions.

Regularity of the loss on the output distribution ρ_y. The core idea of this chapter is to express the output space's structure as a regularity assumption on the loss Δ and replace it by its smoothed version anchored on m elements of $\mathcal{U}_m$. It allows us to derive a theoretically sound principle to exploit the output structure in structured prediction and by this way reduce the size of the constant $c_{\mathcal{Y}}^{\Delta}$ in corresponding bounds.

More precisely, we assume that the loss Δ can be well approximated by its regularized version anchored on m elements of $\mathcal{U}_m$:

$$\Delta_m(\hat{y}, y) = \sum_{j=1}^{m} \beta_j(y) \Delta(\hat{y}, y_j^u), \tag{5.12}$$

where $\beta : \mathcal{Y} \to \mathbb{R}^p$ is a weight function associated to the problem of regressing $y \mapsto \Delta(\hat{y}, y)$.
If the marginal distribution ρ_y lies on a submanifold of the structured space $(\mathcal{Y}, \Delta)$, then this informal hypothesis will be satisfied. In Section 5.4, we properly define the set of assumptions needed to i) convert this informal hypothesis and ii) the additional conditions for which the novel estimator we propose benefit from interesting theoretical guarantees. We adopt the ILE setting that we augment with additional hypotheses.

As for standard ILE framework, the structured prediction estimator relies on the definition of the surrogate regression estimator.

Surrogate regression estimator with output regularization. Given $\mathcal{S}_n$ and $\mathcal{U}_m$,

$$h_{n,m}(x) = \sum_{i=1}^{n} \alpha_i(x) \underbrace{\sum_{j=1}^{m} \beta_j(y_i) \psi(y_j^u)}_{\hat{\psi}(y_i)}, \tag{5.13}$$

Novel estimator with loss regularization. Given $\mathcal{S}_n$ and $\mathcal{U}_m$, we propose the following estimator for the Structured Bayes Predictor in (5.5):

$$f_{n,m}(x) = \operatorname*{arg\,min}_{\hat{y}\in\mathcal{Y}} \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i(x)\beta_j(y_i)\Delta(\hat{y}, y_j^u), \tag{5.14}$$

where α and β are weight functions of two LS regression estimators (see Equation (5.3)): $\alpha : \mathcal{X} \to \mathbb{R}^n$ is the weight function of a non parametric estimator defined with the training data $(x_i, \psi(y_i))_{i=1}^n$, while β is the weight function of a non parametric estimator defined with the training data $(y_i^u, \psi(y_i^u))_{i=1}^m$.

Computation of α and β. Depending on the problem at hand, α and β can be defined with the weights of various non parametric LS estimators: kernel ridge regression (Micchelli and Pontil, 2005; Cortes et al., 2005; Caponnetto and De Vito, 2007; Brouard et al., 2011), k-NN regression (Cabannes et al., 2021b), regression trees (Geurts et al., 2006) or boosting (Geurts et al., 2007; Ciliberto et al., 2020). This choice is not neutral since k-nearest neighbours and kernel methods require the prescription of a metric or a kernel, while tree-based methods empirically learn an input kernel.

Remark 5.5 (Relationship with double representation theorem in vector-valued kernel methods)**.** *Learning in vv-RKHS has been studied in (Laforgue et al., 2020) at the lens of duality principle for general convex loss functions. A "double representation theorem" (see Theorem 4 in (Laforgue et al., 2020) was proved. In particular, it can be applied to surrogate regression problems with infinite dimensional output spaces. It expresses that under mild conditions on the Fenchel-Legendre Transform of the loss ℓ and on operator-valued kernel K, the minimizer of the corresponding empirical regularized ℓ-risk writes as: $g_n^\ell(x) = \sum_{i,j=1}^n K(x, x_i)\hat{\omega}_{ij}\psi(y_i)$ where the matrix $\hat{\Omega} = (\omega_{ij})_{i,i=1}^n$ is solution to an associated optimization problem.*

Illustrative minimal example. In the following, we provide an illustration of the whole approach on a simple toy problem. Let us consider the problem of estimating a step function $h^* : \mathbb{R} \to \mathbb{R}$ taking only two values, thanks to a data set $(x_i, y_i)_{i=1}^n$ with $x_i = 2i/(n-1)$, and $y_i = h^*(x_i)+\epsilon_i$ with noise $\epsilon \sim \mathcal{N}(0, 0.2)$. An unsupervised training data set $(y_j^u)_{j=1}^m$ is available. Here $\mathcal{Y} = \mathbb{R}$, and we choose $\Delta(y, y') = (y-y')^2$. On this problem, the outputs verify indeed a strong structure: if x follows a uniform distribution on $[0, 2]$, $\rho_y(y)$ is a mixture of two normal distributions with means 0 and 1. Such regularity translates as the possibility to estimate $y \to \Delta(\hat{y}, y)$, and $x \to \mathbb{E}_{y|x}[\Delta(\hat{y}, y)]$ with few anchors (See Equations (5.12) and (5.14)).

For α, we use KRR weights with a Gaussian kernel, and we select the ridge regularization parameter λ, as well as the kernel parameter, using a validation set of size 100. For β, we use k-NN with a number of neighbors k equal to half of the size of the unsupervised training data set, i.e. $k = 50$.

First, we consider a training set with $n = 10$ and $m = 100$. In this setting, the quantity of training data, compared to the "difficulty" of the learning problem, is such that the proposed regularization leads to a significant statistical gain. We draw a test set of size 100, and observe indeed that, in terms of test mean squared error, the KRR estimator obtains an error equal to 0.044 for without loss regularization, and 0.018 when using the loss regularization. Notice that because the output space is $\mathcal{Y} = \mathbb{R}$, and $\Delta(y, y') = (y - y')^2$, here we can write $\hat{f}(x) = \operatorname{arg\,min}_{\hat{y}\in\mathcal{Y}} \sum_{i=1}^n \alpha_i(x)\Delta(\hat{y}, \bar{y}_i)$ defining

$\bar{y}_i = \sum_{j=1}^{m} \beta_j(y_i) y_j^u$. Therefore, in the case of the square loss, the loss regularization can be easily interpreted as substituting the output training points $(y_i)_{i=1}^n$ with the local averaging $(\bar{y}_i)_{i=1}^n$. This allows to reduce the effect of the noise ϵ. We plot the two estimated maps in Figure 5.1. We also plot the training points $(y_i)_{i=1}^n$ (in blue), the surrogate training points $(\bar{y}_i)_{i=1}^n$ (in orange), the unsupervised training points $(y_j)_{j=1}^m$ (black crosses on the right border of the plot).

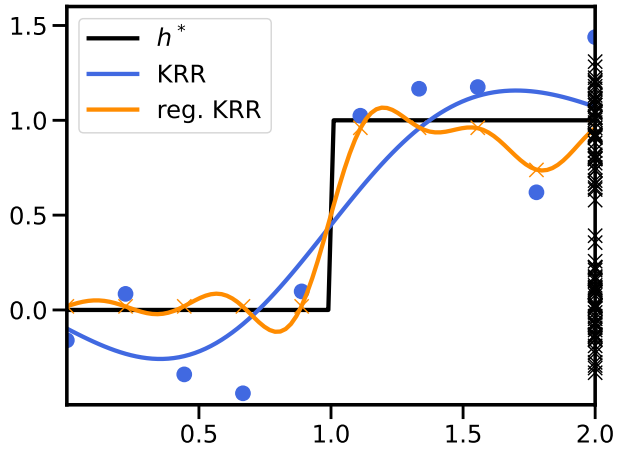

Figure 5.1: Minimal example.

Then, we consider a training set with $n = 1000$ and $m = 100$. In this setting, the quantity of training data, compared to the difficulty of the learning problem, is such that the proposed regularization do not leads to a significant statistical gain. Indeed, in terms of test mean squared error, the KRR estimator with and without loss regularization obtain the same error equal to 0.004. Nevertheless, the estimator with loss regularization only requires $m = 100 \ll n = 1000$ output anchors. This allows to obtain a computational gain when computing the pre-image over a finite candidate set of size $s \in \mathbb{N}^*$: $m(n+s)$ instead of ns which is significant when $m \ll s$ and $m \ll n$.

5.4 Theoretical analysis

In this section, we present a statistical analysis of the proposed estimator. We start, in Section 5.4.1 by giving the assumptions on the learning problem that we consider. Then, in Section 5.4.2, we present the main theoretical results of this work, which are learning bounds, and, finally, we study under which setting the output regularization is computationally and statistically beneficial.

5.4.1 Assumptions

Here, we present and comment the assumptions that we make in order to prove the learning bounds.

Assumption 5.6 (A priori on the regularity of $(\mathcal{Y}, \Delta)$)**.** *We assume that Δ admits an ILE*

$$\Delta(y, y') = \langle \chi(y), \psi(y') \rangle_{\mathcal{H}_y} \tag{5.15}$$

and that we know a kernel $k_y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ *with RKHS* $\tilde{\mathcal{H}}_y$*, such that*

$$\psi \in \mathcal{H}_y \otimes \tilde{\mathcal{H}}_y. \tag{5.16}$$

More precisely, here, "knowing k_y" stands for being able to compute $k_y(y, y')$ for any $y, y' \in \mathcal{Y}$. Assumption 5.6 should be understood as corresponding to have a priori information about the regularity of the maps $y \mapsto \Delta(\hat{y}, y)$ for all $\hat{y} \in \mathcal{Y}$. Indeed, it requires to have an ILE for Δ, and also information about the right embedding. Notice that there is no unicity of the embeddings. Assumption 5.6 can be equivalently formulated as the existence of a Hilbert space $\mathcal{H}_y$, an embedding $\chi : \mathcal{Y} \to \mathcal{H}_y$, a known kernel k_y with canonical map $\tilde{\psi}(y) = k_y(y, .)$, and a Hilbert-Schmidt operator $W : \tilde{\mathcal{H}}_y \to \mathcal{H}_y$ such that

$$\Delta(y, y') = \langle \chi(y), W\tilde{\psi}(y') \rangle_{\mathcal{H}_y}. \tag{5.17}$$

As we will see just below through the examples, Assumption 5.6 is mild, in the sense that having such a priori is possible in most of the practical case. It corresponds to having information about the regularity of the loss Δ.

Examples. Here are some examples, for common structured output spaces $(\mathcal{Y}, \Delta)$, of kernels k_y that verifies Assumption 5.6.

1. **Finite output spaces.** If $|\mathcal{Y}| < +\infty$, Ciliberto et al. (2016) show how to construct explicit loss embeddings χ, ψ with finite output dimension. Therefore, $k_y(y, y') = \langle \psi(y), I\psi(y') \rangle$ verifies Assumption 5.6 with $\tilde{\psi} = \psi$ and $W = I$. This includes, for instance, the following settings.
   - **Multi-label classification.** $\Delta(y, y') = \sum_{j=1}^{d} \mathbb{1}_{y_j \neq y'_j}$ with $\mathcal{Y} = \{0, 1\}^d$.
   - **Loss induced by a kernel.** $\Delta(y, y') = \|\psi(y) - \psi(y')\|^2_{\mathcal{H}_y}$ with $\psi : \mathcal{Y} \to \mathcal{H}_y$ an embedding map taking values in a Hilbert space $\mathcal{H}_y$, and $|\mathcal{Y}| < \infty$.
2. **Smooth structured spaces.** Smoothness conditions on the maps $\forall\, y' \in \mathcal{Y}, y \mapsto \Delta(y, y')$ (e.g. are s-times differentiable, or are analytical functions), can be typically translated as Assumption 5.6 with radial kernels k_y (e.g. Laplace and Gaussian kernel). This is based on the fact that radial kernels can generate Sobolev spaces. Let's give two examples of such result.
   - **Manifold prediction.** If Δ is the squared geodesic distance on the hypersphere S^{d-1}, $d \in \mathbb{N}^*$, then $\tilde{\mathcal{H}}_y$ can be chosen as a Sobolev space on S^{d-1} (See Rudi et al. (2018)).
   - **Probability distribution prediction.** Luise et al. (2018) show that $S \in \tilde{\mathcal{H}}_y \otimes \tilde{\mathcal{H}}_y$ where S is the Sinkhorn distance between probability distribution over a finite space, and $\tilde{\mathcal{H}}_y$ is a Sobolev space which is a RKHS.

The following assumption measures how much the outputs $h^*_{\tilde{\psi}}(x) = \mathbb{E}_{y|x}[\tilde{\psi}(y)]$ are concentrated in comparison to the noise $\epsilon_{\tilde{\psi}} = \tilde{\psi}(y) - \mathbb{E}_{y|x}[\tilde{\psi}(y)]$. Let define $\tilde{\psi}(y) = k_y(y, .)$, and the operators $M_{\tilde{\psi}} = \mathbb{E}[h^*_{\tilde{\psi}}(x) \otimes h^*_{\tilde{\psi}}(x)]$, $E_{\tilde{\psi}} = \mathbb{E}[\epsilon_{\tilde{\psi}} \otimes \epsilon_{\tilde{\psi}}]$.

Assumption 5.7 (concentrated targets)**.** *The operators $M_{\tilde{\psi}}$ and $E_{\tilde{\psi}}$ satisfy the following property. There exists $\gamma > 0$, $c_1 > 0$ such that:*

$$M_{\tilde{\psi}} \preceq c_1 E_{\tilde{\psi}}^{\gamma}, \tag{5.18}$$

This assumption is always verified for $\gamma = 0$ as $\|M_{\tilde{\psi}}\|_\infty < +\infty$, and the greater is γ the more the noise is diffuse in comparison to the targets $h^*_{\tilde{\psi}}(x)$: $M_{\tilde{\psi}}$ has a faster eigenvalue decay rate than $E_{\tilde{\psi}}$. As a limiting case, when $\gamma \to +\infty$ then because $\|E_{\tilde{\psi}}\|_{\mathrm{HS}} < +\infty$, it implies that $M_{\tilde{\psi}}$ is finite-rank.

Mildness of Assumption 5.7. How likely is this regularity assumption to be verified in practice? While Assumption 5.6 makes the proposed model eligible to estimate f^*, Assumption 5.7 makes it possible with $m \ll n$. This can be understood as a measure of the quantity of output anchors $(y_j^u)_{j=1}^m$ required to make the model able to estimate f^*. Taking the minimal example in Section 5.3, this assumption is strongly verified. At the opposite, the generic kernel k_y, proposed in Ciliberto et al. (2020) for any finite output space $\mathcal{Y}$, indeed verifies Assumption 5.6, but corresponds in fact to do not have more structure information about $(\mathcal{Y}, \Delta)$ than only the values $(\Delta(y, y'))_{y,y' \in \mathcal{Y}}$. In this case, Assumption 5.7 is only verified for $\gamma = 0$: one can not interpolate the values of Δ from a subset of the values $(\Delta(y, y'))_{y,y' \in \mathcal{Y}}$ without more information on Δ.

Assumption 5.8 (diffuse noise)**.** *The operator $E_{\tilde{\psi}}$ satisfies the following property. There exists $\tau \in [0, 1[$, $c_2 > 0$ such that:*

$$\mathrm{Tr}(E_{\tilde{\psi}}(E_{\tilde{\psi}} + \mu I)^{-1}) \geq c_2 \mu^{-\tau} \tag{5.19}$$

for all $\mu \leq \|E\|_\infty$.

This assumption is always verified for $\tau = 0, c_2 = 1/2$, and the greater is τ the more the noise is diffuse, namely $E_{\tilde{\psi}}$ has a slow eigenvalue decay rate. This assumption will quantify how much noise one can removed when using the proposed regularization.

5.4.2 Main result

We focus on the weights α, β defined via kernel ridge regression with kernel k_x, k_y respectively, and regularization parameter λ, μ respectively. That is $f_{n,m}(x)$ is the proposed estimator Eq. (5.14), with $\alpha(x) = (K_x + n\lambda I)^{-1} k_x(x)$ with $K_x = (k_x(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n \times n}$, and $k_x(x) = (k_x(x, x_1), \ldots, k_x(x, x_n)) \in \mathbb{R}^n$, and $\beta(y) = (K_y + m\mu I)^{-1} k_y(y)$ with $K_y = (k_y(y_i, y_j))_{i,j=1}^m \in \mathbb{R}^{m \times m}$, and $k_y(y) = (k_y(y, y_1), \ldots, k_y(y, y_m)) \in \mathbb{R}^m$. The proofs of Theorem 5.9, and Corollaries 1 and 2, are provided in Appendix 6.3.

Theorem 5.9 (Learning bounds)**.** *Under Assumptions 5.4, 5.6, and 5.7, using the λ defined in the proof, if $\mu \geq \frac{9c_{\tilde{\psi}}^2}{m} \log(\frac{m}{\delta})$, then with probability at least $1 - \delta$*

$$\mathcal{R}_\Delta(f_{n,m}) - \mathcal{R}_\Delta(f^*) \lesssim \|W P_\mu E_{\tilde{\psi}}^{1/2}\|_{\mathrm{HS}}^{1/2} n^{-1/4} + \mu^{\gamma/2} \tag{5.20}$$

with $P_u = (C_{\tilde{\psi}} + \mu I)^{-1} C_{\tilde{\psi}}$.

When $\mu = 0$, we recover the bound obtained without loss regularization, which is $\mathcal{R}_\Delta(f_{n,m}) - \mathcal{R}_\Delta(f^*) \lesssim \|W E_{\tilde{\psi}}^{1/2}\|_{\mathrm{HS}}^{1/2} n^{-1/4}$. $P_u = (C_{\tilde{\psi}} + \mu I)^{-1} C_{\tilde{\psi}}$ can be understood as a projection operator over the main components of $C_{\tilde{\psi}}$ whose rank is continuously controlled by μ.

Remark 5.10. *For the sake of clarity, we use the notation $a \lesssim b$ when there exists $c > 0$ independent of the parameters of interest (here n, γ and τ) such that $a \leq c \times b$. The constants c are explicitly defined in the proofs. In particular, the bounds holds with probability at least $1 - \delta$, and the dependency of the constants on δ are in $\log^2(4/\delta)$.*

Now, we derive from Theorem 5.9 two corollaries studying when the proposed method can lead to significant computational and statistical gain.

Corollary 1 (Computational gain)**.** *Under Assumptions 5.4, 5.6, and 5.7, taking $\mu = \frac{9c_{\tilde{\psi}}^2}{m}\log(\frac{m}{\delta})$, as soon as*

$$\frac{m}{\log(m)} \gtrsim n^{\frac{1}{2\gamma}} \tag{5.21}$$

then we have with probability at least $1 - \delta$

$$\mathcal{R}_\Delta(f_{n,m}) - \mathcal{R}_\Delta(f^*) \lesssim n^{-1/4} \tag{5.22}$$

Hence, when Assumption 4 is verified with a small γ, then Corollary 1 shows that f^* can be well estimated with only few anchors. This leads to a significant computational gain when computing the pre-image. Indeed, let $s \in \mathbb{N}^*$ be the size of the candidate set $\mathcal{Y}_c \subset \mathcal{Y}$ over which is computed the pre-image. Without loss regularization, one needs to compute $\alpha(x)\Delta_{tr,c}$ where $\Delta_{tr,c} = (\Delta(y_i, y_c))_{i \in [\![1,n]\!], y_c \in \mathcal{Y}_c} \in \mathbb{R}^{n \times s}$, and $\alpha(x) \in \mathbb{R}^n$. The computational complexity for one prediction is then $\mathcal{O}(ns)$. With loss regularization, one needs to compute $\alpha(x)\Delta_{tr,u}\beta_{tr,c}$ where $\beta_{tr,c} = (\beta_j(y_c))_{j \in [\![1,m]\!], y_c \in \mathcal{Y}_c} \in \mathbb{R}^{m \times s}$, $\Delta_{tr,u} = (\Delta(y_i, y_j^u))_{i \in [\![1,n]\!], j \in [\![1,m]\!]} \in \mathbb{R}^{n \times m}$, and $\alpha(x) \in \mathbb{R}^n$. The computational complexity for one prediction is then $\mathcal{O}(m(n+s))$.

Corollary 2 (Statistical gain)**.** *Under Assumptions 5.4, 5.6, 5.7, and 5.8 taking μ defined in the proof, under the same assumptions than in Theorem 5.9, we have with probability at least $1 - \delta$*

$$\mathcal{R}_\Delta(f_{n,m}) - \mathcal{R}_\Delta(f^*) \lesssim \|E_{\tilde{\psi}}^{1/2}\|_{\mathrm{HS}}^{1/2}(1 - n^{-(1-\tau)/\gamma})^{1/4} n^{-1/4}. \tag{5.23}$$

This corresponds to the bound obtained without loss regularization multiplied by $(1 - n^{-(1-\tau)/\gamma})^{1/4}$. In particular, for any $k \in \mathbb{N}^*$, one can obtain a constant k times smaller when using loss regularization than without using it, as soon as:

$$n \leq \left(1 - \frac{1}{k^4}\right)^{-\frac{\gamma}{1-\tau}}. \tag{5.24}$$

That is, one obtains a constant divided by k, when γ is big enough (concentrated signal), τ close enough to 1 (spreaded out noise), and n not too big to benefit from this regularization.

To put it in a nutshell, when Assumptions 5.6 and 5.7 are verified, one can ensure that the proposed loss regularization induces a negligible bias and a significant computational gain. When, in addition, Assumption 5.8 is verified, one can ensure also a significant statistical gain (decreasing with n), through a noise reduction.

Remark 5.11 (Alternative to Assumption 5.7 for the computational gain)**.** *Corollary 1 could be obtained under an alternative condition than the concentration of the targets $h^*_{\tilde{\psi}}(x)$ in comparison to the noise ϵ (Assumption 5.7), by assuming instead concentration of the embeddings $\tilde{\psi}(y)$, via an assumption of the form $\mathrm{Tr}(C_{\tilde{\psi}}(C_{\tilde{\psi}} + \mu I)^{-1}) \leq c\mu^{-\tau}$ with $\tau < 1$, $c > 0$. Assumptions 5.4 and 5.6 are required to obtain the following Corollary 2. For the sake of simplicity, in this work, we consider Assumptions 5.4 and 5.6 for Corollary 1, in order to have only one set of assumptions for the two corollaries.*

5.5 Numerical experiments

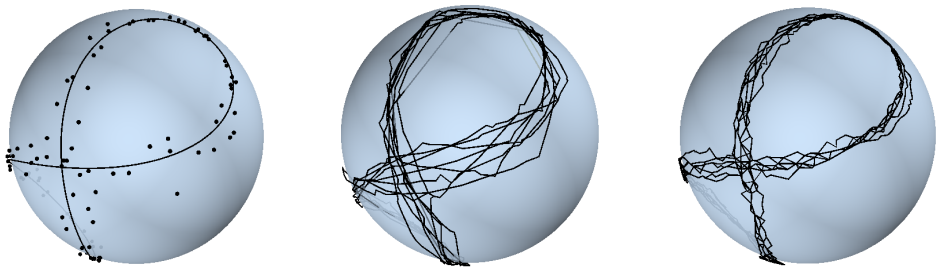

Figure 5.2: Output regularization on the sphere. Left: training data (black points) lie on a noisy submanifold of the sphere (black line). Middle: Predicted map's values with no output regularization. Right: Predicted map's values with output regularization.

In this section, we provide an empirical study to emphasize the relevance of loss regularization on synthetic and real-world structured prediction problems.

5.5.1 Synthetic problem

We first give an illustration of how the proposed method operates when dealing with a non-Euclidean output space.

Problem and data set. Here is the experimental setup: the output training data lie on a known manifold $\mathcal{M}$, and the chosen loss Δ is the geodesic distance. Furthermore, the target function $f^* : \mathcal{X} \to \mathcal{M}$ takes values on a submanifold $\mathcal{S} \subset \mathcal{M}$. Our aim is to measure how much loss regularization with the help of the additional ouput training data allows to exploit this property and improves upon the loss without regularization scheme.

As an instance of such a setting, we assume that $\mathcal{M}$ is a 3D-sphere, x is a random continuous variable uniformly distributed on $[0,1]$ and the $y = (\phi, \theta) \in \mathcal{M}$ are represented in spherical coordinates. $\rho_{y|x}$ is defined by the equation $y = f^*(x) + \epsilon$ with f^* making 8 cycles at uniform speed from 0 to 1 inside the Clelia curve $\phi = c\theta$ with $c = 1$, and $\epsilon \sim \mathcal{N}(0, 0.1 I_{\mathbb{R}^2})$ is a Gaussian noise. More precisely, $f^*(x) = (8x, 8x)$. The probability distribution ρ_y is marginalized out using ρ_x and $\rho_{y|x}$. We consider $n = 100$ training supervised data, with $m = 1000$ additionnal output data.

Experimental setting. We use KRR weights for both α and β with regularization parameters λ_x, λ_y, defined from two Gaussian kernels $k_{\mathcal{X}}$, wit $k_{\mathcal{Y}}$ and bandwidth $\sigma_{\mathcal{X}}, \sigma_{\mathcal{Y}}$, respectively. All hyperparameters $\lambda_x, \lambda_y, \sigma_x, \sigma_y$ are selected using a validation set of size 1000. The mean test error defined with Δ is computed over a test set of size 1000.

Results. We observe that the proposed regularization in this setting indeed yields to a gain in accuracy: the mean test error obtained with and without output regularization on this problem are 0.133 and 0.200, respectively. We plot the training data $(y_i)_{i=1}^n$, the true map's outputs $f^*(x)$, and the estimated maps' outputs $\hat{f}(x)$ (with and without regularization) from left to right in Figure 5.2. We note that the output regularization enforces the model to respect the structure of the output space observed in the unsupervised training set.

Remark 5.12. *Notice that, as for the minimal example presented above of Section 5.6, the problem is very low dimensional for vizualisation purpose. Hence, the quantity of training data considered are very small to make the benefits of output regularization significant. Indeed, the statistical benefits of regularization decreases when n increases, this intuitive result, observed in the minimal example of Section 5.6, corroborates with the theoretical analysis (See Equation* (5.24)*).*

5.5.2 Image reconstruction

The goal of this experiment is to assess the benefits of the proposed regularization on both a higher dimensional space and a real-world problem. Image reconstruction is emblematic of the literature in structured prediction (see for instance Cortes et al. (2005); Geurts et al. (2006)) and will serve here as ....

Problem and data set. The aim of this image reconstruction problem provided by Weston et al. (2003) is to predict the bottom half of a USPS handwritten postal digit (16 x 16 pixels), igen its top half. The data set contains 7291 training labeled images and 2007 test images.

Experimental setting. We consider a number of training data $n = 1000$, and build a validation set with 3000 training data. As in Weston et al. (2003) we used as target loss and RBF loss $\Delta(y, y') = \|\psi(y) - \psi(y')\|^2_{\mathcal{H}_y}$ induced by a Gaussian kernel k_y, and visually chose the kernel's parameter $\sigma_y^2 = 10$, looking at reconstructed images of the method on the validation set. For α, we use a KRR weight function with a Gaussian kernel of width σ_x^2 and regularization parameter λ_x. For β, we use a KRR weight function with a Gaussian kernel of width σ_y^2 and regularization parameter λ_y. We select all the hyper-parameters using the validation set. For the ILE approach (with and without loss regularization), we compute the pre-image by using the $s = 7291$ training outputs.

Comparison with a SOTA method. We start by comparing the ILE approach with the SPEN method Belanger and McCallum (2016) employing the standard architecture and training method described in Belanger and McCallum (2016). Results are given in Table 5.1. We can see that on this problem ILE outperforms SPEN. On this data set the loss regularization does not lead to a gain in performance, but does lead to a computational gain as we will see just below.

Alleviating the decoding computations. The ILE approach decoding complexity for one test point is $\mathcal{O}(n(n+s)$. When using the loss regularization with m anchors, the decoding complexity reduces to $\mathcal{O}(m(n+s))$. We illustrate this experimentally by building unsupervised training sets $(y_j^u)_{j=1}^m$ for various m in $[\![1, 300]\!]$. In Figure 5.3, we plot the decoding time (in orange), and the Mean test loss (in blue), w.r.t. m, along

Method	RBF loss
SPEN	0.801 ± 0.011
ILE	**0.752** ± **0.011**

Table 5.1: Test mean losses and standard errors for the ILE and SPEN methods on the USPS digits reconstruction problem where $n = 1000$, and $n_{test} = 2007$.

with the Mean test loss obtained without regularization. We can see that when m increases the Mean test loss quickly reach the same level of accuracy than the ILE estimator with $n = 1000$ anchors. At the opposite, the decoding time increases linearly with m. Hence, using a reduced number of anchors helps to reduce significantly the decoding time while incurring a negligible loss of performance.

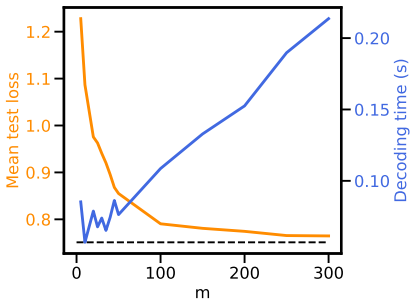

Figure 5.3: Computational gain on USPS.

5.6 Conclusion

Output regularity assumptions in structured prediction can be formulated as regularity conditions on the loss function. In particular, we show that knowledge about the marginal output distribution can be used to smooth the loss function allowing to obtain statistical and computational gain. Backed up by theoretical guarantees, the approach relies on the simplicity of non-parametric estimators and can be applied on a large variety of problems where additional output data are available. Moreover, it completes the set of tools available to improve upon surrogate approaches alongside with low rank approaches whose purpose is different and focused on the control over output dimension. However nothing prevents to combine those approaches or even leverage the same idea in other approaches for structured prediction (energy-based or end-to-end).

6

Proofs and Additional Results

Contents

6.1 Proofs and additional results of Chapter 3

6.1.1 Theory

6.1.1.1 Proof of FGW continuity

We prove the continuity of $\mathrm{FGW}(.,y):\mathcal{Y}_p\to\mathbb{R}$ for any $y'\in\mathcal{Y}_{dis}$. Such result is crucial to prove the ILE property of $\mathrm{FGW}:\mathcal{Y}_p\times\mathcal{Y}_{dis}\to\mathbb{R}$.

Lemma 6.1 (FGW continuity)**.** *Let $y=(C_2,F_2)$ with $C_2\in\mathbb{R}^{p_2\times p_2}, F_2\in\mathbb{R}^{p_2\times d}$, $p_2,d\in\mathbb{N}^*$. The map $FGW(.,y'):\mathcal{Y}_p\to\mathbb{R}$ is continuous.*

Proof Recall that for any $y = (C, F) \in \mathcal{Y}_p$:

$$\mathrm{FGW}_2^2(y, y') = \min_{\pi \in \mathcal{P}_{p_1,p_2}} \sum_{i,k,j,l} \Big[(1-\beta)\|F(i) - F_2(j)\|_{\mathbb{R}^d}^2 + \beta(C(i,k) - C_2(j,l))^2\Big]\pi_{i,j}\pi_{k,l}. \quad (6.1)$$

Using the inequality $|\min_\pi f(\pi) - \min_\pi g(\pi)| \leq \sup_\pi |f(\pi) - g(\pi)|$ for any $f, g : \mathcal{P}_{p_1,p_2} \to \mathbb{R}$, we have for any $dy = (dC, dF) \in \mathcal{Y}_p$

$$|\mathrm{FGW}_2^2(y + dy, y') - \mathrm{FGW}_2^2(y, y')| \leq \sup_{\pi \in \mathcal{P}_{p_1,p_2}} \Big| \sum_{i,k,j,l} \Big[(1-\beta)\Big(\langle dF(i)|F_2(j)\rangle_{\mathbb{R}^d} + o(\|dF(i)\|_{\mathbb{R}^d})\Big) \quad (6.2)$$

$$+ \beta\Big(dC(i,k)C_2(j,l) + o(dC(i,k))\Big)\Big]\pi_{i,j}\pi_{k,l}\Big|$$

$$\leq pp_2\Big[(1-\beta)\Big(\|dF\|_{\mathbb{R}^{p\times d}}\|F_2\|_{\mathbb{R}^{p\times d}} + o(\|dF\|_{\mathbb{R}^{p\times d}})\Big) \quad (6.3)$$

$$+ \beta\Big(\|dC\|_{\mathbb{R}^{p\times p}}\|C_2\|_{\mathbb{R}^{p_2\times p_2}} + o(\|dC\|_{\mathbb{R}^{p\times p}})\Big)\Big]$$

$$= \mathcal{O}(\|dy\|_{\mathbb{R}^{p\times p}\times\mathbb{R}^{p\times d}}) \xrightarrow[dy\to 0]{} 0 \quad (6.4)$$

where from (13) to (14) we have used the Cauchy–Schwarz inequality, and the fact that $\forall (i,j) \in [\![1,p]\!] \times [\![1,p_2]\!], \pi_{ij} \leq 1$.

We conclude that $y \to \mathrm{FGW}_2^2(y, y')$ is a continuous on $\mathbb{R}^{p\times p} \times \mathbb{R}^{p\times d}$, hence on $\mathcal{Y}_p$.

■

6.1.1.2 Universal consistency theorem

We restate the universal consistency theorem from Ciliberto et al. (2020) that is verified by our estimator because of the proved ILE property.

Theorem 6.2 (Universal Consistency)**.** *Let k be a bounded universal reproducing kernel. For any $n \in \mathbb{N}$ and any distribution ρ on $\mathcal{X} \times \mathcal{Y}_{dis}$ let f_W be the proposed estimator built from n independent couples $(x_i, y_i)_{i=1}^n$ drawn from ρ. Then, if $\lambda = n^{-1/2}$,*

$$\lim_{n\to+\infty} \mathcal{R}_\Delta^p(f_W) = \mathcal{R}_\Delta^p(f^*) \quad \textit{with probability} \quad 1. \quad (6.5)$$

6.1.1.3 Attainability assumption

The following assumption is required to obtain finite sample bounds. It is a standard assumption in learning theory (Caponnetto and De Vito, 2007). It corresponds to assume that the solution h^* of the surrogate problem indeed belongs to the considered hypothesis space, namely the reproducing kernel Hilbert space induced by the chosen operator-valued kernel $\mathcal{K}(x, x') = k(x, x')I_{\mathcal{U}}$.

Assumption 6.3 (attainable case)**.** *We assume that there exists a linear operator $H : \mathcal{H}_x \to \mathcal{U}$ with $\|H\|_{\mathrm{HS}} < +\infty$ such that*

$$\mathbb{E}_{Y|x}[\psi(Y)] = Hk(x, .) \quad (6.6)$$

with $\mathcal{H}_x$ the reproducing kernel Hilbert space associated to the kernel $k(x, x')$.

6.1.2 Neural network model and training algorithm

Choice of the templates. As always in deep learning, parameter initialization is an important aspect and we discuss now how to initialize the templates $\bar{y}_j$. In practice they can be initialized at random with matrices $\bar{C}_j$ drawn uniformly in $[0, 1]$ or chosen at random from training samples as suggested by the non-parametric model. One interesting aspect is that the number of nodes do not need to be the same for all templates. This means that one can have both templates with few nodes and templates with a larger number of nodes allowing for a coarse to-fine modeling of the graphs.

Pseudocode. We give the pseudocode for the proposed neural network training algorithm. This algorithm has been implemented in Python using the POT library: Python Optimal Transport (Flamary et al., 2021), and Pytorch library (Paszke et al., 2019).

Algorithm 6.1 Neural network-based model training - One stochastic gradient descent step

Input: $x \to \alpha(x)$ neural network's parameters W. Templates $(\bar{y}_j)_{j=1}^M$. Dictionary learning (True or False).

1. If Dictionary learning is True: $\theta = (W, (\bar{y}_j)_{j=1}^M)$. Otherwise: $\theta = W$.
2. $(\bar{\pi}_j)_{j=1}^M \leftarrow$ Compute the barycenter $f_\theta(x_i)$.
3. $\pi_i \leftarrow$ Compute the losses $\mathrm{FGW}(f_\theta(x_i), y_i))$.
4. $\nabla_\theta \leftarrow$ Compute the gradient of $\mathrm{FGW}(f_\theta(x_i), y_i))$ with fixed OT plans $(\bar{\pi}_j)_j$ and π_i.

Return: Updated neural network's parameters W, updated templates $(\bar{y}_j)_{j=1}^M$.

6.1.3 Justification of the algorithms

Reminder on ILE and surrogate problem:
Recall that $\hat{h}$ is solving a least-squares problem, that is estimate $h^*(x) = \mathbb{E}_{y|x}[\psi(y)]$. Moreover, we can write $f^*(x) = \arg\min_{\hat{y}} \mathbb{E}_{y|x}[\Delta(\hat{y}, y)]$. Now, we can provide intuition in the following derivations about the construction of $\hat{f}$ exploiting the linearity of expectation.

$$\begin{aligned}\hat{f}(x) &= \arg\min_{\hat{y}} \langle \chi(\hat{y}), \hat{h}(x) \rangle_{\mathcal{H}} \\ &\approx \arg\min_{\hat{y}} \langle \chi(\hat{y}), h^*(x) \rangle_{\mathcal{H}}.\end{aligned}$$

Moreover, we have:

$$\begin{aligned}\langle \chi(\hat{y}), h^*(x) \rangle_{\mathcal{H}} &= \mathbb{E}_{y|x}[\langle \chi(\hat{y}), \psi(y) \rangle_{\mathcal{H}}] \\ &= \mathbb{E}_{y|x}[\Delta(\hat{y}, y)]\end{aligned}$$

and thus, taking the "arg min" gives:

$$\hat{f}(x) \approx f^*(x).$$

6.2 Proofs and additional results of Chapter 4

In this section we prove Theorem 4.7 and Corollary 4.9. The proofs are organized as follows:

- Appendix 6.2.1 introduces some necessary notations and definitions.
- Appendix 6.2.2 provides the proof for bounding $\mathbb{E}[\|\hat{P}(\hat{h}(x) - h^*(x))\|_{\mathcal{Z}}^2]$ (Lemma 6.5).
- Appendix 6.2.3 provides the proof for bounding $\mathbb{E}[\|\hat{P}h^*(x) - h^*(x)\|_{\mathcal{Z}}^2]$ (Lemma 6.9).
- Appendix 6.2.4 provides the proof for bounding $\mathbb{E}[\|\hat{P}\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]$ (Theorem 4.7) using Lemmas 6.5 and 6.9.
- Appendix 6.2.5 provides the proof for the Corollary 4.9 using Theorem 4.7.
- Appendix 6.2.6 gives some technical results used in the proofs.
- Appendix 6.2.7 discusses the assumption that $\phi(x)$ and ϵ are independent.

6.2.1 Notations and Definitions

In the following we consider $\mathcal{X}$ to be a Polish space, and $\mathcal{Z}$ a separable Hilbert space. We define here the ideal operators that we will use in the following

- The feature map $\phi : \mathcal{X} \to \mathcal{H}_x$, $\forall x \in \mathcal{X}$, $\phi(x) = k(x, .)$, with $\|\phi(x)\|_{\mathcal{H}_x} \leq \kappa$ with $\kappa > 0$.
- The target $h^*(.) \in \mathcal{H} = \mathbb{E}_{z|.}(z)$, and $Q > 0$ such that $\forall z \in \mathcal{Z}, \|z\|_{\mathcal{Z}} \leq Q$.
- $S : f \in \mathcal{H}_x \to \langle f, \phi(.)\rangle_{\mathcal{H}_x} \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$
- $Z : z \in \mathcal{Z} \to \langle z, h^*(.)\rangle_{\mathcal{Z}} \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$

and their empirical counterparts

- The KRR estimator $\hat{h}(.) \in \mathcal{H}$ trained with n couples $(x_i, z_i)_{i=1}^n$
- $S_n : f \in \mathcal{H}_x \to \frac{1}{\sqrt{n}}(\langle f, \phi(x_i)\rangle_{\mathcal{H}_x})_{1 \leq i \leq n} \in \mathbb{R}^n$
- $Z_n : z \in \mathcal{Z} \to \frac{1}{\sqrt{n}}(\langle z, z_i\rangle_{\mathcal{Z}})_{1 \leq i \leq n} \in \mathbb{R}^n$

From there, we can define the following covariance operators

- $C = \mathbb{E}_x[\phi(x) \otimes \phi(x)] = S^*S$
- $V = \mathbb{E}_z[z \otimes z]$
- $M = \mathbb{E}_x[h^*(x) \otimes h^*(x)]$
- $Z^*S = \mathbb{E}_{x,z}[z \otimes \phi(x)]$

and their empirical counterparts

- $C_n = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i) \otimes \phi(x_i)$
- $V_n = \frac{1}{n}\sum_{i=1}^{n} z_i \otimes z_i$
- $M_n = \frac{1}{n}\sum_{i=1}^{n} \hat{h}(x_i) \otimes \hat{h}(x_i)$
- $Z_n^* S_n = \frac{1}{n}\sum_{i=1}^{n} z_i \otimes \phi(x_i)$

From Lemmas 16 and 17 in Ciliberto et al. (2016) we recall that we have

- $h^*(.) = H\phi(.)$ with $H = Z^* S C^\dagger \in \mathcal{Z} \otimes \mathcal{H}_x$
- $\hat{h}(.) = H_n\phi(.)$ with $H_n = Z_n^* S_n (C_n + \lambda I)^{-1} \in \mathcal{Z} \otimes \mathcal{H}_x$
- $M = HCH^*$
- $M_n = H_n C_n H_n^*$

6.2.2 KRR Error on a Subspace

In this subsection we prove a bound on the kernel ridge regression error on the subspace defined by $\hat{P}$:

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x) - \hat{P}h^*(x)\|_{\mathcal{Z}}^2]^{1/2} = \|\hat{P}(H_n - H)S^*\|_{\mathrm{HS}}. \tag{6.7}$$

Equation (6.7) is obtained by definition of the operators H_n, H, S (see e.g. Ciliberto et al. (2016)).

In order to bound (6.7), one can not directly apply standard learning bounds for kernel ridge estimator on the learning problem $(x, \hat{P}y)$ with $(x, z) \sim \rho$, as $\hat{P}$ depends on the training data. That is why we will decompose (6.7) as

$$\|\hat{P}(H_n - H)S^*\|_{\mathrm{HS}} \leq \|\hat{P}(A + tI)^{1/2}\|_{\mathrm{HS}} \times \|(A + tI)^{-1/2}(H_n - H)S^*\|_\infty \tag{6.8}$$

with a well chosen operator $A : \mathcal{Z} \to \mathcal{Z}$.

As a first step, we give a bound on the KRR estimator excess-risk with respect to the operator norm.

Lemma 6.4 (Bound $\|(H_n - H)S^*\|_\infty$)**.** *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a bounded kernel with $\forall x \in \mathcal{X}, k(x,x) \leq \kappa^2$. Let ρ be a distribution on $\mathcal{X} \times \mathcal{Z}$ such that its marginal w.r.t z is supported on the ball $\|z\|_{\mathcal{Z}} \leq Q$. Let $\hat{h} = H_n\phi(.)$ be the KRR estimator trained with n independent couples drawn from ρ, and regularization parameter $\lambda_2 > \frac{9\kappa^2}{n}\log(\frac{n}{\delta})$. Let $\delta \in [0,1]$. Then, under Assumption 4.1, $H_n S^* - HS^* = A_1 + A_2$, with*

$$A_1 := Z_n^* S_n (C_n + \lambda_2 I)^{-1} S^* - HC_n(C_n + \lambda_2 I)^{-1} S^* \tag{6.9}$$

$$A_2 := HC_n(C_n + \lambda_2 I)^{-1} S^* - HS^* \tag{6.10}$$

and with probability at least $1 - 2\delta$

$$\|A_1\|_\infty \leq \sqrt{\frac{24\eta(Q^2 + \|E\|_\infty \lambda_2^{-1}\kappa^2)}{n}} + \frac{8\kappa Q\eta}{3\sqrt{\lambda_2 n}}; \qquad \|A_2\|_\infty \leq \sqrt{2}\sqrt{\lambda_2}\|H\|_\infty \tag{6.11}$$

with $\eta = \log(\frac{4(\frac{2\mathrm{Tr}(C)}{\lambda_2} + \frac{\mathrm{Tr}(E)}{\|E\|_\infty})}{\delta})$, $E = \mathbb{E}[\epsilon \otimes \epsilon]$, $\epsilon = z - h^*(x)$, $R = \|H\|_{\mathrm{HS}}$.

Proof

Decomposition. The decomposition $H_n S^* - HS^* = A_1 + A_2$ is obtained noticing that we have $H_n = Z_n^* S_n (C_n + \lambda_2 I)^{-1}$ (See section 6.2.1).

1. Bound $\|A_1\|_\infty$. We have

$$\|A_1\|_\infty \leq \|(Z_n^* S_n - HC_n)(C + \lambda_2 I)^{-1/2}\|_\infty \times \|(C + \lambda_2 I)^{1/2}(C_n + \lambda_2 I)^{-1} S^*\|_\infty \tag{6.12}$$

1.1 Bound $\|(Z_n^* S_n - HC_n)(C + \lambda_2 I)^{-1/2}\|_\infty$. We define

$$\xi_i = \epsilon_i \otimes \phi(x_i)(C + \lambda_2 I)^{-1/2} \tag{6.13}$$

with $\epsilon_i = z_i - h^*(x_i)$. In this way,

$$\|(Z_n^* S_n - HC_n)(C + \lambda_2 I)^{-1/2}\|_\infty = \|\frac{1}{n}\sum_{i=1}^{n} \xi_i\|_\infty \tag{6.14}$$

We aim at applying the Bernstein inequality given in Theorem 14 to the random linear operator ξ. So, we define

$$T = 2\kappa Q \lambda_2^{-1/2} \geq \|\xi_i\|_\infty, \tag{6.15}$$

$$\sigma^2 = \max(\|\mathbb{E}[\xi\xi^*]\|_\infty, \|\mathbb{E}[\xi^*\xi]\|_\infty), \tag{6.16}$$

$$d = \mathrm{Tr}(\mathbb{E}[\xi^*\xi] + \mathbb{E}[\xi\xi^*])/\sigma^2. \tag{6.17}$$

Note that $\|\epsilon\| \leq \|z\|_{\mathcal{Z}} + \|h^*(x)\|_{\mathcal{Z}} \leq 2Q$, and $\|\phi(x)\| \leq \kappa$. Then, we have

$$\|\mathbb{E}[\xi\xi^*]\|_\infty = \|\mathbb{E}[\epsilon_i \otimes \epsilon_i \times \langle \phi(x_i), (C + \lambda_2 I)^{-1}\phi(x_i)\rangle_{\mathcal{H}_x}]\|_\infty \tag{6.18}$$

$$\leq \|\mathbb{E}[\epsilon \otimes \epsilon]\|_\infty \times \frac{\kappa^2}{\lambda_2}. \tag{6.19}$$

and

$$\|\mathbb{E}[\xi^*\xi]\|_\infty = \|(C + \lambda_2 I)^{-1/2} C (C + \lambda_2 I)^{-1/2}\|_\infty \times \mathbb{E}[\|\epsilon\|_{\mathcal{Z}}^2] \tag{6.20}$$

$$\leq 4Q^2. \tag{6.21}$$

Moreover, if $\lambda_2 < \|C\|_\infty$,

$$d \leq \frac{\mathrm{Tr}(\mathbb{E}[\xi^*\xi])}{\|\mathbb{E}[\xi^*\xi]\|_\infty} + \frac{\mathrm{Tr}(\mathbb{E}[\xi\xi^*])}{\|\mathbb{E}[\xi\xi^*]\|_\infty} \leq \frac{2\,\mathrm{Tr}(C)}{\lambda_2} + \frac{\mathrm{Tr}(E)}{\|E\|_\infty}. \tag{6.22}$$

Thus, by applying the Bernstein inequality given in Theorem 6.11, we have

$$\|(Z_n^* S_n - HC_n)(C + \lambda_2 I)^{-1/2}\|_\infty \leq \sqrt{\frac{2\eta(4Q^2 + \|E\|_\infty \kappa^2 \lambda_2^{-1})}{n}} + \frac{4\kappa Q \lambda_2^{-1/2}\eta}{3n} \tag{6.23}$$

where $\eta = \log(\frac{4(\frac{2\,\mathrm{Tr}(C)}{\lambda_2} + \frac{\mathrm{Tr}(E)}{\|E\|_\infty})}{\delta})$, $E = \mathbb{E}[\epsilon \otimes \epsilon]$.

1.2 Bound $\|(C+\lambda_2 I)^{1/2}(C_n+\lambda_2 I)^{-1}S^*\|_\infty$. We apply Lemma B.6 in Ciliberto et al. (2020), with $\lambda_2 \geq \frac{9\kappa^2}{n}\log(\frac{n}{\delta})$, and get with probability at least $1-\delta$,

$$\|(C+\lambda_2 I)^{1/2}(C_n+\lambda_2 I)^{-1}S^*\|_\infty \leq \|(C_n+\lambda_2 I)^{-1/2}(C+\lambda_2)^{1/2}\|_\infty^2 \leq 2. \tag{6.24}$$

Finally, we have

$$\|A_1\|_\infty \leq \sqrt{\frac{24\eta(Q^2+\|E\|_\infty\kappa^2\lambda_2^{-1})}{n}} + \frac{8\kappa Q\lambda_2^{-1/2}\eta}{3n}. \tag{6.25}$$

Bound $\|A_2\|_\infty$. We have

$$\begin{aligned}
\|A_2\|_\infty &= \|H(C_n(C_n+\lambda_2 I)^{-1}-I)S^*\|_\infty && (6.26)\\
&= \|H(-\lambda_2(C_n+\lambda_2 I)^{-1})S^*\|_\infty && (6.27)\\
&\leq \lambda_2\|H\|_\infty\|(C_n+\lambda_2 I)^{-1}S^*\|_\infty && (6.28)
\end{aligned}$$

and

$$\begin{aligned}
\|(C_n+\lambda_2 I)^{-1}S^*\|_\infty &\leq \lambda_2^{-1/2}\|(C_n+\lambda_2 I)^{-1/2}S^*\|_\infty && (6.29)\\
&= \lambda_2^{-1/2}\|(C_n+\lambda_2 I)^{-1/2}C^{1/2}\|_\infty && (6.30)\\
&\leq \lambda_2^{-1/2}\|(C_n+\lambda_2 I)^{-1/2}(C+\lambda_2)^{1/2}\|_\infty && (6.31)\\
&\leq \sqrt{2}\lambda_2^{-1/2} && (6.32)
\end{aligned}$$

because $\|(C_n+\lambda_2 I)^{-1/2}(C+\lambda_2)^{1/2}\|_\infty^2 \leq 2$ from Equation (6.24).

Finally, we have

$$\|A_2\|_\infty = \sqrt{2}\sqrt{\lambda_2}\|H\|_\infty. \tag{6.33}$$

Conclusion. The bound on $\|(H_n-H)S^*\|_\infty$ is obtained by summing the two bounds on $\|A1\|_\infty$ and $\|A2\|_\infty$. ■

We are now ready to prove a bound on the excess-risk of the ridge estimator on the random subspace defined by $\hat{P}$, namely $\|\hat{P}(H_n-H)S^*\|_{\mathrm{HS}}$.

Lemma 6.5 (KRR excess-risk on a subspace)**.** *Let $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ be a bounded kernel with $\forall x\in\mathcal{X}, k(x,x)\leq\kappa^2$. Let ρ be a distribution on $\mathcal{X}\times\mathcal{Z}$ such that its marginal w.r.t z is supported on the ball $\|z\|_{\mathcal{Z}}\leq Q$. Let $\hat{h}$ be the KRR estimator trained with n independent couples drawn from ρ. Let $\delta\in[0,1]$. Define $S_p(E)=\sum_{i=1}^p \mu_i(E)$. Then, under the Assumptions 4.1, 4.3, 4.4, taking for n big enough $\lambda_2 = \max(S_p(E)^{1/2}n^{-1/2}, n^{-1}, \frac{9\kappa^2}{n}\log(\frac{n}{\delta}))$, then with probability at least $1-2\delta$*

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x)-\hat{P}h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \left(c_4\sqrt{p}n^{-1/4}+c_5 S_p(E)^{1/4}\right)n^{-1/4}\log(n/\delta)$$

with $c_4 = (7Q + 4\kappa Q + 2\|H\|_{\mathrm{HS}}(1+3\kappa))(1+c_6)$, $c_5 = 10\sqrt{(1+c_6)}\kappa\|E\|_\infty^{1/2} + 2\|H\|_{\mathrm{HS}}$, $c_6 = \log(8(\frac{\mathrm{Tr}(C)}{\|E\|_\infty^{1/2}} + \frac{\mathrm{Tr}(E)}{\|E\|_\infty}))$.

Proof

Decomposition. We decompose $\|\hat{P}(H_n - H)S^*\|_{\mathrm{HS}}$ as follows

$$\|\hat{P}(H_n - H)S^*\|_{\mathrm{HS}} \leq \|\hat{P}A_1\|_{\mathrm{HS}} + \|\hat{P}A_2\|_{\mathrm{HS}} \tag{6.34}$$

with A_1, A_2 defined above in Lemma 6.4. Then, let be $t_1, t_2 > 0$,

$$\|\hat{P}A_1\|_{\mathrm{HS}} \leq \|\hat{P}(E + t_1 I)^{1/2}\|_{\mathrm{HS}} \times \|(E + t_1 I)^{-1/2}A_1\|_\infty \tag{6.35}$$

$$= \mathrm{Tr}(\hat{P}(E + t_1 I))^{1/2} \times \|(E + t_1 I)^{-1/2}A_1\|_\infty \tag{6.36}$$

$$\leq \sqrt{S_p(E) + pt_1} \times \|(E + t_1 I)^{-1/2}A_1\|_\infty. \tag{6.37}$$

and similarly

$$\|\hat{P}A_2\|_{\mathrm{HS}} \leq \sqrt{S_p(HH^*) + pt_2} \times \|(HH^* + t_2 I)^{-1/2}A_2\|_\infty. \tag{6.38}$$

Sketch of the following proof. We are going to bound $\|(E + t_1 I)^{-1/2}A_1\|_\infty$ and $\|(HH^* + t_2 I)^{-1/2}A_2\|_\infty$, using the Lemma 6.4 two times. This is done noticing that $\|(E + t_1 I)^{-1/2}A_1\|_\infty$ is exactly the error "part A_1" of the KRR estimator trained with data $(x_i, (E + t_1 I)^{-1/2}z)_{i=1}^n$, trying to solve the least-squares problem : $(E + t_1 I)^{-1/2}z = (E + t_1 I)^{-1/2}H\phi(x) + (E + t_1 I)^{-1/2}\epsilon$. The same trick is used for $\|(HH^* + t_2 I)^{-1/2}A_2\|_\infty$. In the two cases, we compute then the resulting modified constants in the bound because of these left linear operators multiplications.

1. Bound $\|(E + t_1 I)^{-1/2}A_1\|_\infty$. We apply Lemma 6.4 on the KRR estimator trained with $(x_i, (E + t)^{-1/2}z_i)$.

We have

$$\|(E + t_1 I)^{-1/2}E(E + t_1 I)^{-1/2}\|_\infty \leq 1 \tag{6.39}$$

$$\|(E + t_1 I)^{-1/2}z\| \leq t_1^{-1/2}Q \tag{6.40}$$

$$\|(E + t_1 I)^{-1/2}H\|_{\mathrm{HS}} \leq t_1^{-1/2}\|H\|_{\mathrm{HS}}. \tag{6.41}$$

Furthermore, if $\|E\|_\infty \geq t_1$, we have

$$\frac{\mathrm{Tr}(E(E + t_1)^{-1})}{\|E(E + t_1)^{-1}\|_\infty} = \mathrm{Tr}(E(E + t_1)^{-1})\frac{\|E\|_\infty + t}{\|E\|_\infty} \tag{6.42}$$

$$\leq 2\,\mathrm{Tr}(E(E + t_1)^{-1}) \tag{6.43}$$

$$\leq 2\,\mathrm{Tr}(E)t_1^{-1}. \tag{6.44}$$

Thus we get with probability at least $1 - 2\delta$

$$\|(E + t_1)^{-1/2}A_1\|_\infty \leq \sqrt{\frac{24\eta(Q^2 t_1^{-1} + \lambda_2^{-1}\kappa^2\|E\|_\infty)}{n}} + \frac{8\kappa Q t_1^{-1/2}\eta}{3\sqrt{\lambda_2}n}. \tag{6.45}$$

with $\eta = \log(\frac{8(\frac{\mathrm{Tr}(C)}{\lambda_2} + \frac{\mathrm{Tr}(E)}{t_1})}{\delta})$, $E = \mathbb{E}[\epsilon \otimes \epsilon]$, $\epsilon = z - h^*(x)$.

2. Bound $\|(HH^* + t_2 I)^{-1/2} A_2\|_\infty$. We apply Lemma 6.4 on the KRR estimator trained with $(x_i, (HH^* + t_2)^{-1/2} z_i)$. We have

$$\|(HH^* + t_2 I)^{-1/2} H\|_\infty = \|(HH^* + t_2 I)^{-1} HH^*\|_\infty^{1/2} \leq 1. \tag{6.46}$$

So,

$$\|(HH^* + t_2 I)^{-1/2} A_2\|_\infty \leq \sqrt{2}\sqrt{\lambda_2} \tag{6.47}$$

Conclusion. We conclude by summing the bound. We have

$$\begin{aligned}\|\hat{P}(H_n - H)S^*\|_{\mathrm{HS}} \leq \sqrt{S_p(E) + pt_1} \times \left(\sqrt{\frac{24\eta(Q^2 t_1^{-1} + \lambda_2^{-1}\kappa^2 \|E\|_\infty)}{n}} + \frac{8\kappa Q t_1^{-1/2}\eta}{3\sqrt{\lambda_2} n} \right) \\ + \sqrt{S_p(HH^*) + pt_2} \times \left(\sqrt{\lambda_2}\sqrt{2} \right).\end{aligned}$$

Taking $t_1 = p^{-1} S_p(E) \leq \|E\|_\infty$, and $t_2 = p^{-1} S_p(HH^*)$, we get

$$\begin{aligned}\|\hat{P}(H_n - H)S^*\|_{\mathrm{HS}} \leq \sqrt{\frac{48\eta(Q^2 p + 2S_p(E)\lambda_2^{-1}\kappa^2 \|E\|_\infty)}{n}} + \frac{4\kappa Q\sqrt{p}\eta}{\sqrt{\lambda_2} n} \\ + 2\sqrt{S_p(HH^*)}\sqrt{\lambda_2}.\end{aligned}$$

Now, taking $\lambda_2 = \max(S_p(E)^{1/2} n^{-1/2}, n^{-1}, \frac{9\kappa^2}{n}\log(\frac{n}{\delta}))$, we get

$$\begin{aligned}\|\hat{P}(H_n - H)S^*\|_{\mathrm{HS}} &\leq 7\sqrt{\frac{\eta Q^2 p}{n}} + 7\sqrt{\frac{2\eta S_p(E)\lambda_2^{-1}\kappa^2\|E\|_\infty}{n}} + \frac{4\kappa Q\sqrt{p}\eta}{\sqrt{\lambda_2} n} \\ &\quad + 2\|H\|_{\mathrm{HS}}\sqrt{\lambda_2} \\ &\leq 7\sqrt{\frac{\eta Q^2 p}{n}} + 7\sqrt{\frac{2\eta S_p(E)^{1/2}\kappa^2\|E\|_\infty}{n^{1/2}}} + \frac{4\kappa Q\sqrt{p}\eta}{n^{1/2}} \\ &\quad + 2\|H\|_{\mathrm{HS}}\left(S_p(E)^{1/4} n^{-1/4} + n^{-1/2} + 3\kappa n^{-1/2}\log^{1/2}(\frac{n}{\delta}) \right) \\ &\leq \left[\left(7\sqrt{\eta} Q + 4\kappa Q\eta + 2\|H\|_{\mathrm{HS}}(1 + 3\kappa\log(\frac{n}{\delta})) \right)\sqrt{p} n^{-1/4} \right. \\ &\quad \left. + \left(10\sqrt{\eta}\kappa\|E\|_\infty^{1/2} + 2\|H\|_{\mathrm{HS}} \right) S_p(E)^{1/4} \right] n^{-1/4} \\ &\leq \left(c_4\sqrt{p} n^{-1/4} + c_5 S_p(E)^{1/4} \right) n^{-1/4}\log(n/\delta)\end{aligned}$$

with $c_4 = (7Q + 4\kappa Q + 2\|H\|_{\mathrm{HS}}(1 + 3\kappa))(1 + c_6)$, $c_5 = 10\sqrt{(1 + c_6)}\kappa\|E\|_\infty^{1/2} + 2\|H\|_{\mathrm{HS}}$, $c_6 = \log(8(\frac{\mathrm{Tr}(C)}{\|E\|_\infty^{1/2}} + \frac{\mathrm{Tr}(E)}{\|E\|_\infty}))$, as $\eta \leq c_6 + \log(n/\delta) \leq (c_6 + 1)\log(n/\delta)$ if $p \leq n$.

■

6.2.3 Supervised Subspace Learning

In this subsection we prove a bound on the supervised reconstruction error:

$$\mathbb{E}_x[\|\hat{P}h^*(x) - h^*(x)\|_{\mathcal{Y}}^2]^{1/2} = \|(\hat{P} - I)M^{1/2}\|_{\mathrm{HS}}. \tag{6.48}$$

We use the proof scheme of Rudi et al. (2013) for subspace learning, retaking also the Lemma 6.6 restated just below. The novelty to deal with is that the random variable, whose reconstruction error is minimized here, is $h^*(x)$. The unknown $h^*(x_i)$ are estimated via our supervised subspace learning method (4.7) thanks to the couples $(x_i, y_i)_{i=1}^n$. This leads to additional derivations in our proofs.

We start by restating the Lemma 3.6 from Rudi et al. (2013) in a convenient form for our purposes.

Lemma 6.6 (Convergence of covariance operators)**.** *Let $\mathcal{X}, \mathcal{Y}$ be two Hilbert spaces, $H \in \mathcal{Y} \otimes \mathcal{X}$, $A = \mathbb{E}_x[Hx \otimes Hx]$, $(x_i)_{i=1}^n$ i.i.d from a distribution ρ on $\mathcal{X}$ supported on the unit ball, $A_n = \frac{1}{n}\sum_{i=1}^n Hx_i \otimes Hx_i$, $B \in \mathcal{Y} \otimes \mathcal{Y}$ any positive semidefinite operator, $\frac{9}{n}\log(\frac{n}{\delta}) \leq t \leq \|A\|_\infty$, then with probability at least $1 - \delta$ it is*

$$\sqrt{\frac{2}{3}} \leq \|(A + B + tI)^{\frac{1}{2}}(A_n + B + tI)^{-\frac{1}{2}}\|_\infty \leq \sqrt{2}$$

Proof By defining $B_n = (A + B + tI)^{-\frac{1}{2}}(A - A_n)(A + B + tI)^{-\frac{1}{2}}$, we have

$$\|(A + B + tI)^{\frac{1}{2}}(A_n + B + tI)^{-\frac{1}{2}}\|_\infty = \|(I - B_n)^{-1}\|_\infty^{1/2} \tag{6.49}$$

and B is positive semidefinite so

$$\|B_n\|_\infty = \|(A + B + tI)^{-\frac{1}{2}}(A - A_n)(A + B + tI)^{-\frac{1}{2}}\|_\infty \tag{6.50}$$

$$\leq \|(A + tI)^{-\frac{1}{2}}(A - A_n)(A + tI)^{-\frac{1}{2}}\|_\infty \tag{6.51}$$

Now, by applying Lemma 3.6 from Rudi et al. (2013), we get with probability at least $1 - \delta$, if $\frac{9}{n}\log(\frac{n}{\delta}) \leq t \leq \|A\|_\infty$

$$\|B_n\|_\infty \leq \frac{1}{2}. \tag{6.52}$$

We conclude by observing that

$$\frac{1}{\sqrt{1 + \|B_n\|_\infty}} \leq \|(I - B_n)^{-1}\|_\infty^{1/2} \leq \frac{1}{\sqrt{1 - \|B_n\|_\infty}}. \tag{6.53}$$

■

The two following lemmas handle the estimation of $M = HCH^* = \mathbb{E}[h^*(x) \otimes h^*(x)]$ in our supervised subspace learning method. In particular, here is exploited the Assumption 4.3, whose the divergence rate of the plateau threshold p_{max}, from which the error remains constant (See Rudi et al. (2013)), depends on.

Lemma 6.7. *Let be $\xi > 0, \delta \in [0,1]$. Under Assumptions 4.1, 4.3, taking*

$$t \geq n^{-\frac{1}{\beta+1}} (\xi/2)^{-\frac{4}{\beta+1}} \left(4\kappa(Q+\kappa R)\left(1 + 2\kappa \|M\|_\infty^{\frac{1}{4}(1-\beta)}\right) \log^2 \frac{8}{\delta} + c_2^{1/2} \right)^{\frac{4}{\beta+1}} \tag{6.54}$$

and

$$\lambda_1 = t^{-\frac{1-\beta}{2}} n^{-1/2}$$

is enough to achieve with probability at least $1-\delta$

$$\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)S^*\|_\infty \leq \xi. \tag{6.55}$$

Proof We note for convenience $A = (HCH^* + tI)^{-\frac{1}{2}}$. We proceed as in the proof of Lemma 18 and Theorem 5 in (Ciliberto et al., 2016) (showing a learning bound for the kernel ridge estimator). However, we monitor the action of A, and we use Assumption 4.3, in order to obtain the best bound w.r.t t and n, decreasing fast when n and t increase. We have

$$\begin{aligned} \|A(H_n - H)S^*\|_\infty &= \|AZ_n^* S_n (C_n + \lambda_1)^{-1} S^* - AZ^*\|_\infty && (6.56)\\ &\leq (\mathrm{I}) + (\mathrm{II}) + (\mathrm{III}) && (6.57) \end{aligned}$$

with

$$\begin{aligned} (\mathrm{I}) &= \|AZ_n^* S_n (C_n + \lambda_1)^{-1} - AZ^* S (C_n + \lambda_1)^{-1} S^*\|_\infty \\ &\leq \sqrt{\frac{1}{t}} \times \|Z_n^* S_n (C_n + \lambda_1)^{-1} - Z^* S (C_n + \lambda_1)^{-1} S^*\|_{\mathrm{HS}} \\ (\mathrm{II}) &= \|AZ^* S (C_n + \lambda_1)^{-1} S^* - AZ^* S (C + \lambda_1)^{-1} S^*\|_\infty \\ &\leq \sqrt{\frac{1}{t}} \times \|Z^* S (C_n + \lambda_1)^{-1} S^* - Z^* S (C + \lambda_1)^{-1} S^*\|_{\mathrm{HS}} \\ (\mathrm{III}) &= \|AZ^* S (C + \lambda_1)^{-1} S^* - AZ^*\|_\infty \end{aligned}$$

Bound (III). From Assumption 4.1 we have $Z^* = HS^*$, and

$$\begin{aligned} (\mathrm{III}) &= \|AZ^*(S(C + \lambda_1)^{-1} S^* - I)\|_\infty && (6.58)\\ &= \|AHS^*(S(C + \lambda_1)^{-1} S^* - I)\|_\infty && (6.59)\\ &= \|AH(S^* - \lambda_1 (C + \lambda_1)^{-1} - S^*)\|_\infty && (6.60)\\ &= \lambda_1 \|AH(C + \lambda_1)^{-1} S^*\|_\infty && (6.61)\\ &\leq \|AH\|_\infty \times \lambda_1 \|(C + \lambda_1)^{-1} S^*\|_\infty && (6.62)\\ &\leq \|AH\|_\infty \times \sqrt{\lambda_1}. && (6.63) \end{aligned}$$

Using Assumption 4.3 we have

$$\begin{aligned} \|AH\|_\infty &= \|(M + tI)^{-\frac{1}{2}} H\|_\infty && (6.64)\\ &\leq \|(HCH^* + tI)^{-\frac{1}{2}} c_2^{1/2} M^{(1-\beta)/2}\|_\infty && (6.65)\\ &\leq c_2^{1/2} \times t^{-\frac{\beta}{2}}. && (6.66) \end{aligned}$$

Bound (I) and (II). We bound (I) and (II), as in Ciliberto et al. (2016) (Lemma 18).

Conclusion. This leads to the following bound with probability at least $1-\delta$:

$$\|A(H_n - H)S^*\|_\infty \leq 4\kappa \frac{Q+\kappa R}{\sqrt{\lambda_1 nt}}\left(1+\sqrt{\frac{4\kappa^2}{\lambda_1\sqrt{n}}}\right)\log^2\frac{8}{\delta} + c_2^{1/2}\sqrt{\lambda_1}t^{-\frac{\beta}{2}}. \tag{6.67}$$

Now, choosing $\lambda_1 = \frac{t^{-\frac{1}{2}(1-\beta)}}{\sqrt{n}}$, if $t \leq \|M\|_\infty$, we obtain

$$\|A(H_n - H)S^*\|_\infty \leq \left(4\kappa(Q+\kappa R)\left(1+2\kappa t^{\frac{1}{4}(1-\beta)}\right)\log^2\frac{8}{\delta} + c_2^{1/2}\right)n^{-1/4}t^{-\frac{1}{4}(\beta+1)} \tag{6.68}$$

$$\leq \left(4\kappa(Q+\kappa R)\left(1+2\kappa\|M\|_\infty^{\frac{1}{4}(1-\beta)}\right)\log^2\frac{8}{\delta} + c_2^{1/2}\right)n^{-1/4}t^{-\frac{1}{4}(\beta+1)} \tag{6.69}$$

Hence, taking $t \geq n^{-\frac{1}{\beta+1}}(\xi/2)^{-\frac{4}{\beta+1}}\left(4\kappa(Q+\kappa R)\left(1+2\kappa\|M\|_\infty^{\frac{1}{4}(1-\beta)}\right)\log^2\frac{8}{\delta} + c_2^{1/2}\right)^{\frac{4}{\beta+1}}$ is enough to achieve

$$\|A(H_n - H)S^*\|_\infty \leq \xi. \tag{6.70}$$

■

We combine Lemmas 6.6 and 6.7 to finally prove a concentration bound for $H_nC_nH_n^*$ deviating from HCH^*.

Lemma 6.8 (Convergence of the supervised covariance M_n)**.** *Let be $\delta \in [0,1]$. Under Assumptions 4.1, 4.3, and defining*

$$B_n = (HCH^* + tI)^{-\frac{1}{2}}(H_nC_nH_n^* - HCH^*)(HCH^* + tI)^{-\frac{1}{2}}$$

if $t \geq c_8\log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}}$, $n \geq n_0$ (constant independent of δ), then with probability $1-2\delta$

$$\|B_n\|_\infty \leq \frac{1}{2}$$

with $c_8 = (\xi/2)^{-\frac{4}{\beta+1}}\left(4\kappa(Q+\kappa R)\left(1+2\kappa\|M\|_\infty^{\frac{1}{4}(1-\beta)}\right) + c_2^{1/2}\right)^{\frac{4}{\beta+1}}$ and $\xi = \frac{1}{14}$, $n_0 \in \mathbb{N}^*$ constant defined in the proof.

Proof We decompose in 7 terms the difference of products, then we will bound each associated term in $\|B_n\|_\infty$.

$$\begin{aligned}
H_nC_nH_n^* - HC_nH^* = &(H_n - H)CH^* \quad (i)\\
&+ HC(H_n - H)^* \quad (ii)\\
&+ (H_n - H)C(H_n - H)^* \quad (iii)\\
&+ (H_n - H)(C_n - C)H^* \quad (iv)\\
&+ H(C_n - C)(H_n - H)^* \quad (v)\\
&+ (H_n - H)(C_n - C)(H_n - H)^* \quad (vi)\\
&+ H(C_n - C)H^* \quad (vii)
\end{aligned}$$

Bound (i) **and** (ii)**.**

$$\|(HCH^* + tI)^{-\frac{1}{2}} HC(H_n - H)^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty \leq \|(HCH^* + tI)^{-\frac{1}{2}} HS^*\|_\infty \times \|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)S^*\|_\infty$$

But:

$$\begin{aligned}\|(HCH^* + tI)^{-\frac{1}{2}} HS^*\|_\infty &= \|(HCH^* + tI)^{-\frac{1}{2}} HS^* SH^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty^{1/2} \\ &= \|(HCH^* + tI)^{-\frac{1}{2}} HCH^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty^{1/2} \\ &\leq 1\end{aligned}$$

And from Lemma 6.7, defining $c_8 = (\xi/2)^{-\frac{4}{\beta+1}} \left(4\kappa(Q + \kappa R)\left(1 + 2\kappa\|M\|_\infty^{\frac{1}{4}(1-\beta)}\right) + c_2^{1/2}\right)^{\frac{4}{\beta+1}}$, $\xi = 14$,if $t \geq c_8 \log^8(\frac{8}{\delta}) n^{-\frac{1}{\beta+1}}$ we get with probability at least $1 - \delta$

$$\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)S^*\|_\infty \leq \frac{1}{14}$$

Bound (iii)**.** As for (i) and (ii), from Lemma 6.7 we have

$$\begin{aligned}\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)C(H_n - H)^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty &\leq \|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)S^*\|_\infty^2 \\ &\leq \frac{1}{14^2} \leq \frac{1}{14}.\end{aligned}$$

Bound (iv) **and** (v)**.** We decompose

$$\begin{aligned}&\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)(C_n - C)H^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty \leq \\ &\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)C_t^{1/2}\|_\infty \times \|C_t^{-1/2}(C_n - C)C_t^{-1/2}\|_\infty \times \|C_t^{1/2}H^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty.\end{aligned}$$

We bound

$$\begin{aligned}\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)C_t^{1/2}\|_\infty &= \|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)C_t(H_n - H)^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty^{1/2} \\ &\leq \|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)S^*\|_\infty + t^{1/2}\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)\|_\infty \\ &\leq \|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)S^*\|_\infty + \|H_n - H\|_\infty\end{aligned}$$

and similarly,

$$\begin{aligned}\|C_t^{1/2}H^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty &\leq \|(HCH^* + tI)^{-\frac{1}{2}}HS^*\|_\infty + t^{1/2}\|(HCH^* + tI)^{-\frac{1}{2}}cH\|_\infty \\ &\leq \|(HCH^* + tI)^{-\frac{1}{2}}HS^*\|_\infty + \|H\|_\infty \\ &\leq 1 + \|H\|_\infty\end{aligned}$$

finally we obtain

$$\begin{aligned}&\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)(C_n - C)H^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty \leq \\ &\left(\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)S^*\|_\infty + \|H_n - H\|_\infty\right) \\ &\times \|C_t^{-1/2}(C_n - C)C_t^{-1/2}\|_\infty \\ &\times \left(1 + \|H\|_\infty\right).\end{aligned}$$

From Lemma 6.7, $\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)(C_n - C)H^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty \leq 1/14$ if $t \geq c_8 \log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}}$. From Lemma 6.12, $\|H_n - H\|_\infty \leq 2\log^8(\frac{8}{\delta})R$ if $n \geq n_1$ with n_1 a constant independent of δ. So, defining $u = (1/14 + 2R) \times (1 + R)$. Now, using Lemma 3.6 from Rudi et al. (2013), we can have $\|C_t^{-1/2}(C_n - C)C_t^{-1/2}\|_\infty \leq 1/14 \times u^{-1}\log^{-8}(\frac{8}{\delta})$ if $t \geq a_1 \frac{\log n/\delta}{n}$ with $a_1 > 0$ a constant independent of δ. We conclude that

$$\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)(C_n - C)H^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty \leq \frac{1}{14}.$$

Bound (vi)**.** Similarly as for (v), we have

$$\begin{aligned}&\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)(C_n - C)(H_n - H)^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty \leq \\ &\left(\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)S^*\|_\infty + \|H_n - H\|_\infty\right)^2 \\ &\times \|C_t^{-1/2}(C_n - C)C_t^{-1/2}\|_\infty\end{aligned}$$

and, if $t \geq a_2 \frac{\log n/\delta}{n}$, with a_2 a constant independent of δ, we also have

$$\|(HCH^* + tI)^{-\frac{1}{2}}(H_n - H)(C_n - C)(H_n - H)^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty \leq 1/14.$$

Bound (vii)**.** As previously, from Lemma 3.6 from Rudi et al. (2013), there exists a constant $a_3 > 0$ such that with probability at least $1 - \delta$ if $t \geq a_3 \frac{\log n/\delta}{n}$, with $a_3 > 0$, we have

$$\|(HCH^* + tI)^{-\frac{1}{2}}H(C_n - C)H^*(HCH^* + tI)^{-\frac{1}{2}}\|_\infty \leq 1/14.$$

Conclusion. But there exists n_0 independent of δ such that $\forall n \geq n_0 \geq n_1$, $c_8 \log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}} \geq \max(a_1, a_2, a_3)\frac{\log n/\delta}{n}$. So, we conclude that, if $t \geq c_8 \log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}}$, and $n \geq n_0$,

$$\|B_n\|_\infty \leq \frac{1}{2}.$$

■

We are now ready to prove the main result of this section. We prove a bound on the reconstruction error of $\hat{P}$ when reconstructing the $h^*(x)$, namely $\mathbb{E}_x[\|\hat{P}h^*(x) - h^*(x)\|_{\mathcal{Y}}^2]^{1/2}$.

Lemma 6.9 (Supervised subspace learning)**.** *Let $(x_i, y_i)_{i=1}^n$ be drawn independently from a probability measure ρ and $(y_i)_{i=1}^m$ be drawn independently from the marginal ρ w.r.t y with support in the ball $\|y\|_{\mathcal{Y}} \leq Q$. Let $\hat{P}$ be the estimated projection in the proposed method. Then, under Assumptions 4.1, 4.2 and 4.3, there exist constants $c_8 > 0, n_0 \in \mathbb{N}^*$, such that, if $\mu_{p+1}(M) \geq c_8 \log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}}$, $n \geq n_0$, $\lambda_1 = \mu_{p+1}(M)^{-\frac{1-\beta}{2}}n^{-\frac{1}{2}}$, then with probability at least $1 - 3\delta$*

$$\mathbb{E}_x[\|\hat{P}h^*(x) - h^*(x)\|_{\mathcal{Y}}^2]^{\frac{1}{2}} \leq \sqrt{3c_1}\mu_{p+1}(M)^{1/2(1-\alpha)}.$$

Proof We have (See Proposition C.4. in Rudi et al. (2013)):

$$\mathbb{E}_x[\|\hat{P}h^*(x) - h^*(x)\|]^{1/2} = \|(\hat{P} - I)M_c^{\frac{1}{2}}\|_{\text{HS}}^2 \tag{6.71}$$

Then, as in the proofs of Rudi et al. (2013), we split (6.71) into three parts, and bound each term,

$$\|(\hat{P}-I)M^{\frac{1}{2}}\|_{\mathrm{HS}} \leq \underbrace{\|(M+tI)^{\frac{1}{2}}(M_n+tI)^{-\frac{1}{2}}\|_\infty}_{\mathcal{A}} \times \underbrace{(\mu_{p+1}(M_n)+t)^{\frac{1}{2}}}_{\mathcal{B}} \times \underbrace{\|(M+tI)^{-\frac{1}{2}}M^{\frac{1}{2}}\|_{\mathrm{HS}}}_{\mathcal{C}}$$

Bound $\mathcal{A} = \|(M+tI)^{\frac{1}{2}}(M_n+tI)^{-\frac{1}{2}}\|_\infty$. We have:

$$\begin{aligned}\|(M+tI)^{\frac{1}{2}}(M_n+tI)^{-\frac{1}{2}}\|_\infty &= \|(M+tI)^{\frac{1}{2}}(M_n+tI)^{-1}(M+tI)^{\frac{1}{2}}\|_\infty^{1/2} \\ &= \|(I-B_n)^{-1}\|_\infty^{1/2}\end{aligned}$$

with $B_n = (M+tI)^{-1/2}(M-M_n)(M+tI)^{-1/2}$. So, if $\|B_n\|_\infty < 1$,

$$\frac{1}{\sqrt{1+\|B_n\|_\infty}} \leq \|(M+tI)^{\frac{1}{2}}(M_n+tI)^{-\frac{1}{2}}\|_\infty \leq \frac{1}{\sqrt{1-\|B_n\|_\infty}}$$

Then applying Lemma 6.8, if $t \geq c_8 \log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}}$, with probability $1-3\delta$ it is

$$\sqrt{\frac{2}{3}} \leq \|(M+tI)^{\frac{1}{2}}(M_n+tI)^{-\frac{1}{2}}\|_\infty \leq \sqrt{2}$$

Bound $\mathcal{B} = (\mu_{p+1}(M_n)+t)^{\frac{1}{2}}$. $\sqrt{\frac{2}{3}} \leq \|(M+tI)^{\frac{1}{2}}(M_n+tI)^{-\frac{1}{2}}\|_\infty$ is equivalent to $M_n + t \leq \frac{3}{2}(M+t)$ (by Lemma B.2 point 4 in (Rudi et al., 2013)). Then, $\forall k \in \mathbb{N}^*, \mu_k(M_n+t) \leq \frac{3}{2}\mu_k(M+t)$, so we have

$$\sqrt{\mu_{p+1}(M_n)+t} \leq \sqrt{\frac{3}{2}}\sqrt{\mu_{p+1}(M)+t}. \tag{6.72}$$

Bound $\mathcal{C} = \|(M+tI)^{-\frac{1}{2}}M^{\frac{1}{2}}\|_{\mathrm{HS}}$. We have

$$\begin{aligned}\mathcal{C}^2 &= \mathrm{Tr}(M(M+t)^{-1}) && (6.73)\\ &= \mathrm{Tr}(M^\alpha M^{1-\alpha}(M+t)^{-1}) && (6.74)\\ &\leq \mathrm{Tr}(M^\alpha)\|M^{1-\alpha}(M+t)^{-1}\|_\infty && (6.75)\\ &\leq c_1 \times t^{-\alpha} \quad \text{(from Assumption 4.2 and Young's inequality for products).} && (6.76)\end{aligned}$$

Finally, we get the following upper bound.

$$\mathbb{E}_x[\|\hat{P}h^*(x) - h^*(x)\|]^{1/2} \leq \sqrt{3}\sqrt{\mu_{p+1}(M)+t} \times c_1^{1/2} \times t^{-\alpha/2} \tag{6.77}$$

Taking $t = \mu_{p+1}(M)$, which is possible if $\mu_{p+1}(M) \geq c_8 \log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}}$, we get

$$\mathbb{E}_x[\|\hat{P}h^*(x) - h^*(x)\|]^{1/2} \leq \sqrt{3c_1}\mu_{p+1}(M)^{1/2(1-\alpha)}. \tag{6.78}$$

We get the wanted upper bound.

■

6.2.4 Theorem

In this subsection we give the main result of this chapter which is a learning bound for the proposed method. That is we bound:

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]. \tag{6.79}$$

The proof consists in decomposing this excess-risk in two terms, as in equation (4.27), then bounding each term applying the two lemmas previously proved.

Theorem 4.7 (Learning bounds)**.** *Let $\hat{P}\hat{h}$ be the proposed estimator in Eq.* (4.8) *with* $\operatorname{rank}(\hat{P}) = p$*, built from n independent couples $(x_i, z_i)_{i=1}^n$ drawn from ρ. Let $\delta \in [0,1]$. Under the Assumptions 4.1, 4.2, 4.3, 4.4, there exists constants $c_4, c_5, c_8 > 0$, $n_0 \in \mathbb{N}^*$ defined in the proof, and independent of p, n, δ, such that, if $\mu_{p+1}(M) \geq c_8 \log^8(\frac{8}{\delta}) n^{-\frac{1}{\beta+1}}$ and $n \geq n_0$, then with probability at least $1 - 3\delta$,*

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \left(c_4\sqrt{p}n^{-1/4} + c_5 S_p(E)^{1/4}\right) n^{-1/4}\log(n/\delta) + \sqrt{3c_1}\mu_{p+1}(M)^{1/2(1-\alpha)} \tag{4.13}$$

with $S_p(E) = \sum_{i=1}^p \mu_i(E)$.

Proof We decompose the excess-risk as follows

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \underbrace{\mathbb{E}_x[\|\hat{P}\hat{h}(x) - \hat{P}h^*(x)\|_{\mathcal{Z}}^2]^{1/2}}_{\text{regr. error on a subspace}} + \underbrace{\mathbb{E}_x[\|\hat{P}h^*(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2}}_{\text{reconstruction error}}. \tag{6.80}$$

We apply the Lemmas 6.5 and 6.9, and we get, if $\mu_{p+1}(M) \geq c_8 \log^8(\frac{8}{\delta}) n^{-\frac{1}{\beta+1}}$, with probability at least $1 - 3\delta$:

$$\mathbb{E}_x[\|P\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \left(c_4\sqrt{p}n^{-1/4} + c_5 S_p(E)^{1/4}\right) n^{-1/4}\log(n/\delta) + \sqrt{3c_1}\mu_{p+1}(M)^{1/2(1-\alpha)}. \tag{6.81}$$

with $c_4 = (7Q + 4\kappa Q + 2\|H\|_{\mathrm{HS}}(1 + 3\kappa))(1 + c_6)$, $c_5 = 10\sqrt{(1 + c_6)}\kappa\|E\|_\infty^{1/2} + 2\|H\|_{\mathrm{HS}}$, $c_6 = \log(8(\frac{\operatorname{Tr}(C)}{\|E\|_\infty^{1/2}} + \frac{\operatorname{Tr}(E)}{\|E\|_\infty}))$.

■

6.2.5 Corollary

In this subsection we derive from the Theorem 4.7 a corollary in the case where M and E have polynomial eigenvalue decay rates. This allows to explicit the optimal quantity of components p, and also obtaining a condition on the decay rates $s, e > 1$ in order to obtain a statistical gain.

Corollary 4.9 (Learning bounds (polynomial decay rates))**.** *Let $\delta \in]0,1]$, $n \geq n_0$. Under Assumptions 4.1, 4.3, and 4.8, assuming $\frac{B}{b} \leq \theta$ with $\theta \geq 1$, then by taking only*

$$p = c_9(\log^8(\frac{8}{\delta}))^{-\frac{1}{s}} n^{\frac{1}{(\beta+1)s}}, \tag{4.17}$$

we have with probability at least $1-3\delta$*:*

$$\mathbb{E}_x[\|\hat{P}\hat{h}(x)-h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq c_{10}(s,e)\log^{5/4}(\frac{n}{\delta})n^{-1/4} + c_{11}(e)n^{-\frac{1}{2}\frac{1-2/s}{1+\beta}}\log^8(\frac{8}{\delta}), \tag{4.18}$$

where $c_{10}(s,e) = \tilde{c}_{10}\left(\frac{e(e-1)}{s}\right)^{1/4}\left(1+\log\left(\frac{e}{e-1}\right)\right)$, $c_{11}(e) = \tilde{c}_{11}\left(1+\log\left(\frac{e}{e-1}\right)\right)$. $\tilde{c}_{10}$, $\tilde{c}_{11}$, n_0, *are constants independent of* n,δ,s,e, *and* c_9 *is a constant independent of* n,δ, *defined in the proofs.*

Proof The proof consists in applying the Theorem 4.7 in the specific case of polynomial eigenvalue decay rates. If $\mu_{p+1}(M) \geq c_8\log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}}$, with probability at least $1-3\delta$:

$$\mathbb{E}_x[\|P\hat{h}(x)-h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \left(c_4\sqrt{p}n^{-1/4} + c_5 S_p(E)^{1/4}\right)n^{-1/4}\log(n/\delta) + \sqrt{3c_1}\mu_{p+1}(M)^{1/2(1-\alpha)}. \tag{6.82}$$

Bound $S_p(E)$**.** The polynomial eigenvalue decay assumption, give us that $\frac{a}{p^s} \leq \mu_p(M) \leq \frac{A}{p^s}$. So, Assumption 4.1 is verified with $\alpha = \frac{2}{s}$, and $c_1 = \mathrm{Tr}(M^\alpha) \leq \sum_i i^{-2}\times A^\alpha \leq 2A^\alpha$. Hence,

$$\sqrt{3c_1}\mu_{p+1}(M)^{1/2(1-\alpha)} \leq \frac{\sqrt{6A^\alpha}A^{1/2(1-\alpha)}}{p^{\frac{1}{\alpha}-1}} = \frac{\sqrt{6A}}{p^{\frac{s}{2}-1}}. \tag{6.83}$$

Moreover,

$$S_p(E) = \sum_{i=1}^{p}\mu_i(E) \leq B\sum_{i=1}^{p} i^{-e} \leq B(1+\int_{x=1}^{p} x^{-e}dx) \leq \frac{B}{1-e^{-1}}\times(1-\frac{e^{-1}}{p^{e-1}}) \tag{6.84}$$

and using $(1-1/x) \leq \log(x) \leq x-1$, we get

$$\begin{aligned}
S_p(E) &\leq \frac{B}{1-e^{-1}}\times\Big((e-1)\log(p)+\log(e)\Big) && (6.85)\\
&\leq \frac{B}{1-e^{-1}}\times\Big((e-1)\log(p)+(e-1)\Big) && (6.86)\\
&= \frac{B}{1-e^{-1}}\times(e-1)(\log(p)+1) && (6.87)\\
&\leq \frac{B}{1-e^{-1}}\times 2(e-1)\log(p) \quad (\text{if } p>3) && (6.88)\\
&= 2Be\log(p). && (6.89)
\end{aligned}$$

Now, taking $p = c_9(\log^8(\frac{8}{\delta}))^{-\frac{1}{s}}n^{\frac{1}{s(\beta+1)}}$, defining $c_9 = (\frac{c_8}{a})^{-\frac{1}{s}}$, ensures $\mu_p(M) \geq c_8\log^8(\frac{8}{\delta})n^{-\frac{1}{\beta+1}}$. Moreover, $B \leq \theta\times b \leq \theta\,\mathrm{Tr}(E)(\sum_{i=1}^{+\infty} i^{-e})^{-1} = \frac{\theta\,\mathrm{Tr}(E)}{\zeta(e)}$ by definition of the Riemann zeta function. So, using this defined p, we get,

$$\begin{aligned}
S_p(E) &\leq 2Be\left(\frac{1}{s}\log\left(\frac{a}{c_8}\right)+\frac{\log(n)}{s(\beta+1)}\right) && (6.90)\\
&\leq \frac{2\theta\,\mathrm{Tr}(E)e\log(n)}{\zeta(e)s}\left(\log\left(\frac{a}{c_8}\right)+1\right) \quad (\text{if } n>3) && (6.91)
\end{aligned}$$

Bound $\sqrt{3c_1}\mu_{p+1}(M)^{1/2(1-\alpha)}$. Now, taking $p = c_9(\log^8(\frac{8}{\delta}))^{-\frac{1}{s}} n^{\frac{1}{s(\beta+1)}}$, defining $c_9 = (\frac{c_8}{a})^{-\frac{1}{s}}$, ensures $\mu_p(M) \geq c_8 \log^8(\frac{8}{\delta}) n^{-\frac{1}{\beta+1}}$. Using this defined p, we get

$$\sqrt{3c_1}\mu_{p+1}(M)^{1/2(1-\alpha)} \leq \sqrt{6A}(\frac{c_8}{a}\log^8(\frac{8}{\delta}))^{\frac{1}{2}(1-\frac{2}{s})} n^{-\frac{1}{2}\frac{1-\frac{2}{s}}{1+\beta}} \tag{6.92}$$

$$\leq \sqrt{6A}(\sqrt{\frac{c_8}{a}}+1)\log^8(\frac{8}{\delta}) n^{-\frac{1}{2}\frac{1-\frac{2}{s}}{1+\beta}}. \tag{6.93}$$

Bound $\sqrt{p}n^{-1/2}$. Furthermore, one can check that $(\frac{1}{2} - \frac{1}{2s(\beta+1)}) > \frac{1}{2}\frac{1-2/s}{1+\beta}$, hence we have

$$\sqrt{p}n^{-1/2} \leq \left(\frac{a}{c_8}\right)^{1/2s} n^{-(\frac{1}{2}-\frac{1}{2s(\beta+1)})} \leq \left(\frac{a}{c_8}+1\right) n^{-\frac{1}{2}\frac{1-2/s}{1+\beta}}. \tag{6.94}$$

Studying c_4, c_5, c_8, n_0 dependencies in s, e. In this work we study the behavior of the bound when the shape of E and M vary, i.e. when s and e vary. Therefore, it's important to make some derivations to studying c_4, c_5, c_8, n_0's dependencies in s and e. First, c_8, n_0 are independent of δ, s, e.

Then, observing that we have $\|E\|_\infty^{-1} = \mu_1(E)^{-1} \leq b^{-1} \leq \frac{\theta}{B} \leq \theta\frac{\zeta(e)}{\mathrm{Tr}(E)}$, leads to $c_6 \leq \log(8(\frac{\theta^{1/2}\mathrm{Tr}(C)}{\mathrm{Tr}(E)^{1/2}}+\theta)) + \log(\zeta(e))$. So, we have

$$c_4 = (7Q + 4\kappa Q + 2R(1+3\kappa))(1+c_6) \tag{6.95}$$

$$\leq \Big(\log(\zeta(e))+1\Big)\left(1+\log(8(\frac{\theta^{1/2}\mathrm{Tr}(C)}{\mathrm{Tr}(E)^{1/2}}+\theta))\right)\Big(7Q+4\kappa Q+2R(1+3\kappa)\Big) \tag{6.96}$$

and also

$$c_5 = 10\sqrt{(1+c_6)}\kappa\|E\|_\infty^{1/2} + 2\|H\|_{\mathrm{HS}} \tag{6.97}$$

$$\leq \Big(\log(\zeta(e)+1)\Big)\left(1+\log(8(\frac{\theta^{1/2}\mathrm{Tr}(C)}{\mathrm{Tr}(E)^{1/2}}+\theta))\right)\Big(10\kappa\|E\|_{\mathrm{HS}}^{1/2} + 2\|H\|_{\mathrm{HS}}\Big). \tag{6.98}$$

Conclusion. Thanks to the previous derivations we obtain the following bound

$$\mathbb{E}_x[\|P\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq c_{10}(s,e)\log^{5/4}(\frac{n}{\delta})n^{-1/4} + c_{11}(e)n^{-\frac{1}{2}\frac{1-2/s}{1+\beta}}\log^8(\frac{8}{\delta})$$

with $c_{10}(s,e) = \tilde{c}_{10}(\log(\zeta(e))+1)\left(\frac{e}{\zeta(e)\times s}\right)^{1/4}$, $c_{11}(e) = \tilde{c}_{11}(\log(\zeta(e))+1)$. $\tilde{c}_{10}$ and $\tilde{c}_{11}$ are constants independent of n, δ, s, e, defined below

$$\tilde{c}_{10} = \left(1+\log(8(\frac{\theta^{1/2}\mathrm{Tr}(C)}{\mathrm{Tr}(E)^{1/2}}+\theta))\right)\Big(10\kappa\|E\|_{\mathrm{HS}}^{1/2}+2\|H\|_{\mathrm{HS}}\Big)\left(2\theta\,\mathrm{Tr}(E)(\log(\frac{a}{c_8})+1)\right)^{1/4} \tag{6.99}$$

$$\tilde{c}_{11} = \sqrt{6A}(\sqrt{\frac{c_8}{a}}+1) + \left(\frac{a}{c_8}+1\right)\left(1+\log(8(\frac{\theta^{1/2}\mathrm{Tr}(C)}{\mathrm{Tr}(E)^{1/2}}+\theta))\right)\Big(7Q+4\kappa Q+2R(1+3\kappa)\Big). \tag{6.100}$$

The inequalities $\frac{1}{e-1} \leq \zeta(e) \leq \frac{e}{e-1}$ allow to conclude the proof.

■

6.2.6 Auxiliary Results

In this section, we give four auxiliary results:

- A bound on the KRR estimator which monitors the role of the total amount of noise $\mathrm{Tr}(E)$.
- A Bernstein inequality for bounded operator and the operator norm.
- A bound on $\|H_n - H\|_\infty$, used in the proof of Lemma 6.9.
- Some properties of Löwner's partial ordering

Lemma 6.10 (Full-rank KRR excess-risk). *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a bounded kernel with $\forall x \in \mathcal{X}, k(x,x) \leq \kappa^2$. Let ρ be a distribution on $\mathcal{X} \times \mathcal{Z}$ such that its marginal w.r.t y is supported on the ball $\|y\|_{\mathcal{Z}} \leq Q$. Let $\hat{h}$ be the KRR estimator trained with n independent couples drawn from ρ. Let $\delta \in [0,1]$. Then, under the assumption 4.1 and 4.3, taking*

$$\lambda_1 = \max\left(\frac{1}{n}, \frac{\|E^{1/2}\|_{\mathrm{HS}}}{\sqrt{n}}\right) \tag{6.101}$$

the following holds with probability at least $1-\delta$

$$\mathbb{E}_x[\|\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{\frac{1}{2}} \leq C(p) n^{-\frac{1}{4}} \log\frac{4}{\delta} \tag{6.102}$$

with $C(p) = 10\left[\mathcal{O}(n^{-\frac{1}{4}}) + (\kappa + R)\|E^{1/2}\|_{\mathrm{HS}}^{\frac{1}{2}}\right]$, $R = \|H\|_{\mathrm{HS}}$.

Proof We follow the proofs of (Ciliberto et al., 2020) in order to derive a learning bound of the KRR estimator. We carefully monitor the role of the total amount of noise $\mathrm{Tr}(E)$.

We make appear the conditional variance by modifying the Proposition B.7 in (Ciliberto et al., 2020), with the following change from equation (B.55) to (B.58):

$$\mathbb{E}_x[\|C_\lambda^{-1/2}\phi(x)\|^2\sigma(x)^2] \leq \frac{\kappa^2}{\lambda} \times \mathbb{E}_x[\sigma(x)^2] \tag{6.103}$$

$$= \frac{\kappa^2}{\lambda} \times \mathbb{E}[\|\epsilon\|_{\mathcal{Z}}^2] \tag{6.104}$$

$$= \frac{\kappa^2}{\lambda} \times \|E^{1/2}\|_{\mathrm{HS}}^2 \tag{6.105}$$

by defining the noise $\epsilon = \psi(y) - h^*(x)$, and $E = \mathbb{E}[\epsilon \otimes \epsilon]$.

Then, doing the same proof than Theorem B.8 from (Ciliberto et al., 2020), we get the following bound

$$\begin{aligned}\mathbb{E}_x[\|P\hat{h}(x) - Ph^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \quad & \frac{8\kappa \log\frac{2}{\delta}}{\sqrt{\lambda}n} \times (Q + \kappa\|L_\lambda^{-1/2} Z\|_{\mathrm{HS}}) \\ & + \frac{1}{\sqrt{n}} \times \sqrt{64(d_{\mathrm{eff}}(\lambda) \times \|E^{1/2}\|_{\mathrm{HS}}^2 + \kappa^2\lambda\|L_\lambda^{-1} Z\|_{\mathrm{HS}}^2)\log\frac{4}{\delta}} \\ & + 10 \times \lambda\|L_\lambda^{-1} Z\|_{\mathrm{HS}}\end{aligned}$$

Now, using the assumption 1, we have

$$\begin{aligned}\|L_\lambda^{-1} Z\|_{\mathrm{HS}} &= \|L_\lambda^{-1} SH^*\|_{\mathrm{HS}} && (6.106)\\ &\leq \|L_\lambda^{-1} S\|_{\mathrm{HS}} \times \|H\|_{\mathrm{HS}} && (6.107)\\ &\leq \lambda^{-\frac{1}{2}} \times R && (6.108)\end{aligned}$$

and similarly $\|L_\lambda^{-1} Z\|_{\mathrm{HS}} \leq R$. Moreover,

$$d_{\mathrm{eff}}(\lambda) := \mathrm{Tr}((C + \lambda I)^{-1} C) \leq \lambda^{-1}\kappa^2. \tag{6.109}$$

So, we get

$$\begin{aligned}\mathbb{E}_x[\|\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \quad & \frac{\kappa(Q + \kappa R)}{\sqrt{\lambda}n} \times 10\log\frac{4}{\delta} \\ & + \frac{1}{\sqrt{n}} \times \sqrt{(\lambda^{-1}\kappa^2\|E^{1/2}\|_{\mathrm{HS}}^2 + \kappa^2 R)} \times 10\log\frac{4}{\delta} \\ & + \lambda^{\frac{1}{2}} \times R \times 10\log\frac{4}{\delta}\end{aligned}$$

Now, we define λ in order to minimize this bound, with

$$\lambda = \max\left(\frac{1}{n}, \frac{\|E^{1/2}\|_{\mathrm{HS}}}{\sqrt{n}}\right)$$

so we obtain

$$\begin{aligned}\mathbb{E}_x[\|\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq \quad & n^{-\frac{1}{4}} \times 10\log\frac{4}{\delta} \times \Bigg[(\kappa(Q + \kappa R))n^{-\frac{1}{4}} \\ & + \sqrt{\kappa^2\|E^{1/2}\|_{\mathrm{HS}} + \kappa^2 R^2 n^{-\frac{1}{2}}} \\ & + \left(n^{-\frac{1}{4}} + \|E^{1/2}\|_{\mathrm{HS}}^{\frac{1}{2}}\right) \times R\Bigg].\end{aligned}$$

We conclude

$$\mathbb{E}_x[\|\hat{h}(x) - h^*(x)\|_{\mathcal{Z}}^2]^{1/2} \leq C(p) n^{-\frac{1}{4}} \log\frac{4}{\delta} \tag{6.110}$$

with

$$C(p) = 10\left[(\kappa(Q+\kappa R))n^{-\frac{1}{4}} + \kappa\sqrt{\|E^{1/2}\|_{\mathrm{HS}} + R^2 n^{-\frac{1}{2}}} + \left(n^{-\frac{1}{4}} + \|E^{1/2}\|_{\mathrm{HS}}^{\frac{1}{2}}\right)R\right]$$
$$= 10\left[\mathcal{O}(n^{-\frac{1}{4}}) + (\kappa + R)\|E^{1/2}\|_{\mathrm{HS}}^{\frac{1}{2}}\right].$$

■

Theorem 6.11 (Concentration inequality on the operator norm, Tropp (2012)(Theorem 7.3.2))**.** *Let ξ_i be independent copies of the random variable ξ with values in the space of bounded operators over a Hilbert space $\mathcal{H}$ such that $\mathbb{E}[\xi] = 0$. Let there be $R > 0$ such that $\|\xi\|_\infty \leq T$. Define $\sigma^2 = \max(\|\mathbb{E}[\xi\xi^*]\|_\infty, \|\mathbb{E}[\xi^*\xi]\|_\infty)$, and $d = \mathrm{Tr}(\mathbb{E}[\xi^*\xi] + \mathbb{E}[\xi\xi^*])/\sigma^2$. Then, if $\delta \in [0,1]$, with probability at least $1-\delta$*

$$\left\|\frac{1}{n}\sum_{i=1}^{n}\xi_i\right\|_\infty \leq \sqrt{\frac{2\eta\sigma^2}{n}} + \frac{2T\eta}{3n} \tag{6.111}$$

where $\eta = \log(\frac{4d}{\delta})$.

Proof This theorem is a restatement of Theorem 7.3.2 of (Tropp, 2012) generalized to the separable Hilbert space case by means of the technique in Section 3.2 of (Minsker, 2017). ■

Lemma 6.12 (Bound $\|H_n - H\|_\infty$)**.** *With probability at least $1-2\delta$ it is*

$$\|H_n - H\|_\infty \leq \frac{4\log\frac{2}{\delta}}{\lambda_1\sqrt{n}}(Q\kappa + \kappa^2\|h_\psi^*\|_{\mathcal{H}}) + \|h_\psi^*\|_{\mathcal{H}}$$

Proof In order to bound $\|H_n - H\|_\infty$ we do the following decomposition in three terms, and bound each term:

$$\|H_n - H\|_\infty = \|Z_n^* S_n (C_n + \lambda_1 I)^{-1} - Z^* S C^\dagger\|_\infty$$
$$\leq \underbrace{\|(Z_n^* S_n - Z^* S)(C_n + \lambda_1 I)^{-1}\|_\infty}_{(A)} + \underbrace{\|Z^* S((C_n + \lambda_1 I)^{-1} - (C + \lambda_1 I)^{-1})\|_\infty}_{(B)}$$
$$+ \underbrace{\|Z^* S((C + \lambda_1 I)^{-1} - C^\dagger)\|_\infty}_{(C)}$$

Bound (A). We have:

$$(A) = \|(Z_n^* S_n - Z^* S)(C_n + \lambda_1 I)^{-1}\|_\infty \leq \frac{1}{\lambda_1}\|Z_n^* S_n - Z^* S\|_{\mathrm{HS}}$$

From Ciliberto et al. (2016) (proof of lemma 18.), with probability $1-\delta$: $(A) \leq \frac{4Q\kappa\log\frac{2}{\delta}}{\lambda_1\sqrt{n}}$.

Bound (B). We have:

$$\begin{aligned}(B) &= \|Z^*S((C+\lambda_1 I)^{-1} - (C_n+\lambda_1 I)^{-1})\|_\infty \\ &= \|Z^*S((C+\lambda_1 I)^{-1}(C_n - C)(C_n+\lambda_1 I)^{-1})\|_\infty \\ &\leq \|Z^*S(C+\lambda_1 I)^{-1}\|_\infty \|(C_n - C)\|_\infty \|(C_n+\lambda_1 I)^{-1}\|_\infty \\ &\leq \frac{1}{\lambda_1}\|h_\psi^*\|_\mathcal{H}\|(C_n - C)\|_\infty\end{aligned}$$

where we used the fact that for two invertible operators A, B: $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$, and noting that $\|Z^*S(C+\lambda_1 I)^{-1}\|_\infty \leq \|Z^*S(C+\lambda_1 I)^{-1}\|_{\text{HS}} \leq \|H\|_{\text{HS}} = \|h_\psi^*\|_\mathcal{H}$. From Ciliberto et al. (2016), with probability $1-\delta$: $(B) \leq \frac{4\|h_\psi^*\|_\mathcal{H}\kappa^2 \log\frac{2}{\delta}}{\lambda_1\sqrt{n}}$.

Bound (C). We have:

$$\begin{aligned}(C) &= \|Z^*S((C+\lambda_1 I)^{-1} - C^\dagger)\|_\infty \\ &= \|HS^*S((C+\lambda_1 I)^{-1} - C^\dagger)\|_\infty \\ &= \|H(C(C+\lambda_1 I)^{-1} - I)\|_\infty \\ &= \lambda_1\|H(C+\lambda_1 I)^{-1}\|_\infty \\ &\leq \|h_\psi^*\|_\mathcal{H}\end{aligned}$$

We conclude by union bound, with probability at least $1-2\delta$:

$$\|H_n - H\|_\infty \leq \frac{4Q\kappa\log\frac{2}{\delta}}{\lambda_1\sqrt{n}} + \frac{4\|h_\psi^*\|_\mathcal{H}\kappa^2\log\frac{2}{\delta}}{\lambda_1\sqrt{n}} + \|h_\psi^*\|_\mathcal{H}$$

Notice that if we choose $\lambda_1 = (c_8\log^8(\frac{8}{\delta}))^{-\frac{1-\beta}{2}} n^{-\frac{\beta}{\beta+1}}$ as chosen in Lemma 6.7, we obtain

$$\|H_n - H\|_\infty \leq (4Q\kappa + R\kappa^2)\log\frac{2}{\delta} \times a \times n^{\frac{\beta}{1+\beta}-\frac{1}{2}} + R \tag{6.112}$$

with $a = c_8\log^8(\frac{8}{\delta})^{\frac{1}{2}}$, such that $\|H_n - H\|_\infty \leq 2R\log^9(\frac{8}{\delta})^{\frac{1}{2}}$ when $n \geq N$ with $N > 0$ a constant independent of δ.

■

Lemma 6.13 (Properties of Löwners's partial ordering $\preceq$)**.** *Let A, B be positive semidefinite linear operators on $\mathcal{Y}$ such that $A \preceq B$, and M a bounded linear operator on $\mathcal{Y}$, then*

1. *If A, B are random variables then $\mathbb{E}[A] \preceq \mathbb{E}[B]$.*
2. *$MAM^* \preceq MBM^*$.*

Proof

1) For any $u \in \mathcal{Y}$, we have $\langle u, \mathbb{E}[A]u\rangle_\mathcal{Y} = \mathbb{E}[\langle u, Au\rangle_\mathcal{Y}] \leq \mathbb{E}[\langle u, Bu\rangle_\mathcal{Y}] = \langle u, \mathbb{E}[B]u\rangle_\mathcal{Y}$.

2) From Lemma B.2 in Rudi et al. (2013).

■

6.2.7 About the Independence of $\phi(x)$ and ϵ

In this section, we discuss the assumption that the random variables $\phi(x)$ and ϵ are independent.

In this work, this assumption allows to obtain shorter and lighter derivations, and an easier reading of the proofs. Nevertheless, such assumption is not exploited by the proposed method, and similar results can be proven without this assumption. More precisely, one can prove bounds with the same dependencies in the parameters of the learning setting, leading to the same conclusions. We discuss how below.

How to obtain similar bounds without this assumption? The independence of $\phi(x)$ and ϵ allow simpler derivations when bounding expectations involving products of these two random variables using $\mathbb{E}[f(\phi(x))g(\epsilon)] = \mathbb{E}[f(\phi(x)) \times \mathbb{E}[g(\epsilon)]$. This is used multiple times from Equations (38) to (48) to prove the Lemma 6.4, and only there.

We carried out derivations below in order to bound the same quantities but we do not make use of the assumption. Then, we will check that the dependencies in the parameters of the learning setting are similar.

Sketch of the proof (Bound $\|(Z_n^* S_n - HC_n)(C + \lambda_2 I)^{-1/2}\|_\infty$ without the independence assumption). We define

$$\xi_i = \epsilon_i \otimes \phi(x_i)(C + \lambda_2 I)^{-1/2} \tag{6.113}$$

with $\epsilon_i = y_i - h^*(x_i)$. In this way,

$$\|(Z_n^* S_n - HC_n)(C + \lambda_2 I)^{-1/2}\|_\infty = \|\frac{1}{n}\sum_{i=1}^{n} \xi_i - \mathbb{E}[\xi]\|_\infty. \tag{6.114}$$

We aim at applying the Bernstein inequality given in Theorem 6.11 to the random linear operator $u := \xi - \mathbb{E}[\xi]$. So, we define

$$T := 4\kappa Q \lambda_2^{-1/2} \geq \|u\|_\infty, \tag{6.115}$$

$$\sigma^2 := \max(\|\mathbb{E}[uu^*]\|_\infty, \|\mathbb{E}[u^*u]\|_\infty), \tag{6.116}$$

$$d := \mathrm{Tr}(\mathbb{E}[u^*u] + \mathbb{E}[uu^*])/\sigma^2. \tag{6.117}$$

Note that $\|\epsilon\| \leq \|z\|_\mathcal{Z} + \|h^*(x)\|_\mathcal{Z} \leq 2Q$, and $\|\phi(x)\| \leq \kappa$. Then, we have

$$\mathbb{E}[uu^*] = \mathbb{E}[(\epsilon \otimes \phi(x) - \mathbb{E}[\epsilon \otimes \phi(x)])(C + \lambda_2 I)^{-1}(\epsilon \otimes \phi(x) - \mathbb{E}[\epsilon \otimes \phi(x)])^*] \tag{6.118}$$

$$\leq \lambda_2^{-1}\mathbb{E}[(\epsilon \otimes \phi(x) - \mathbb{E}[\epsilon \otimes \phi(x)])(\epsilon \otimes \phi(x) - \mathbb{E}[\epsilon \otimes \phi(x)])^*] \tag{6.119}$$

$$= \lambda_2^{-1}\mathbb{E}[(\epsilon \otimes \phi(x)(\epsilon \otimes \phi(x))^*] - \mathbb{E}[\epsilon \otimes \phi(x)]\mathbb{E}[\epsilon \otimes \phi(x)]^* \tag{6.120}$$

$$\leq \lambda_2^{-1}\mathbb{E}[\epsilon \otimes \epsilon \|\phi(x)\|^2] \tag{6.121}$$

$$\leq \lambda_2^{-1}\kappa^2\mathbb{E}[\epsilon \otimes \epsilon] = \lambda_2^{-1}\kappa^2 E \tag{6.122}$$

where $\preceq$ denotes the Löwner's partial ordering of positive semidefinite operators. We used properties of Löwner's partial ordering (cf. Lemma 6.13). So, we have

$$\|\mathbb{E}[uu^*]\|_\infty \leq \lambda_2^{-1}\kappa^2\|E\|_\infty. \tag{6.123}$$

Then, similarly, we have

$$\mathbb{E}[u^*u] = (C+\lambda_2 I)^{-1/2}\mathbb{E}[(\epsilon\otimes\phi(x) - \mathbb{E}[\epsilon\otimes\phi(x))^*(\epsilon\otimes\phi(x) - \mathbb{E}[\epsilon\otimes\phi(x))](C+\lambda_2 I)^{-1/2} \tag{6.124}$$

$$= (C+\lambda_2 I)^{-1/2}\Big(\mathbb{E}[\phi(x)\otimes\phi(x)\|\epsilon\|^2] - \mathbb{E}[\phi(x)\otimes\epsilon]\mathbb{E}[\phi(x)\otimes\epsilon]^*\Big)(C+\lambda_2 I)^{-1/2} \tag{6.125}$$

$$\leq (C+\lambda_2 I)^{-1/2}4Q^2C(C+\lambda_2 I)^{-1/2} \tag{6.126}$$

$$\leq 4Q^2 I_{\mathcal{Z}}. \tag{6.127}$$

So, we have

$$\|\mathbb{E}[u^*u]\|_\infty \leq 4Q^2. \tag{6.128}$$

Now, from previous derivations, if $\lambda_2 < \|C\|_\infty$, we also have

$$\operatorname{Tr}(\mathbb{E}[uu^*]) \leq \lambda_2^{-1}\operatorname{Tr}(E)\kappa^2, \tag{6.129}$$

$$\operatorname{Tr}(\mathbb{E}[u^*u]) \leq 4Q^2\lambda_2^{-1}\operatorname{Tr}(C), \tag{6.130}$$

$$\|\mathbb{E}[uu^*]\|_\infty \geq \frac{\|\operatorname{Var}(\epsilon\otimes\phi(x))\|_\infty}{2\|C\|_\infty}. \tag{6.131}$$

by defining $\operatorname{Var}(\epsilon\otimes\phi(x)) = \mathbb{E}[(\epsilon\otimes\phi(x) - \mathbb{E}[\epsilon\otimes\phi(x)])(\epsilon\otimes\phi(x) - \mathbb{E}[\epsilon\otimes\phi(x)])^*]$. So, we have

$$d \leq \frac{\operatorname{Tr}(\mathbb{E}[u^*u]) + \operatorname{Tr}(\mathbb{E}[uu^*])}{\|\mathbb{E}[uu^*]\|_\infty} \tag{6.132}$$

$$\leq \lambda_2^{-1}\frac{2(\operatorname{Tr}(E)\kappa^2 + 4Q^2\operatorname{Tr}(C))\|C\|_\infty}{\|\operatorname{Var}(\epsilon\otimes\phi(x))\|_\infty}. \tag{6.133}$$

Conclusion. Then, one can bound $\|(Z_n^*S_n - HC_n)(C+\lambda_2 I)^{-1/2}\|_\infty$ as in the proof of Lemma 6.4 by applying the Bernstein inequality given in Theorem 6.11.

The dependencies in the learning setting's parameters of the resulting bound will depend on the dependencies in the learning setting's parameters of the obtained bounds on $\|\mathbb{E}[u^*u]\|_\infty$, $\|\mathbb{E}[uu^*]\|_\infty$, and d.

Notice that the bounds on $\|\mathbb{E}[u^*u]\|_\infty$, $\|\mathbb{E}[uu^*]\|_\infty$ have the same dependencies in the learning setting's parameters than the ones obtained in Lemma 6.4 on $\|\mathbb{E}[\xi^*\xi]\|_\infty$, $\|\mathbb{E}[\xi\xi^*]\|_\infty$.

The bound on d obtained above without the independence assumption has poorer dependencies in the learning setting's parameters than the one obtained in Lemma 6.4. More precisely, d has poorer dependencies in t_1 and λ_2. Nevertheless, it remains polynomial dependencies in t_1^{-1} and λ_2^{-1}, such that the resulting $\eta = \log(\frac{4d}{\delta})$, in the proof of Lemma 6.5, has similar dependencies in the learning setting's parameters than the one obtained in Lemma 6.5.

We conclude that, without the independence assumption of $\phi(x)$ and ϵ, one can prove bounds similar to Theorem 4.7, namely with the same dependencies in the parameters of the learning setting.

In this section, we give an additional synthetic experiment (Section 6.2.8) that aims at discussing the difference between the output source condition (Assumption 4.3) and the standard source condition (Ciliberto et al., 2020). We also give additional details on the experiments for the sake of reproducibility (Sections 6.2.9, 6.2.10).

6.2.8 Difference Between Standard Source Condition and Assumption 4.3.

From Assumption 4.1 we have $M = HCH^*$. Hence, Assumption 4.3 measures the alignment between HCH^* and HH^*. Notice that it's a different assumption than requiring the alignment of C and H^*H (source condition). Indeed, in general strong Assumption 4.3 doesn't imply strong source condition. For instance, when H is finite rank (e.g. $H = z_0 \otimes h_0$ with $z_0 \in \mathcal{Y}, h_0 \in \mathcal{H}_x$), Assumption 4.3 is verified with $\beta = 0$ (best case), while the source condition can be arbitrarily bad (e.g. if $\langle h_0 | C^{-(1-v)} h_0 \rangle_{\mathcal{H}_x} = +\infty$ with $v > 0$, then the source condition can't be verified for $r \leq v$). Source condition is verified with $r = 1 - 2u$ by operators of the form $H = H_0 C^u$ with $H_0 \in \mathcal{Y} \otimes \mathcal{H}_x$, $\|H_0\|_{\text{HS}} < +\infty$, $u \in [0, \frac{1}{2}]$. Similarly, Assumption 4.3 is verified with $\beta = \frac{1}{2u+1}$ by operators of the form $H = (H_0 C H_0^*)^u H_0$ with $\|H_0\|_\infty < +\infty$, $u \in [0, +\infty[$.
We illustrate this empirically. For $d = 200$, $\mathcal{X} = \mathcal{H}_x = \mathcal{Z} = \mathbb{R}^d$, we choose $\mu_p(C) = \frac{1}{p^2}$ and draw randomly the eigenvector associated to each eigenvalue. We draw $H_0 \in \mathbb{R}^{d \times d}$ with independently drawn coefficients from the standard normal distribution. Notice that β and r can be measured as the increasing rates, when $t, \lambda \to 0$, in $t^{-\beta}$ and λ^{-r} of the quantities $\|(M+t)^{-\frac{1}{2}} H\|_\infty^2$ and $\|H(C+\lambda)^{-\frac{1}{2}}\|_\infty^2$. Hence, we compute and plot on Figure 6.1 $\|H(C+\lambda)^{-\frac{1}{2}}\|_\infty^2$ w.r.t λ (left), and $\|(M+t)^{-\frac{1}{2}} H\|_\infty^2$ w.r.t t (right), with $H = (H_0 C H_0^*)^\gamma H_0$ for various $\gamma \in [0, 1.5]$. We also plot in Figure 6.1 (right) the slopes $\beta = \frac{1}{2\gamma+1}$. Firstly, we see that Assumption 4.3 indeed improved when γ increases, while the source condition is low and does not change. Then, as explained $H = (H_0 C H_0^*)^\gamma H_0$ verifies Assumption 4.3 with at least $\beta = \frac{1}{2\gamma+1}$, but depending on H_0 it might be verified for $\beta \ll \frac{1}{2\gamma+1}$. Nonetheless, notice that with our generated H_0, $\beta = \frac{1}{2\gamma+1}$ are sharp for $H = (H_0 C H_0^*)^\gamma H_0$.

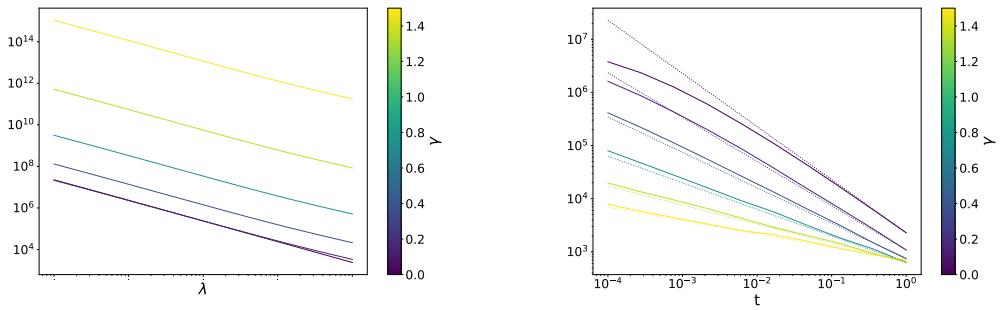

Figure 6.1: Source condition $\|H(C+\lambda)^{-\frac{1}{2}}\|_\infty^2$ w.r.t λ (left) and output source condition $\|(M+t)^{-\frac{1}{2}} H\|_\infty^2$ w.r.t t (right) in log-log scale for $H = (H_0 C H_0^*)^\gamma H_0$ and various $\gamma \in \{0, 0.1, 0.25, 0.5, 0.9, 1.5\}$.

6.2.9 Image Reconstruction

Link to downloadable data set. `web.stanford.edu/~hastie/StatLearnSparsity_files/DATA/zipcode.html`

SPEN USPS experiments' details. We used an implementation of SPEN in python with PyTorch by Philippe Beardsell and Chih-Chao Hsu (cf. github.com/philqc/deep-value-networks-pytorch). Small changes have been made. SPEN was trained using standard architecture from Belanger and McCallum (2016), that is a simple 2-hidden layers neural network for the feature network with equal layer size $n_h = 110$, and a single-hidden layer neural network for the structure learning network with size $n_s = 50$. The size of the two hidden layers $n_h \in [10, 30, 50, 70, 90, 110, 130]$ was selected during the pre-training of the feature network using 5 repeated random sub-sampling validation (80%/20%) selecting the best mean validation MSE (cf. Figure 6.2 for convergence of this phase). $n_s \in [5, 10, 20, 50, 70]$ was selected during the training phase of the SPEN network (training of the structure learning network plus the last layer of the feature network) doing approximate loss-augmented inference (cf. Figure 6.2 for inferences' convergences), and minimizing the SSVM loss, using 5 repeated random sub-sampling validation (80%/20%) selecting the best mean validation MSE (cf. Figure 6.2 for convergence of this phase).

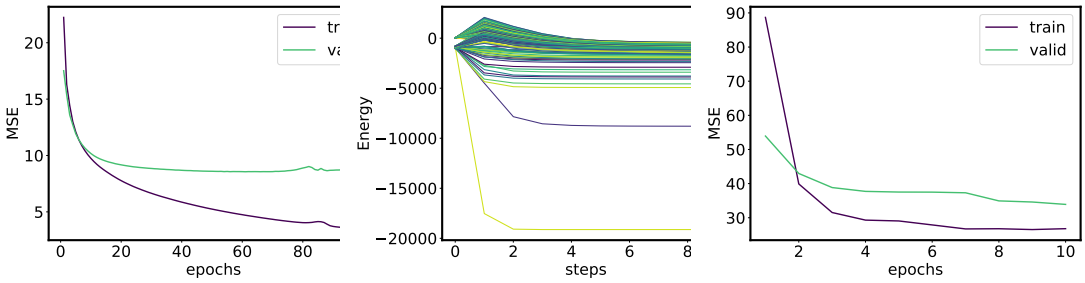

Figure 6.2: Left: Convergence of train/validation MSE when pre-training the feature network. / Center: approximate loss-augmented inferences' convergences. / Right: Convergence of train/validation SSVM loss when training the SPEN network.

6.2.10 Multi-label Classification

Link to downloadable data set `http://mulan.sourceforge.net/datasets-mlc.html`

6.3 Proofs and additional results of Chapter 5

In this section, we provides the proofs for the Theorem 1, and Corollary 1 and 2.

6.3.1 Definitions and notations

- $\preceq$ denotes the Loewner partial order for positive definite operators: $A \preceq B$ iff $\langle u, Au \rangle \leq \langle u, Bu \rangle$ for any u.
- We note $\tilde{\psi}(y) = k_y(y,.)$, $c_\psi = \sup_y \|\psi(y)\|$, $c_{\tilde{\psi}} = \sup_y \|\tilde{\psi}(y)\|$, $c_\chi = \sup_y \|\chi(y)\|$, $\kappa = \sup_x \|k_x(x,.)\|_{\mathcal{H}_x}$.

- We define the following Least-squares solutions $h^*_\psi(x) = \mathbb{E}_{y|x}[\psi(y)]$, $h^*_{\tilde{\psi}}(x) = \mathbb{E}_{y|x}[\tilde{\psi}(y)]$.
- We define the following ideal covariance operators: $C_x = \mathbb{E}[k_x(x,.)\otimes k_x(x,.)]$, $C_{\tilde{\psi}} = \mathbb{E}[\tilde{\psi}(y)\otimes\tilde{\psi}(y)]$, $C_\psi = \mathbb{E}[\psi(y)\otimes\psi(y)]$, $M_{\tilde{\psi}} = \mathbb{E}[h^*_{\tilde{\psi}}(x))\otimes h^*_{\tilde{\psi}}(x)]$, $E_{\tilde{\psi}} = \mathbb{E}[\epsilon_{\tilde{\psi}}\otimes\epsilon_{\tilde{\psi}}]$, $M_\psi = \mathbb{E}[h^*_\psi(x))\otimes h^*_\psi(x)]$, $E_\psi = \mathbb{E}[\epsilon_\psi\otimes\epsilon_\psi]$.
- We note $R_\psi = \|H_\psi\|_{\text{HS}}$, $R_{\tilde{\psi}} = \|H_{\tilde{\psi}}\|_{\text{HS}}$, $R_W = \|W\|_{\text{HS}}$.

Let's recall the following results giving a closed-form formula for the kernel ridge estimator.

Lemma 6.14 (KRR estimator)**.** *Let $\mathcal{Z}$ be a separable Hilbert space, let $\mathcal{H}$ be the RKHS induced by the operator-valued kernel $K(x,x') = k(x,x')I_{\mathcal{Z}}$. Then, the solution to the following regularized empirical risk minimization*

$$\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \|h(x_i) - z_i\|^2_{\mathcal{Z}} + \lambda\|h\|^2_{\mathcal{H}} \tag{6.134}$$

is given by

$$\hat{h}_z(x) := \sum_{i=1}^{n} \alpha_i(x) z_i \quad \textit{with} \quad \alpha(x) = (K_x + n\lambda)^{-1} k_x(x) \tag{6.135}$$

with $K_x = (k_x(x_i,x_j))_{i,j=1}^n \in \mathbb{R}^{n\times n}$, and $k_x(x) = (k_x(x,x_1),\dots,k_x(x,x_n)) \in \mathbb{R}^n$. Moreover, the following inequality holds

$$\hat{h}_z(x) = \hat{H}_z k_x(x,.) \quad \textit{with} \quad \hat{H}_z = \hat{C}_{zx}(\hat{C}_x + \lambda)^{-1} \tag{6.136}$$

with $\hat{C}_x = \frac{1}{n}\sum_{i=1}^n k_x(x_i,.)\otimes k_x(x_i,.)\otimes k_x(x,.)$, and $\hat{C}_{zx} = \frac{1}{n}\sum_{i=1}^n z_i\otimes k_x(x_i,.)$.

Proof Equation (6.136) holds from B.4 in Ciliberto et al. (2020). Equation (6.135) can be obtained from the Woodbury matrix identity and the Equation (6.136) (see for example Lemma 3 in Ciliberto et al. (2016)).

■

Now, we can make the following definitions and notations.

- We note $\hat{h}_{\tilde{\psi}} = \hat{h}_{\tilde{\psi}(y)}$, and $\hat{h}_\psi = \hat{h}_{\psi(y)}$.
- We define $\hat{W} = \hat{C}_{\psi,\tilde{\psi}}(\hat{C}_{\tilde{\psi}} + \mu I)^{-1}$ with $C_{\psi,\tilde{\psi}} = \frac{1}{m}\sum_{j=1}^m \psi(y)\otimes\tilde{\psi}(y)$, such that $\hat{W}\tilde{\psi}(y) = \sum_{j=1}^m \beta_j(y)\psi(y_j)$ with $\beta(y) = (K_y + m\mu)^{-1}k_y(y)$ with $K_y = (k_y(y_i,y_j))_{i,j=1}^m \in \mathbb{R}^{m\times m}$, and $k_y(y) = (k_y(y,y_1),\dots,k_y(y,y_m)) \in \mathbb{R}^m$.

We recall the comparison inequality, and a kernel ridge excess-risk bounds from Ciliberto et al. (2020) (Theorem 3 and Theorem B.8).

Lemma 6.15 (Comparison inequality). *Let $f^* : \mathcal{X} \to \mathcal{Y}$ be the measurable function minimizing the expected risk $\mathcal{R}(f) = \mathbb{E}[\Delta(f(x), y]$. Then the following inequality holds*

$$\mathcal{R}_\Delta(\hat{f}) - \mathcal{R}_\Delta(f^*) \leq c_\chi \, \mathbb{E}[\|\hat{h}(x) - h^*_\psi(x)\|^2_{\mathcal{H}_y}]^{1/2}, \tag{6.137}$$

with $\hat{f}(x) = \arg\min_{\hat{y} \in \mathcal{Y}} \sum_{i=1}^n \sum_{j=1}^m \alpha_i(x)\beta_j(y_i)\Delta(\hat{y}, y_j)$, and $\hat{h}(x) = \sum_{i=1}^n \sum_{j=1}^m \alpha_i(x)\beta_j(y_i)\psi(y_j)$.

Lemma 6.16 (Kernel ridge excess-risk bounds). *Let $\hat{h}_z = \hat{H}_z k_x(x,.)$ be the kernel ridge estimator from $\mathcal{X}$ to $\mathcal{Z}$, with regularization parameter $\mu > 0$ and training data $(x_i, z_i)_{i=1}^n$, as defined in Lemma 6.14. Let be $E = E[\epsilon \otimes \epsilon]$ with $\epsilon = z - h^*_z(x)$. In the attainable case, i.e. if $h^*_z(x) := \mathbb{E}_{z|x}[z] = H_z k_x(x,.)$ with $\|H_z\|_{\text{HS}} < +\infty$, then with probability $1 - \delta$, we have*

$$\begin{aligned}
\mathbb{E}[\|\hat{h}_z(x) - h^*_z(x)\|^2_{\mathcal{H}_y}]^{1/2} \leq\ & 8\log(4/\delta)(\sup_y \|z\| + \kappa\|H_z\|_{\text{HS}})\mu^{-1/2}n^{-1} && (6.138)\\
& + 8\kappa\|E^{1/2}\|_{\text{HS}}\mu^{-1/2}n^{-1/2}\log(4/\delta) && (6.139)\\
& + \kappa\mu^{1/2}n^{-1/2}\log(4/\delta)\|H_z(C_x + \mu I)^{-1}C_x^{1/2}\|_{\text{HS}} && (6.140)\\
& + 10\mu\|H_z(C_x + \mu I)^{-1}C_x^{1/2}\|_{\text{HS}}. && (6.141)
\end{aligned}$$

We show an excess-risk bounds for the kernel ridge estimator, in the noiseless case $E = 0$.

Lemma 6.17 (Kernel ridge excess-risk bounds in the noiseless regime). *Let $\hat{h}_z = \hat{H}_z k_x(x,.)$ be the kernel ridge estimator from $\mathcal{X}$ to $\mathcal{Z}$, with regularization parameter $\mu > 0$ and training data $(x_i, z_i)_{i=1}^n$, as defined in Lemma 6.14. Let be $E = E[\epsilon \otimes \epsilon]$ with $\epsilon = z - h^*_z(x)$. In the attainable case, with $h^*_z(x) := H_z k_x(x,.)$ with $\|H_z\|_{\text{HS}} < +\infty$, and the noiseless case $E = 0$, $\mu \geq \frac{9\kappa^2}{n}\log(\frac{n}{\delta})$, then with probability $1 - \delta$, we have*

$$\mathbb{E}[\|\hat{h}_z(x) - h^*_z(x)\|^2_{\mathcal{H}_y}]^{1/2} \leq 4\mu\|H_z(C + \mu I)^{-1/2}\|_{\text{HS}}. \tag{6.142}$$

Proof Performing similar derivations than in the proof of Theorem B.5 in Ciliberto et al. (2020), and then applying Lemma B.6, we obtain directly with probability at least $1 - \delta$, if $\mu \geq \frac{9\kappa^2}{n}\log(\frac{n}{\delta})$, then

$$\mathbb{E}[\|\hat{h}_z(x) - h^*_z(x)\|^2_{\mathcal{H}_y}]^{1/2} \leq 4\mu\|H_z(C + \mu I)^{-1/2}\|_{\text{HS}}. \tag{6.143}$$

■

6.3.2 Proof of Theorem 1 (Learning bounds)

In this section, we prove the main theorem of this chapter, which is an excess-risk bounds for $\hat{h}$ that carrefully monitors the three following parameter: n, λ, μ. Then, it will allow us to show that having non zero μ (output regularization parameter) can leads to better constants, exploiting a regularized output embedding $\psi_\beta(y)$ that leads to a greater variance reduction than a bias increase.

To this end, we start by proving the two following lemmas.

Lemma 6.18. *Let $\delta \in [0,1]$. If $\mu \geq \frac{9c_{\tilde{\psi}}^2}{m}\log(\frac{m}{\delta})$, then the following holds with probability at least $1-\delta$*

$$\mathbb{E}[\|(\hat{W}-W)h^*_{\tilde{\psi}}(x))\|^2_{\mathcal{H}_y}]^{1/2} \leq 4c_1^{1/2}\mu^{\gamma/2}R_W. \tag{6.144}$$

Proof Notice that,

$$E_{\tilde{\psi}} \preceq C_{\tilde{\psi}} \tag{6.145}$$

because $\tilde{\psi}(y) = h^*(x) + \epsilon$, and $h^*(x) = \mathbb{E}_{y|x}[\tilde{\psi}(y)]$.

Using $M_{\tilde{\psi}} \preceq c_1 E^{\gamma}_{\tilde{\psi}}$, and $E_{\tilde{\psi}} \preceq C_{\tilde{\psi}}$, we have

$$\mathbb{E}[\|(\hat{W}-W)h^*_{\tilde{\psi}}(x))\|^2_{\mathcal{H}_y}]^{1/2} = \|(\hat{W}-W)M^{1/2}_{\tilde{\psi}}\|_{\text{HS}} \tag{6.146}$$

$$\leq c_1^{1/2}\|(\hat{W}-W)C^{\gamma/2}_{\tilde{\psi}}\|_{\text{HS}} \tag{6.147}$$

Now, notice that $\hat{W}$ is the ridge estimate for the noiseless problem $\tilde{\psi}(y) \to \psi(y)$. So, we can use the kernel ridge bound Lemma 6.17, and we get

$$\|(\hat{W}-W)C^{\gamma/2}_{\tilde{\psi}}\|_{\text{HS}} \leq 4\mu\|W(C_{\tilde{\psi}}+\mu I)^{-1/2}C^{(\gamma-1)/2}_{\tilde{\psi}}\|_{\text{HS}} \tag{6.148}$$

$$\leq 4\mu^{\gamma/2}\|W\|_{\text{HS}}. \tag{6.149}$$

■

Lemma 6.19. *Let $\delta \in [0,1]$, and $P_\mu = C_{\tilde{\psi}}(C_{\tilde{\psi}}+\mu I)^{-1}$. When $\lambda = \max(\|WP_\mu E^{1/2}_{\tilde{\psi}}\|_{\text{HS}} n^{-1/2}, n^{-1})$, and $\frac{9c_{\tilde{\psi}}^2}{m}\log(\frac{m}{\delta}) \geq \mu \leq \|C_{\tilde{\psi}}\|_\infty$, then the following holds with probability at least $1-\delta$*

$$\mathbb{E}[\|\hat{W}\hat{h}_{\tilde{\psi}}(x) - \hat{W}h^*_{\tilde{\psi}}(x))\|^2_{\mathcal{H}_y}]^{1/2} \leq c\log^2(4/\delta) \times \left[n^{-1/4}\|WP_\mu E^{1/2}_{\tilde{\psi}}\|^{1/2}_{\text{HS}} + \mu^{\gamma/2} + n^{-1/2}\right] \tag{6.150}$$

with $c_3 = 20(c_\psi + 4c_{\tilde{\psi}} + 5\kappa(R_\psi + R_{\tilde{\psi}}) + 3R_W(1+4\kappa+c_1^{1/2}) + \kappa + R_\psi + c^{-1}_{\tilde{\psi}})$.

Proof From the attainability assumption, we have

$$h^*_{\tilde{\psi}(x)} = H_{\tilde{\psi}}k_x(x,.) \quad \text{with} \quad \|H_{\tilde{\psi}}\|_{\text{HS}} < +\infty. \tag{6.151}$$

Then, using the kernel ridge bound from Lemma 6.16, we have

$$\mathbb{E}[\|\hat{W}\hat{h}_{\tilde{\psi}}(x) - \hat{W}h^*_{\tilde{\psi}}(x))\|^2_{\mathcal{H}_y}]^{1/2} \leq 8\log(4/\delta)(\sup_y\|\hat{W}\tilde{\psi}(y)\| + \kappa\|\hat{W}H_{\tilde{\psi}}\|_{\text{HS}})\lambda^{-1/2}n^{-1} \tag{6.152}$$

$$+ 8\kappa\|\hat{W}E^{1/2}_{\tilde{\psi}}\|_{\text{HS}}\lambda^{-1/2}n^{-1/2}\log(4/\delta) \tag{6.153}$$

$$+ \kappa\lambda^{1/2}n^{-1/2}\log(4/\delta)\|\hat{W}H_{\tilde{\psi}}(C_x+\lambda I)^{-1}C_x^{1/2}\|_{\text{HS}} \tag{6.154}$$

$$+ 10\lambda\|\hat{W}H_{\tilde{\psi}}(C_{\tilde{\psi}}+\lambda I)^{-1}C_x^{1/2}\|_{\text{HS}}. \tag{6.155}$$

Let's bound each of the four terms depending on $\hat{W}$.

1. Bound $\|\hat{W}\tilde{\psi}(y)\|$. First,

$$\|\hat{W}\tilde{\psi}(y)\| \leq \|W\tilde{\psi}(y)\| + \|\hat{W} - W\|_\infty \|\tilde{\psi}(y)\| \tag{6.156}$$
$$= \|\psi(y)\| + \|\hat{W} - W\|_\infty \|\tilde{\psi}(y)\|. \tag{6.157}$$

Then, from Lemma 15 in Brogat-Motte et al. (2022), with probability at least $1 - \delta$:

$$\|\hat{W} - W\|_\infty \leq 4\log(2/\delta)\mu^{-1}m^{-1/2}(\sup_y \|\psi(y)\|\kappa + c_{\tilde{\psi}}^2\|W\|_\infty) + \|W\|_\infty \tag{6.158}$$

Hence, we conclude

$$\sup_y \|\hat{W}\tilde{\psi}(y)\| \leq c_\psi + c_{\tilde{\psi}}\left(4\log(2/\delta)\mu^{-1}m^{-1/2}(c_\psi\kappa + c_{\tilde{\psi}}^2 R_W) + R_W\right). \tag{6.159}$$

2. Bound $\|\hat{W}H_{\tilde{\psi}}\|_{\text{HS}}$. Similarly as before,

$$\|\hat{W}H_{\tilde{\psi}}\|_{\text{HS}} \leq \|WH_{\tilde{\psi}}\|_{\text{HS}} + \|\hat{W} - W\|_\infty \|H_{\tilde{\psi}}\|_{\text{HS}} \tag{6.160}$$
$$= \|H_\psi\|_{\text{HS}} + \|\hat{W} - W\|_\infty \|H_{\tilde{\psi}}\|_{\text{HS}}. \tag{6.161}$$

Then, using the same bound than in the previous paragraph, we get

$$\|\hat{W}H_{\tilde{\psi}}\|_{\text{HS}} \leq R_\psi + \left(4\log(2/\delta)\mu^{-1}m^{-1/2}(c_\psi\kappa + c_{\tilde{\psi}}^2 R_W) + R_W\right)R_{\tilde{\psi}}. \tag{6.162}$$

3. Bound $\|\hat{W}H_{\tilde{\psi}}(C_x + \lambda I)^{-1}C_x^{1/2}\|_{\text{HS}}$. First,

$$\|\hat{W}H_{\tilde{\psi}}(C_x + \lambda I)^{-1}C_x^{1/2}\|_{\text{HS}} \leq \|(\hat{W} - W)H_{\tilde{\psi}}(C_x + \lambda I)^{-1}C_x^{1/2}\|_{\text{HS}} + \|WH_{\tilde{\psi}}(C_x + \lambda I)^{-1}C_x^{1/2}\|_{\text{HS}}. \tag{6.163}$$

Then,

$$\|WH_{\tilde{\psi}}(C_x + \lambda I)^{-1}C_x^{1/2}\|_{\text{HS}} \leq \|H_\psi\|_{\text{HS}} \times \lambda^{-1/2}. \tag{6.164}$$

Moreover, as previously

$$\|(\hat{W} - W)H_{\tilde{\psi}}(C_x + \lambda I)^{-1}C_x^{1/2}\|_{\text{HS}} \leq \|(\hat{W} - W)H_{\tilde{\psi}}C_x^{1/2}\|_{\text{HS}} \times \lambda^{-1} \tag{6.165}$$
$$= \|(\hat{W} - W)M_{\tilde{\psi}}^{1/2}\|_{\text{HS}} \times \lambda^{-1} \tag{6.166}$$
$$\leq c_1^{1/2}\|(\hat{W} - W)C_{\tilde{\psi}}^{\gamma/2}\|_{\text{HS}} \times \lambda^{-1} \tag{6.167}$$

From Lemma 6.17 this term can be bounded with probability at least $1 - \delta$, as follows

$$\|(\hat{W} - W)C_{\tilde{\psi}}^{\gamma/2}\|_{\text{HS}} \leq 4\mu^{\gamma/2}\|W\|_{\text{HS}}. \tag{6.168}$$

To conclude, we sum up the two bounds on each term in Equation (6.163), and we obtain:

$$\|\hat{W}H_{\tilde{\psi}}(C_x + \lambda I)^{-1}C_x^{1/2}\|_{\text{HS}} \leq 4c_1^{1/2}\mu^{\gamma/2}R_W\lambda^{-1} + R_\psi\lambda^{-1/2}. \tag{6.169}$$

4. Bound $\|\hat{W}E_{\tilde{\psi}}^{1/2}\|_{\text{HS}}$. We note $P_\mu = C_{\tilde{\psi}}(C_{\tilde{\psi}} + \mu I)^{-1}$, and $W_\mu = WP_\mu$,

$$\|\hat{W}E_{\tilde{\psi}}^{1/2}\|_{\text{HS}} \leq \|(\hat{W} - WP_\mu)E_{\tilde{\psi}}^{1/2}\|_{\text{HS}} + \|WP_\mu E_{\tilde{\psi}}^{1/2}\|_{\text{HS}} \tag{6.170}$$

Then, using $E_{\tilde{\psi}} \preceq C_{\tilde{\psi}}$, we have

$$\|(\hat{W} - WP_\mu)E_{\tilde{\psi}}^{1/2}\|_{\text{HS}} \leq \|(\hat{W} - WP_\mu)C_{\tilde{\psi}}^{1/2}\|_{\text{HS}}. \tag{6.171}$$

Following similar proof than the one of Lemma 6.17, and using Lemma 3.6 in Rudi et al. (2013), this term can also be bounded as

$$\|(\hat{W} - W_\mu)C_{\tilde{\psi}}^{1/2}\|_{\text{HS}} \leq 4\mu^{\gamma/2}\|W\|_{\text{HS}}. \tag{6.172}$$

Conclusion We come back to Equation (6.155), summing up all the bounds obtained just above, and choosing $\lambda = \max(\|WP_\mu E_{\tilde{\psi}}^{1/2}\|_{\text{HS}} n^{-1/2}, n^{-1})$, and $\frac{9c_{\tilde{\psi}}^2}{m}\log(\frac{m}{\delta}) \leq \mu \leq \|C_{\tilde{\psi}}\|_\infty$, we get:

$$\|\hat{W}(\hat{H}_{\tilde{\psi}} - H_{\tilde{\psi}})S^*\|_{\text{HS}} \leq c\log^2(4/\delta) \times \left[n^{-1/4}\|WP_\mu E_{\tilde{\psi}}^{1/2}\|_{\text{HS}}^{1/2} + \mu^{\gamma/2} + n^{-1/2}\right] \tag{6.173}$$

with $c_3 = 20(c_\psi + 4c_{\tilde{\psi}} + 5\kappa(R_\psi + R_{\tilde{\psi}}) + 3R_W(1 + 4\kappa + c_1^{1/2}) + \kappa + R_\psi + c_{\tilde{\psi}}^{-1})$.

■

We can now state and prove the theorem.

Theorem 6.20 (Learning bounds)**.** *Let $\hat{h}(x) = \sum_{i=1}^n \sum_{j=1}^m \alpha_i(x)\beta_j(y_i)\psi(y_j)$, with $\alpha(x) = (K_x + m\mu)^{-1}k_x(x)$ with $K_x = (k_x(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n\times n}$, and $k_x(x) = (k_x(x, x_i))_{i=1}^n \in \mathbb{R}^n$, and $\beta(y) = (K_y + m\mu)^{-1}k_y(y)$ with $K_y = (k_y(y_i, y_j))_{i,j=1}^m \in \mathbb{R}^{m\times m}$, and $k_y(y) = (k_y(y, y_i))_{i=1}^m \in \mathbb{R}^m$. Using the λ defined in the proof, if $\mu \geq \frac{9c_{\tilde{\psi}}^2}{m}\log(\frac{m}{\delta})$, then with probability at least $1 - \delta$*

$$\mathcal{R}_\Delta(\hat{f}) - \mathcal{R}_\Delta(f^*) \leq c\log^2(4/\delta) \times \left(n^{-1/4}\|WP_\mu E_{\tilde{\psi}}^{1/2}\|_{\text{HS}}^{1/2} + \mu^{\gamma/2} + n^{-1/2}\right) \tag{6.174}$$

$c_4 = 20c_\chi(c_\psi + 4c_{\tilde{\psi}} + 5\kappa(R_\psi + R_{\tilde{\psi}}) + 4R_W(1 + 4\kappa + c_1^{1/2}) + \kappa + R_\psi + c_{\tilde{\psi}}^{-1})$.

Proof First, we use the Lemma 2 6.15 to bound the structured excess-risk by the Least-squares excess-risk. Then, we decompose the Least-squares excess-risk as follows.

Least-squares risk decomposition. By definition of $\hat{h}_{\tilde{\psi}}$, $\hat{h}_\psi$, and $\hat{W}$, we have

$$\hat{h}_\psi = \hat{W}\hat{h}_{\tilde{\psi}}. \tag{6.175}$$

From Assumption 3, we have

$$\psi(y) = W\tilde{\psi}(y) \quad \text{with} \quad \|W\|_{\text{HS}} < +\infty. \tag{6.176}$$

So, we have

$$\mathbb{E}[\|\hat{h}(x) - h^*_{\psi}(x)\|^2_{\mathcal{H}_y}] = \mathbb{E}[\|\hat{W}\hat{h}_{\tilde{\psi}}(x) - Wh^*_{\tilde{\psi}}(x))\|^2_{\mathcal{H}_y}] \tag{6.177}$$

$$\leq \underbrace{\mathbb{E}[\|\hat{W}\hat{h}_{\tilde{\psi}}(x) - \hat{W}h^*_{\tilde{\psi}}(x))\|^2_{\mathcal{H}_y}]}_{(1)} + \underbrace{\mathbb{E}[\|(\hat{W} - W)h^*_{\tilde{\psi}}(x))\|^2_{\mathcal{H}_y}]}_{(2)}. \tag{6.178}$$

Bound (1) and (2). We bound the two terms by applying Lemma 6.19 and Lemma 6.18.

Conclusion. We conclude the proof by summing up the two bounds on (1) and (2). We obtain the same bound than for (1):

$$\mathbb{E}[\|\hat{h}(x) - h^*_{\psi}(x)\|^2_{\mathcal{H}_y}]^{1/2} \leq c\log^2(4/\delta) \times \left[n^{-1/4}\|WP_{\mu}E^{1/2}_{\tilde{\psi}}\|^{1/2}_{\mathrm{HS}} + \mu^{\gamma/2} + n^{-1/2} \right] \tag{6.179}$$

but with the slightly modified constant $c_4 = 20(c_{\psi} + 4c_{\tilde{\psi}} + 5\kappa(R_{\psi} + R_{\tilde{\psi}}) + 4R_W(1 + 4\kappa + c_1^{1/2}) + \kappa + R_{\psi} + c_{\tilde{\psi}}^{-1})$.

■

6.3.3 Proof of Corollary 3 (Computational gain)

Here, we show that one can use a very reduce number of anchors $m \ll n$ and obtaining the same statistical guarantees, but a significantly improved computational complexity of the pre-image step.

Corollary 3 (Computational gain)**.** *Taking* $\mu = \frac{9c^2_{\tilde{\psi}}}{m}\log(\frac{m}{\delta})$*, under the same assumptions than in Theorem 6.20, as soon as*

$$\frac{m}{\log(m)} \gtrsim n^{\frac{1}{2\gamma}} \tag{6.180}$$

then we have with probability at least $1 - \delta$

$$\mathcal{R}_{\Delta}(\hat{f}) - \mathcal{R}_{\Delta}(f^*) \lesssim n^{-1/4} \tag{6.181}$$

Proof Notice that $\|WP_{\mu}E^{1/2}_{\tilde{\psi}}\|^{1/2}_{\mathrm{HS}} = \|WC_{\tilde{\psi}}(C_{\tilde{\psi}}+\mu I)^{-1}E^{1/2}_{\tilde{\psi}}\|^{1/2}_{\mathrm{HS}} \leq \|WE_{\tilde{\psi}}(E_{\tilde{\psi}}+\mu I)^{-1}E^{1/2}_{\tilde{\psi}}\|^{1/2}_{\mathrm{HS}} \leq \|WE^{1/2}_{\tilde{\psi}}\|_{\mathrm{HS}}$ using $C_{\tilde{\psi}} \preceq E_{\tilde{\psi}}$. Then, using the Theorem 6.20, with $\mu = \frac{9c^2_{\tilde{\psi}}}{m}\log(\frac{m}{\delta})$ and , we obtain the desired result.

■

That is, we proved the same learning rate $n^{-1/4}$ than the kernel ridge estimator using n anchors (all the training data) for the proposed method that only requires a reduced numbers of anchors $\mathcal{O}(n^{\frac{1}{2\gamma}})$ (negleting the log term). This depends on the parameter γ, when γ increases the number of required anchors decreases. For instance, if the assumption holds with $\gamma = 1$, $\mathcal{O}(n^{\frac{1}{2}}) \ll n$ is required for obtaining the learning rate $n^{-1/4}$.

6.3.4 Proof of Corollary 4 (Statistical gain)

In this section we derive the Corollary 4 which aims at showing from Theorem 6.20, that the excess-risk bounds with loss regularization can be significantly smaller than without regularization under the same assumptions.

Corollary 4 (Statistical gain)**.** *Taking* $\mu = c_5^{-(1-\tau)^{-1}} n^{-1/\gamma} \geq \frac{9c_{\tilde{\psi}}^2}{m} \log(\frac{m}{\delta})$*, under the same assumptions than in Theorem 6.20, we have with probability at least* $1-\delta$

$$\mathcal{R}_\Delta(\hat{f}) - \mathcal{R}_\Delta(f^*) \leq c \log^2(4/\delta) \times n^{-1/4} \left(\|E_{\tilde{\psi}}^{1/2}\|_{\mathrm{HS}}^{1/2} (1 - n^{-(1-\tau)/\gamma})^{1/4} + (1 + c_5^{-\gamma/2(1-\tau)}) n^{-1/4} \right) \tag{6.182}$$

with c *the constant defined in Theorem 6.20, and* $c_5 = c_2 \operatorname{Tr}(E_{\tilde{\psi}})^{-1}(1+c_1)^\tau$.

Proof From $h_{\tilde{\psi}}^*(x) = \mathbb{E}_{y|x}[\tilde{\psi}(y)]$, $\epsilon = \tilde{\psi}(y) - h^*(x)$, we have $\mathbb{E}[h^*(x) \otimes \epsilon] = \mathbb{E}_x[\mathbb{E}_{y|x}[h^*(x) \otimes \epsilon]] = \mathbb{E}_x[h^*(x) \otimes \mathbb{E}_{y|x}[\epsilon]] = 0$, which gives

$$C_{\tilde{\psi}} = M_{\tilde{\psi}} + E_{\tilde{\psi}}. \tag{6.183}$$

We have $\|W P_\mu E_{\tilde{\psi}}^{1/2}\|_{\mathrm{HS}} \leq \|W\|_\infty \|C_{\tilde{\psi}}(C_{\tilde{\psi}} + \mu I)^{-1} E_{\tilde{\psi}}^{1/2}\|_{\mathrm{HS}}$. Then, using $M_{\tilde{\psi}} \preceq c_1 E_{\tilde{\psi}}^\gamma$, we have

$$\begin{aligned}
\|P_\mu E_{\tilde{\psi}}^{1/2}\|_{\mathrm{HS}}^2 &= \operatorname{Tr}(P_\mu^2 E_{\tilde{\psi}}) && (6.184)\\
&= \operatorname{Tr}((M_{\tilde{\psi}} + E_{\tilde{\psi}})^2 (M_{\tilde{\psi}} + E_{\tilde{\psi}} + \mu I)^{-2} E_{\tilde{\psi}}) && (6.185)\\
&\leq \operatorname{Tr}(((1+c_1)E_{\tilde{\psi}})((1+c_1)E_{\tilde{\psi}} + \mu I)^{-1} E_{\tilde{\psi}}) && (6.186)\\
&= \operatorname{Tr}(E_{\tilde{\psi}}(E_{\tilde{\psi}} + \mu(1+c_1)^{-1} I)^{-1} E_{\tilde{\psi}}). && (6.187)
\end{aligned}$$

But for any $\mu > 0$, denoting e_k the k-th top eigenvalue of $E_{\tilde{\psi}}$, we have

$$\begin{aligned}
\operatorname{Tr}(E_{\tilde{\psi}}(E_{\tilde{\psi}} + \mu I)^{-1} E_{\tilde{\psi}}) &= \sum_{k=1}^{+\infty} \frac{e_k}{e_k + \mu} e_k && (6.188)\\
&= \sum_{k=1}^{+\infty} e_k - \mu \sum_{k=1}^{+\infty} \frac{e_k}{e_k + \mu} && (6.189)\\
&= \operatorname{Tr}(E_{\tilde{\psi}}) - \mu \operatorname{Tr}(E_{\tilde{\psi}}(E_{\tilde{\psi}} + \mu I)^{-1}) && (6.190)\\
&\leq \operatorname{Tr}(E_{\tilde{\psi}}) - c_2 \mu^{1-\tau}. && (6.191)
\end{aligned}$$

by using Assumption 5. So, we have

$$\operatorname{Tr}(E_{\tilde{\psi}}(E_{\tilde{\psi}} + \mu(1+c_1)^{-1} I)^{-1} E_{\tilde{\psi}}) \leq \operatorname{Tr}(E_{\tilde{\psi}})(1 - c_5 \mu^{1-\tau}), \tag{6.192}$$

defining $c_5 = c_2 \operatorname{Tr}(E_{\tilde{\psi}})^{-1}(1+c_1)^\tau$.

Therefore, we have by applying Theorem 6.20

$$\mathbb{E}[\|\hat{h}(x) - h_{\tilde{\psi}}^*(x)\|_{\mathcal{H}_y}^2]^{1/2} \leq c \log^2(4/\delta) \times \left(n^{-1/4} \|E_{\tilde{\psi}}^{1/2}\|_{\mathrm{HS}}^{1/2} (1 - c_5 \mu^{1-\tau})^{1/4} + \mu^{\gamma/2} + n^{-1/2} \right) \tag{6.193}$$

Now, using $\mu = c_5^{-(1-\tau)^{-1}} n^{-1/\gamma}$, we have

$$\mathbb{E}[\|\hat{h}(x) - h^*_{\psi}(x)\|^2_{\mathcal{H}_y}]^{1/2} \leq c \log^2(4/\delta) \times n^{-1/4} \left(\|E^{1/2}_{\tilde{\psi}}\|^{1/2}_{\mathrm{HS}} (1 - n^{-(1-\tau)/\gamma})^{1/4} + (1 + c_5^{-\gamma/2(1-\tau)}) n^{-1/4} \right) \tag{6.194}$$

■

That is, we have

$$\mathbb{E}[\|\hat{h}(x) - h^*_{\psi}(x)\|^2_{\mathcal{H}_y}]^{1/2} \lesssim \left(\|E^{1/2}_{\tilde{\psi}}\|^{1/2}_{\mathrm{HS}} (1 - n^{-(1-\tau)/\gamma})^{1/4} \right) n^{-1/4}. \tag{6.195}$$

Which should be compare to the kernel ridge bound under the same assumptions, which essentially is

$$\mathbb{E}[\|\hat{h}(x) - h^*_{\psi}(x)\|^2_{\mathcal{H}_y}]^{1/2} \lesssim \|W E^{1/2}_{\tilde{\psi}}\|^{1/2}_{\mathrm{HS}} n^{-1/4} \tag{6.196}$$

where $\lesssim$ is used here to simplify the discussion, by only keeping the dependencies in the dominant terms with respect to n, γ, τ, but the constants are explicited in the proofs.

In particular, for any $k \in \mathbb{N}^*$, one can obtain a constant k times smaller with the proposed estimator (kernel ridge with output regularization) than with the mere kernel ridge estimator, as soon as:

$$n \leq \left(1 - \frac{1}{k^4}\right)^{-\frac{\gamma}{1-\tau}}. \tag{6.197}$$

That is, one obtains a constant divided by k, when γ is enough big (concentrated signal), τ enough close to 1 (spreaded out noise), and n not too big to benefit from this regularization.

To put it in a nutshell, when $M_{\tilde{\psi}}$ has a fast eigenvalue decay rate, then Corrolary 3 shows that $\hat{y} \in \mathcal{Y}, x \to \mathbb{E}[y|x][\Delta(\hat{y}, y)]$ can be well approximated with few anchors. This leads to a significant computational gain when computing the pre-image. Moreover, if $E_{\tilde{\psi}}$ has a slow eigenvalue decay rate, then Corrolary 4 shows that this approximation substantially reduces the noise. This leads to a significant statistical gain.

7

Conclusion and Perspectives

7.1 Summary of the contributions

In this manuscript, we tackled the problem of dealing with high-dimensional and non-linear output spaces in supervised learning. We pointed out the importance of making maximum use of the available information on the output geometry to avoid suffering from the curse of dimensionality with respect to the output dimension. Consistent with this idea, we proposed statistically and computationally efficient structured prediction methods, supported by theoretical guarantees, and experimental assessments on both synthetic and real-world problems. In Chapter 3, we proposed a novel model for graph prediction by exploiting the natural geometry provided by the Gromov-Wasserstein metric on graph space: Gromov-Wasserstein barycentric models. The method is proposed in two versions: kernel-based and neural network based. Well-documented and user-friendly Python implementation of the method was made publicly available on GitHub. In Chapters 4 and 5, we proposed two least-squares estimators exploiting the structure provided by a kernel over the output space. We carried out a theoretical analysis of these estimators, showing, under output regularity assumptions, that the estimators allow reducing the output variance (or labeling noise), and the pre-image computational cost when used as surrogate regression estimators in structured prediction. These works highlight the principle of *output regularization* or *loss regularization* in structured prediction, which can be intuitively understood as the idea of tailoring the level of detail of predictions, depending on the quantity of training data, by imposing regularity conditions on the outputs of an estimator.

7.2 Perspectives

The work of Chapter 3 opens different perspectives.

- **Neural-network barycentric models to deal with complex output spaces.** The Gromov-Wasserstein barycentric model allows one to deal with output graph spaces. This principle could be generalized to other structured spaces by considering barycentric models induced by different metrics. As with the model proposed for graph prediction, such modeling would be intended to deal with the curse of dimensionality with respect to the output dimension.
- **Graph regularization.** From a theoretical point of view, it would be insightful to study the bias-variance trade-off with respect to the chosen size of the predicted graphs which can be understood as a resolution in the case of the Gromov-

Wasserstein metric. Similarly, it would be interesting to study the bias-variance trade-off with respect to the chosen number and size of the graph templates.

- **Large scale experiments.** From an experimental point of view, the proposed graph prediction approach could benefit from experimental tests on large scale real-world graph prediction problems, as for instance other molecular graph predictions problems than the metabolite identification one, or shape prediction problems (Pavlakos et al., 2018). This would lead to consider algorithmic improvements to deal with the computational complexity of the pre-image, which badly scales with the number of templates used. For instance, this could be done by leveraging recent advances in computational optimal transport for computing Gromov-Wasserstein barycenters.

- **Fine-tuning the GW distance.** The proposed model depends on the chosen geometry on the space of nodes' labels. This could be beneficially fine-tuned for specific graph spaces. For instance, in the case of molecular graph spaces, this would consist in well choosing the distances between atoms.

The work of Chapters 4 and 5 open various perspectives.

- **Direct estimation of the conditional expectations.** These works focused on exploiting information on the structure of the output space, given by a kernel over the output space $\mathcal{Y}$. One could consider leveraging regularity information on the maps $x, \hat{y} \to \mathbb{E}_{y|x}[\Delta(\hat{y}, y)]$, by means of a kernel over $\mathcal{X} \times \mathcal{Y}$.

- **Learning Unknown Losses.** A direct extension of the work in Chapter 5 is to consider settings where the target loss is unknown, namely when one is only given a finite number of loss evaluations.

- **Transfer learning.** A straightforward generalization of the method proposed in Chapter 5 would be to consider unsupervised output data y whose distribution differ from the supervised output data z, assuming one is provided two data sets $(x_i, z_i)_{i=1}^n$ and $(z_j, y_j)_{j=1}^m$. Allowing us to consider much more practical situations.

- **Beyond ridge regularization.** In Chapter 5, other loss regularization than the ridge one could be studied, for instance, other spectral regularizations (Rosasco et al., 2005; Bauer et al., 2007) or manifold regularizations (Zhu et al., 2003; Cabannes et al., 2021a; Belkin et al., 2005).

- **Dictionary approach.** The proposed method in Chapter 5 could be also used in a setting where $(y_j)_{j=1}^n$ is a well-chosen set of outputs instead of using random output data (Bouche et al., 2021).

- **Non-homogeneous source condition.** From our theoretical analysis it is natural to complete the set of assumptions by allowing non-homogeneous conditioning over dimensions of the output space.

Bibliography

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. *Advances in neural information processing systems*, 30, 2017.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 4(3):195–266, 2012. ISSN 1935-8237.

Theodore Wilbur Anderson. Estimating linear restrictions on regression coefficients for multivariate normal distributions. *The Annals of Mathematical Statistics*, pages 327–351, 1951.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, January 2015.

Gökhan Bakir, Thomas Hofmann, Alexander J Smola, Bernhard Schölkopf, and Ben Taskar. *Predicting structured data*. MIT Press, 2007.

Luca Baldassarre, Lorenzo Rosasco, Annalisa Barla, and Alessandro Verri. Multi-output learning via spectral filtering. *Machine Learning*, 87(3):259–301, 2012.

Amélie Barbe, Marc Sebban, Paulo Gonçalves, Pierre Borgnat, and Rémi Gribonval. Graph diffusion Wasserstein distances. In *ECML PKDD 2020-European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 1–16, 2020.

Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Frank Bauer, Sergei Pereverzev, and Lorenzo Rosasco. On regularization algorithms in learning theory. *Journal of Complexity*, 23(1):52–72, 2007.

David Belanger and Andrew McCallum. Structured prediction energy networks. In *International Conference on Machine Learning*, pages 983–992. PMLR, 2016.

David Belanger, Bishan Yang, and Andrew McCallum. End-to-end learning for structured prediction energy networks. In *International Conference on Machine Learning*, pages 429–439. PMLR, 2017.

Misha Belkin, Partha Niyogi, and Vikas Sindhwani. On manifold regularization. In *International Workshop on Artificial Intelligence and Statistics*, pages 17–24. PMLR, 2005.

J Frédéric Bonnans and Alexander Shapiro. Optimization problems with perturbations: A guided tour. *SIAM Review*, 40(2):228–264, 1998.

Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi. Wasserstein barycentric coordinates: Histogram regression using optimal transport. *ACM Trans. Graph.*, 35(4):71–1, 2016.

Dimitri Bouche, Marianne Clausel, François Roueff, and Florence d'Alché Buc. Nonlinear functional output regression: A dictionary approach. In *International Conference on Artificial Intelligence and Statistics*, pages 235–243. PMLR, 2021.

Luc Brogat-Motte, Rémi Flamary, Céline Brouard, Juho Rousu, and Florence d'Alché Buc. Learning to predict graphs with fused Gromov-Wasserstein barycenters. In *International Conference on Machine Learning*, pages 2321–2335. PMLR, 2022a.

Luc Brogat-Motte, Alessandro Rudi, Céline Brouard, Juho Rousu, and Florence d'Alché Buc. Vector-valued least-squares regression under output regularity assumptions. *Journal of Machine Learning Research*, 23(344):1–50, 2022b.

Céline Brouard, Florence d'Alché-Buc, and Marie Szafranski. Semi-supervised penalized output kernel regression for link prediction. In *Proceedings of the 28th International Conference on Machine Learning*, pages 593–600, 2011.

Céline Brouard, Huibin Shen, Kai Dührkop, Florence d'Alché Buc, Sebastian Böcker, and Juho Rousu. Fast metabolite identification with input output kernel regression. *Bioinformatics*, 32(12):i28–i36, 2016a.

Céline Brouard, Marie Szafranski, and Florence d'Alché Buc. Input output kernel regression: Supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research*, 17:np, 2016b.

Vivien Cabannes, Loucas Pillaud-Vivien, Francis Bach, and Alessandro Rudi. Overcoming the curse of dimensionality with Laplacian regularization in semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:30439–30451, 2021a.

Vivien A Cabannes, Francis Bach, and Alessandro Rudi. Fast rates for structured prediction. In *Conference on Learning Theory*, pages 823–865. PMLR, 2021b.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Andrea Caponnetto, Charles A Micchelli, Massimiliano Pontil, and Yiming Ying. Universal multi-task kernels. *The Journal of Machine Learning Research*, 9:1615–1646, 2008.

Claudio Carmeli, Ernesto De Vito, Alessandro Toigo, and Veronica Umanitá. Vector valued reproducing kernel Hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61, 2010.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):1–58, 2009.

Liang-Chieh Chen, Alexander Schwing, Alan Yuille, and Raquel Urtasun. Learning deep structured models. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1785–1794, Lille, France, 07–09 Jul 2015. PMLR.

Vincent S Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2580–2590, 2019.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

Andreas Christmann and Ingo Steinwart. Support vector machines. *Springer*, 2008.

Carlo Ciliberto, Lorenzo Rosasco, and Alessandro Rudi. A consistent regularization approach for structured prediction. *Advances in Neural Information Processing Systems*, 29:4412–4420, 2016.

Carlo Ciliberto, Alessandro Rudi, Lorenzo Rosasco, and Massimiliano Pontil. Consistent multitask learning with nonlinear output relations. *Advances in Neural Information Processing Systems*, 30, 2017.

Carlo Ciliberto, Francis Bach, and Alessandro Rudi. Localized structured prediction. *Advances in Neural Information Processing Systems*, 32, 2019.

Carlo Ciliberto, Lorenzo Rosasco, and Alessandro Rudi. A general framework for consistent structured prediction with implicit loss embeddings. *Journal of Machine Learning Research*, 21(98):1–67, 2020.

Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8, 2002.

Corinna Cortes, Mehryar Mohri, and Jason Weston. A general regression technique for learning transductions. In Luc De Raedt and Stefan Wrobel, editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 153–160. ACM, 2005.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2016.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26:2292–2300, 2013.

Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693. PMLR, 2014.

George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2011.

Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924, 2017.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

Kai Dührkop, Huibin Shen, Marvin Meusel, Juho Rousu, and Sebastian Böcker. Searching molecular structure databases with tandem mass spectra using CSI: FingerID. *Proceedings of the National Academy of Sciences*, 112(41):12580–12585, 2015.

Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

Rémi Flamary, Marco Cuturi, Nicolas Courty, and Alain Rakotomamonjy. Wasserstein discriminant analysis. *Machine Learning*, 107(12):1923–1945, 2018.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Rina Foygel, Michael Horrell, Mathias Drton, and John Lafferty. Nonparametric reduced rank regression. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a Wasserstein loss. *Advances in Neural Information Processing Systems*, 28, 2015.

Thomas Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58, 2003.

Pierre Geurts, Louis Wehenkel, and Florence d'Alché Buc. Kernelizing the output of tree-based methods. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 345–352, 2006.

Pierre Geurts, Louis Wehenkel, and Florence d'Alché Buc. Gradient boosting for kernelized output spaces. In *Proceedings of the 24th International Conference on Machine Learning*, pages 289–296, 2007.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In *International Conference on Machine Learning*, pages 2357–2365. PMLR, 2019.

Michael Gygli, Mohammad Norouzi, and Anelia Angelova. Deep value networks learn to evaluate and iteratively refine structured outputs. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1341–1351, 2017.

John A Hartigan and Manchek A Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1): 100–108, 1979.

Markus Heinonen, Huibin Shen, Nicola Zamboni, and Juho Rousu. Metabolite identification and molecular fingerprint prediction through machine learning. *Bioinformatics*, 28(18):2333–2341, 07 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts437. URL https://doi.org/10.1093/bioinformatics/bts437.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Heiko Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

Martin A Hoffmann, Louis-Félix Nothias, Marcus Ludwig, Markus Fleischauer, Emily C Gentry, Michael Witting, Pieter C Dorrestein, Kai Dührkop, and Sebastian Böcker. High-confidence structural annotation of metabolites absent from spectral libraries. *Nature Biotechnology*, pages 1–11, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning*, pages 137–142. Springer, 1998.

Hachem Kadri, Mohammad Ghavamzadeh, and Philippe Preux. A generalized kernel approach to structured output learning. In *International Conference on Machine Learning*, pages 471–479. PMLR, 2013.

Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(20):1–54, 2016.

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel text classification for automated tag suggestion. *ECML PKDD Discovery Challenge 2008*, page 75, 2008.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Anna Korba, Alexandre Garcia, and Florence d'Alché Buc. A structured prediction approach for label ranking. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966. PMLR, 2015.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pages 1945–1954. PMLR, 2017.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, 2001.

Pierre Laforgue, Alex Lambert, Luc Brogat-Motte, and Florence d'Alché Buc. Duality in RKHSs with infinite dimensional outputs: Application to robust losses. In *International Conference on Machine Learning*, pages 5598–5607. PMLR, 2020.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995, 1995.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 1(0), 2006.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

Christina Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Biocomputing 2002*, pages 564–575. World Scientific, 2001.

Sijin Li and Antoni B Chan. 3D human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

Néhémy Lim, Florence d'Alché Buc, Cédric Auliac, and George Michailidis. Operator-valued kernel-based vector autoregressive models for network inference. *Machine Learning*, 99(3):489–513, 2015.

Xi Victoria Lin, Sameer Singh, Luheng He, Ben Taskar, and Luke Zettlemoyer. Multi-label learning with posterior regularization. In *NIPS Workshop on Modern Machine Learning and Natural Language Processing*, 2014.

Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Central Science*, 3(10):1103–1113, 2017.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee International Conference on Data Mining*, pages 413–422. IEEE, 2008.

Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of Sinkhorn approximation for learning with Wasserstein distance. *Advances in Neural Information Processing Systems*, 31, 2018.

Giulia Luise, Dimitrios Stamos, Massimiliano Pontil, and Carlo Ciliberto. Leveraging low-rank relations between surrogate tasks in structured prediction. In *International Conference on Machine Learning*, pages 4193–4202. PMLR, 2019.

Helmut Lütkepohl. Vector autoregressive models. In *Handbook of Research Methods and Applications in Empirical Macroeconomics*. Edward Elgar Publishing, 2013.

Haggai Maron and Yaron Lipman. (probably) concave graph matching. *Advances in Neural Information Processing Systems*, 31, 2018.

Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, pages 3462–3471. PMLR, 2018.

Arthur Mensch, Mathieu Blondel, and Gabriel Peyré. Geometric losses for distributional learning. In *International Conference on Machine Learning*, pages 4516–4525. PMLR, 2019.

Charles A Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Computation*, 17(1):177–204, 2005.

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12), 2006.

Stanislav Minsker. On some extensions of Bernstein's inequality for self-adjoint operators. *Statistics & Probability Letters*, 127:111–119, 2017.

Youssef Mroueh, Tomaso Poggio, Lorenzo Rosasco, and Jean-Jeacques Slotine. Multiclass learning with simplex coding. *Advances in Neural Information Processing Systems*, 25, 2012.

Ashin Mukherjee and Ji Zhu. Reduced rank ridge regression and its kernel extensions. *Statistical Analysis and Data mining: the ASA Data Science Journal*, 4(6):612–622, 2011.

Facundo Mémoli. The Gromov–Wasserstein distance and the metric approach to oject matching. *Foundations of Computational Mathematics*, 11(4):417–487, 2011.

Dai Hai Nguyen, Canh Hao Nguyen, and Hiroshi Mamitsuka. SIMPLE: Sparse Interaction Model over Peaks of moLEcules for fast, interpretable metabolite identification from tandem mass spectra. *Bioinformatics*, 34(13):i323–i332, 2018.

Alex Nowak, Francis Bach, and Alessandro Rudi. Sharp analysis of learning with discrete losses. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1920–1929. PMLR, 2019.

Alex Nowak, Francis Bach, and Alessandro Rudi. Consistent structured prediction with max-min margin Markov networks. In *International Conference on Machine Learning*, pages 7381–7391. PMLR, 2020.

Alex Nowak-Vila, Francis Bach, and Alessandro Rudi. A general theory for structured prediction with smooth convex surrogates. *arXiv preprint arXiv:1902.01958*, 2019.

Sebastian Nowozin, Christoph H Lampert, et al. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3–4): 185–365, 2011.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9 (1):1–14, 2017.

Anton Osokin, Francis Bach, and Simon Lacoste-Julien. On structured prediction theory with calibrated convex surrogate losses. *Advances in Neural Information Processing Systems*, 30, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019.

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 459–468, 2018.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pages 2664–2672. PMLR, 2016.

Venkata K Pillutla, Vincent Roulet, Sham M Kakade, and Zaid Harchaoui. A smoother way to train structured prediction models. *Advances in Neural Information Processing Systems 31*, 2018.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

Guillaume Rabusseau and Hachem Kadri. Low-rank regression with tensor responses. *Advances in Neural Information Processing Systems*, 29:1867–1875, 2016.

Liva Ralaivola, Sanjay Joshua Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110, 2005.

Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

Bernardino Romera-Paredes, Hane Aung, Nadia Bianchi-Berthouze, and Massimiliano Pontil. Multilinear multitask learning. In *International Conference on Machine Learning*, pages 1444–1452. PMLR, 2013.

Lorenzo Rosasco, Ernesto De Vito, and Alessandro Verri. Spectral methods for regularization in learning theory. *DISI, Universita degli Studi di Genova, Italy, Technical Report DISI-TR-05-18*, 2005.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *NIPS*, pages 3215–3225, 2017.

Alessandro Rudi, Guillermo D. Cañas, and Lorenzo Rosasco. On the sample complexity of subspace learning. In *Advances in Neural Information Processing Systems*, pages 2067–2075, 2013.

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *NIPS*, pages 1657–1665, 2015.

Alessandro Rudi, Carlo Ciliberto, GianMaria Marconi, and Lorenzo Rosasco. Manifold structured prediction. *Advances in Neural Information Processing Systems*, 31, 2018.

David E Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. Backpropagation: The basic theory. *Backpropagation: Theory, Architectures and Applications*, pages 1–34, 1995.

Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association*, 2014.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*, pages 4470–4479. PMLR, 2018.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426. Springer, 2001.

Emma L Schymanski, Christoph Ruttkies, Martin Krauss, Céline Brouard, Tobias Kind, Kai Dührkop, Felicity Allen, Arpana Vaniya, Dries Verdegem, Sebastian Böcker, et al. Critical assessment of small molecule identification 2016: Automated methods. *Journal of Cheminformatics*, 9(1):1–21, 2017.

E. Senkene and Arkady Tempel'man. Hilbert spaces of operator-valued functions. *Mathematical Transactions of the Academy of Sciences of the Lithuanian SSR*, 13(4): 665–670, 1973.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

David Silver, Hado Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, et al. The predictron: End-to-end learning and planning. In *International Conference on Machine Learning*, pages 3191–3199. PMLR, 2017.

Florian Steinke, Matthias Hein, and Bernhard Schölkopf. Nonparametric regression between general Riemannian manifolds. *SIAM Journal on Imaging Sciences*, 3(3): 527–563, 2010.

Nicholas Sterge, Bharath Sriperumbudur, Lorenzo Rosasco, and Alessandro Rudi. Gain with no pain: Efficiency of kernel-PCA by Nyström sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 3642–3652. PMLR, 2020.

Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov networks. *Advances in Neural Information Processing Systems*, 16, 2003.

Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 896–903, 2005.

Joel A. Tropp. User-friendly tools for random matrices: An introduction. Technical report, California Institute of Technology Division of Engineering and Applied Science, 2012.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(9), 2005.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 1999.

Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. Last iterate convergence of SGD for least-squares in the interpolation regime. *Advances in Neural Information Processing Systems*, 34:21581–21591, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning (ICML)*, 2019.

Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Fused Gromov-Wasserstein distance for structured objects. *Algorithms*, 13(9):212, 2020.

Raja Velu and Gregory C Reinsel. *Multivariate reduced-rank regression: Theory and applications*, volume 136. Springer Science & Business Media, 2013.

Cédric Vincent-Cuaz, Titouan Vayer, Rémi Flamary, Marco Corneli, and Nicolas Courty. Online graph dictionary learning. In *International Conference on Machine Learning (ICML)*, 2021.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007.

G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in. Applied Mathematics*. Philadelphia: SIAM, 1990.

Mingxun Wang, Jeremy J Carver, Vanessa V Phelan, Laura M Sanchez, Neha Garg, Yao Peng, Don Duy Nguyen, Jeramie Watrous, Clifford A Kapono, Tal Luzzatto-Knaan, et al. Sharing and community curation of mass spectrometry data with global natural products social molecular networking. *Nature biotechnology*, 34(8): 828–837, 2016.

Jason Weston, Olivier Chapelle, Vladimir Vapnik, André Elisseeff, and Bernhard Schölkopf. Kernel dependency estimation. In *Advances in Neural Information Processing Systems*, pages 897–904, 2003.

David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable Gromov-Wasserstein learning for graph partitioning and matching. *Advances in Neural Information Processing Systems*, 32:3052–3062, 2019a.

Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-Wasserstein learning for graph matching and node embedding. In *International Conference on Machine Learning*, pages 6932–6941. PMLR, 2019b.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in Neural Information Processing Systems*, 31, 2018.

Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34:249–270, 2022.

Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 912–919, 2003.

Titre : Prédiction structurée avec régularisation de sortie : améliorer les performances statistiques et computationnelles

Mots clés : théorie de l'apprentissage statistique, prédiction structurée, méthodes à noyaux, transport optimal

Résumé : Les algorithmes d'apprentissage supervisé visent à identifier des relations entre des entrées et des sorties grâce à des jeux d'entraînement constitués de couples (entrée, sortie). La situation d'apprentissage supervisé la plus étudiée considère des entrées de grande dimension et des sorties de faible dimension, comme les nombres réels dans le cas de la régression, et les valeurs zéro ou un dans le cas de la classification binaire. Néanmoins, être capable de prédire des sorties complexes, comme des graphes, des séquences ou des images, permet de résoudre plus de tâches en pratique. C'est le problème traité par la prédiction structurée. La question qui a motivé cette thèse est la suivante : comment tirer parti de la structure de l'espace de sortie pour obtenir des méthodes de prédiction structurée statistiquement et computationnellement performantes ? Nous essayons de répondre à cette question à travers le prisme des méthodes de substitution pour la prédiction structurée. Plus précisément, ce manuscrit commence par considérer le problème de la prédiction de graphes. Nous proposons de mettre à profit la distance de Gromov-Wasserstein (GW), définissant une géométrie naturelle pour les espaces de graphes, en tant que fonction de perte, donnant lieu à une nouvelle famille de modèles pour la prédiction de graphes : les modèles barycentriques de GW. Dans une deuxième contribution, nous proposons de généraliser la régression à rang réduit aux espaces de sortie non linéaires. La méthode proposée consiste à résoudre les problèmes de régression des méthodes de substitution grâce à un estimateur de régression à rang réduit. Nous menons une étude théorique de l'estimateur de rang réduit proposé, et prouvons sous des hypothèses de régularité de sortie que la régularisation de rang est statistiquement et computationnellement bénéfique. En particulier, nos résultats étendent l'intérêt de la régression à rang réduit au-delà du cas standard où l'optimum est supposé de rang fini et faible. Dans une troisième contribution, nous proposons un principe de régularisation de la fonction de perte. La méthode proposée vise à obtenir des gains statistique et computationnel en prédiction structurée, grâce à l'exploitation de données de sortie supplémentaires et des informations de régularité sur la fonction de perte. Nous étudions théoriquement dans quelle situation la méthode est en effet bénéfique. Nos résultats montrent qu'il est bénéfique d'adapter le niveau de détail des objets structurés prédits, en fonction de la quantité de données d'entraînement disponible, pour réduire les effets de la variance de sortie (ou du bruit d'étiquetage) d'une part, et pour alléger la complexité computationnelle de la prédiction d'autre part.

Title : Structured Prediction with Output Regularization: Improving Statistical and Computational Efficiency

Keywords : statistical learning theory, structured prediction, kernel methods, optimal transport

Abstract : Supervised learning algorithms aim at identifying relationships between inputs and outputs thanks to training sets of couples (input, output). The most studied setting of supervised learning deals with high-dimensional inputs but low-dimensional outputs, as, for example, real numbers in the case of regression, and the values zero or one in the case of binary classification. Nevertheless, being able to predict complex outputs, such as graphs, sequences, or images, allows for addressing much more practical tasks. This is the so-called structured output prediction setting. The question that has motivated this thesis is the following: How to take advantage of the structure of the output space in order to obtain statistically and computationally efficient structured prediction methods? We try to answer this question through the lens of the structured prediction framework of surrogate methods. More precisely, this manuscript starts by considering the problem of graph prediction. We propose to leverage the Gromov-Wasserstein (GW) distance, carrying a natural geometry for graph spaces, as a loss function. From this idea, we derive a new family of models for graph prediction: GW barycentric models. In a second contribution, we propose a generalization of reduced-rank regression which allows handling non-linear output spaces. It consists in solving the surrogate regression problems appearing in surrogate methods thanks to a reduced-rank regression estimator. We carry out a theoretical study of the reduced-rank estimator, taking values in a Hilbert space of possibly infinite dimension, and prove under output regularity assumptions that the rank regularization is statistically and computationally beneficial. Our results extend the interest of reduced-rank regression beyond the standard setting where the optimum is assumed to be low-rank. In a third contribution, we propose the principle of loss regularization. The method aims at obtaining a statistical and computational gain in structured prediction, by exploiting additional output data, and regularity information on the loss function. We study theoretically under which setting the method is beneficial. Our results show, intuitively, that one had better adapt the level of detail of the structured outputs predicted with respect to the quantity of training data, to reduce the effects of the output variance (or labeling noise), and also to alleviate the computational complexity of the pre-image in surrogate methods.

Institut Polytechnique de Paris
91120 Palaiseau, France