SPARSE LEARNING OF PARSIMONIOUS REPRODUCING KERNEL HILBERT SPACE MODELS

Maria Peifer, Luiz F. O. Chamon, Santiago Paternain, and Alejandro Ribeiro

Electrical and Systems Engineering, University of Pennsylvania
e-mail: {mariaop,luizf,spater,aribeiro}@seas.upenn.edu

ABSTRACT

Reproducing kernel Hilbert spaces (RKHSs) have been at the core of successful non-parametric tools in signal processing, statistics, and machine learning. Despite their success, the computational complexity of these models often hinders their use in practice. Indeed, fitting RKHS models typically relies on representer theorems to express the solution space as a combination of kernels evaluated at the training samples. Thus, the computational cost of evaluating these models is proportional to the number of training samples, which in many applications is prohibitively high. This issue is often addressed by sparsifying the coefficients of the kernel expansion, despite the fact that classical representer theorems no longer hold in the presence of sparsity penalties. In this work, we propose to directly tackle sparse learning over RKHSs by posing it as a functional problem. In other words, by formulating the RKHS model as a sparse, continuous combination of atoms from an overparametrized, continuous dictionary containing the value of the kernel evaluated at every point of the function domain. We show that despite the infinite dimensionality and non-convexity of the underlying optimization problem, these models can be fit exactly and efficiently using duality. We illustrate the performance of this technique in numerical experiments.

Index Terms— Kernel methods, RKHS, sparsity, kernel selection, functional optimization

1. INTRODUCTION

Reproducing kernel Hilbert spaces (RKHSs) methods are fundamental tools in signal processing, statistics, and machine learning [1–5]. These non-parametric techniques seek to fit the data using functions lying in a given RKHS. The attractiveness of RKHSs comes from the fact that their functions can be written as a (possibly infinite) linear combination of so-called reproducing kernels evaluate over the function domain. Kernels in this context are simply positive definite functions [2]. Furthermore, when fitting a function with an RKHS norm penalty, representer theorems show that the solution can be expressed as a combination of kernels evaluated only at the data points [6, 7]. In other words, the original functional program can be formulated as finite dimensional problem. Nevertheless, this can lead to computational complexity issues that hinder the use of RKHS models in practice [5, 8, 9].

Although representer theorems reduce the problem of estimating smooth functions in RKHSs to that of estimating coefficients of linear combinations, there are as many coefficients as the number of training samples. In many application, the resulting complexity of evaluating the function is therefore prohibitively high. Typically, this issue is addressed by imposing a sparsity penalty on the coefficients to reduce the number of kernel evaluations. To cope with the combinatorial nature of the resulting problem, ℓ_1-norm relaxations [10] or greedy heuristics [8, 11] are then often deployed. Many of these methods, however, implicitly rely on the classical representer theorems [6, 7], despite the fact that they no longer hold in the presence of the sparsity penalties (see Remark 1). Hence, even if the sparse problem could be solved exactly, the solution would remain suboptimal with respect to the original functional program.

In this work, we propose to directly tackle the problem of finding the function in an RKHSs that fits the data and can be represented with as few kernels as possible. To solve this non-convex, infinite dimensional problem, we use an "overparametrize then simplify" approach. First, we express the functions in the RKHS as a (continuous) combination of atoms from an overparametrized dictionary containing the kernel evaluated at every point of the domain. By minimizing the support of the continuous combination coefficients when fitting the model, i.e., by making them sparse, we simplify the function representation and determine the minimum number of kernels necessary and their centers. Despite the infinite dimensional and non-convex nature of the resulting problem, it can be solved exactly and efficiently using duality.

To derive this approach, we first formulate the sparse RKHS model problem (Section 2.1) and recast it in functional terms (Section 2.2). Then, we prove that strong duality holds for this functional problem (Section 3.1), thus showing that it can be solved exactly through its dual problem. Leveraging this result, we propose an algorithm to fit sparse RKHS models (Section 3.2) and use it to illustrate the effectiveness of this functional approach in finding parsimonious RKHS models (Section 4).

2. PROBLEM FORMULATION

2.1. Parsimonious RKHS models

Consider the dataset $\{(\boldsymbol{x}_n, y_n)\}$, $n = 1, \ldots, N$, where $\boldsymbol{x}_n \in \mathbb{R}^p$ is the n-th feature vector and $y_n \in \mathbb{R}$ is the corresponding observation (or label for classification problems) and an RKHS $\mathcal{H}$. Our goal is to find a function $f \in \mathcal{H}$ such that the $\hat{y}_n = f(\boldsymbol{x}_n)$ minimize some loss with respect to y_n. This problem, however, is underdetermined for finite N due to the infinite dimensionality of the function space $\mathcal{H}$. An additional penalty function $\rho : \mathcal{H} \to \mathbb{R}$ is used to overcome this issue, leading to the optimization problem

$$\begin{aligned} \underset{f \in \mathcal{H}}{\text{minimize}} \quad & \rho(f) \\ \text{subject to} \quad & (y_n - \hat{y}_n)^2 \leq \epsilon, \quad n = 1, \ldots, N \\ & \hat{y}_n = f(\boldsymbol{x}_n) \end{aligned} \tag{PI}$$

where $\epsilon > 0$ is a constant error bound. Though performance metrics than the square loss can be used, we restrict ourselves to this case in this manuscript for simplicity. The most common choice of penalty promotes smoothness by taking $\rho(f) = \|f\|_{\mathcal{H}}$, the RKHS norm [1,

2, 8]. In this case, the representer theorem in [6, 7] can be used to reduce the functional (PI) to a finite dimensional problem. Indeed, it can be shown that the solution of (PI) regularized by the RKHS norm is of the form

$$f(\cdot) = \sum_{n=1}^{N} a_n \kappa(\boldsymbol{x}_n, \cdot), \tag{1}$$

where $\kappa : \mathbb{R}^p \times \mathbb{R}^p$ is the reproducing kernel of $\mathcal{H}$ and $a_n \in \mathbb{R}$. In other words, f can be written as a combination of kernels evaluated at the data points [6, 7]. Using (1), (PI) therefore reduces to optimizing the coefficients a_n. Still, note that the computational complexity of evaluating f is proportional to the sample size N, which can hinder the use of RKHS methods in many applications [8, 12].

This issue is often addressed by incorporating a sparsity penalty in the objective of (PI). To do so, observe that the reproducing kernels of an RKHS form a basis for its functions, so that $f \in \mathcal{H}$ can be written as a (possibly infinite) linear combination of kernels [13]. We can therefore fit sparse RKHS models using

$$\begin{aligned} \underset{a_i \in \mathbb{R},\ \boldsymbol{z}_i \in \mathbb{R}^p}{\text{minimize}} \quad & \rho(f) + \gamma \sum_{i=1}^{\infty} \mathbb{I}(a_i \neq 0) \\ \text{subject to} \quad & (y_n - \hat{y}_n)^2 \leq \epsilon, \quad n = 1, \dots, N \\ & \hat{y}_n = f(\boldsymbol{x}_n) = \sum_{i=1}^{\infty} a_i \kappa(\boldsymbol{z}_i, \boldsymbol{x}_n) \end{aligned} \tag{PII}$$

where $\mathbb{I}(a_i \neq 0) = 1$ when a_i is non-zero and zero otherwise, $\gamma > 0$ is a parameter that controls the sparsity of the solution, and the $\boldsymbol{z}_i$ are the kernel centers. Notice that, in contrast to (1), f in (PII) is written as an infinite combination of kernels over arbitrary centers. In the presence of sparsity penalties, the representer theorem does not hold (see Remark 1). In fact, using (1) in (PII), as is often considered in the literature, leads to suboptimal solutions since classical representer theorems no longer hold in the presence of sparsity penalties (see Remark 1).

Without relying on representer theorems, however, solving (PII) is challenging due to its infinite dimensionality and non-convexity. To overcome this issue, we reformulate (PII) as a sparse functional program in the next section. At first, this may appear to be a distinction without a difference since sparse functional programs are both infinite dimensional and non-convex as well. Nevertheless, this turns out to be a fruitful approach in view of the strong duality result from Section 3.1.

Remark 1. In the presence of sparsity penalties, as in (PII), classical representer theorems [6, 7] no longer hold and solutions are not necessarily of the form (1). Indeed, consider the following counterexample which we illustrate Figure 1. Construct a dataset by taking $y_i = f^o(\boldsymbol{x}_i)$, where $f^o(\boldsymbol{x}) = \kappa(\boldsymbol{z}, \boldsymbol{x})$. Let $f_\gamma^\star$ be the solution of (PII) for a given choice of parameter γ. Hence, we know $f_0^\star$ must have the form (1) due to the representer theorem in [7]. However, notice that f^o itself is the sparsest representation of this dataset, so there exist Γ such that $f_\gamma^\star(\cdot) = f^o(\cdot)$ for $\gamma \geq \Gamma$. Thus, unless $\boldsymbol{x}_i = \boldsymbol{z}$ for some i, the solution of (PII) need not be of the form (1).

2.2. A functional reformulation

Our approach to solving (PII) is to express its solutions as combinations of atoms from an overparametrized, continuous dictionary containing the kernel evaluated at every point of the domain. By imposing sparsity on the continuous combination coefficients, we then simplify this functional model to recover the minimum number of

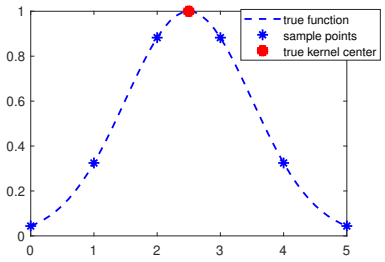


Fig. 1. Signal with the kernel center not part of the sampling set

kernels and their centers. For this reason, we dub this approach "overparametrize then simplify."

Formally, start by rewriting the definition of f in (PII) in functional terms. In other words, replace the discrete coefficients a_n by a function $\alpha : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D} \subset \mathbb{R}^p$ is a compact set representing the domain of f. For instance, take $\mathcal{D}$ to be the convex hull of the feature vectors $\boldsymbol{x}_n$. Then, write f as

$$f(\cdot) = \int_{\mathcal{D}} \alpha(\boldsymbol{x}) \kappa(\boldsymbol{x}, \cdot) d\boldsymbol{x}. \tag{2}$$

Note that the definition of f in (PII) is recovered for $\alpha(\boldsymbol{z}) = \sum_{i=1}^{\infty} a_i \delta(\boldsymbol{z} - \boldsymbol{z}_i)$, where δ is the Dirac delta distribution. Hence, every (PII)-feasible function can be written using (2). We can therefore formulate the functional version of (PII):

$$\begin{aligned} \underset{\alpha \in L_2}{\text{minimize}} \quad & \int_{\mathcal{D}} \Big\{ h\left[\alpha(\boldsymbol{z}), \boldsymbol{z}\right] + \gamma\, \mathbb{I}\left[\alpha(\boldsymbol{z}) \neq 0\right] \Big\} d\boldsymbol{z} \\ \text{subject to} \quad & (y_n - \hat{y}_n)^2 \leq \epsilon, \quad n = 1, \dots, N \\ & \hat{y}_n = f(\boldsymbol{x}_n) = \int_{\mathcal{D}} \alpha(\boldsymbol{z}) \kappa(\boldsymbol{z}, \boldsymbol{x}_n) d\boldsymbol{z} \end{aligned} \tag{PIII}$$

where h is an arbitrary regularization function. For instance, (PIII) can penalize the RKHS norm of the solution by taking $h(x, \boldsymbol{z}) = x^2 \int_{\mathcal{D}} \kappa(\boldsymbol{z}, \boldsymbol{y}) d\boldsymbol{y}$. Due to space constraints, we consider $h(x, \boldsymbol{z}) = x^2/2$ in what follows and thus take $\alpha \in L_2$ in (PIII). Though other choices are possible, they will be explored in future work. It is worth noting that since α in (PIII) is a function, it cannot contain Dirac deltas. Thus, the solution of (PIII) $\alpha^\star$ is actually composed of bump functions around the correct kernel centers (see Fig. 2).

Still, the issue remain of how to solve (PIII) given that, as (PII), it is both infinite dimensional and non-convex. In the sequel, we show that (PIII) has null duality gap and can be solved exactly and efficiently using its dual problem.

3. FITTING SPARSE RKHS MODELS

3.1. Learning in the dual domain

We address the infinite dimensionality and non-convexity of (PIII) by using duality. To be sure, the dual problem of (PIII) has dimension $2N$ and is convex by definition [14]. Still, it is not straightforward that a solution of (PIII) can be obtained by solving its dual. Although duality is often used to solve semi-infinite convex programs, this is due to the fact that strong duality holds under mild conditions [15]. In this section, we show that this is also the case for (PIII).

To do so, we start by deriving the dual problem of (PIII). Introduce the dual variables $\lambda_n \in \mathbb{R}$ associated with the equality constraints and $\mu_n \in \mathbb{R}_+$ associated with the inequality constraints, to

write the Lagrangian of (PIII) as

$$\begin{aligned}\mathcal{L}\left(\alpha, \hat{y}_n, \lambda_n, \mu_n\right) = &\int_{\mathcal{D}} \left\{ \frac{\alpha(\boldsymbol{z})^2}{2} + \gamma\, \mathbb{I}\left[\alpha(\boldsymbol{z}) \neq 0\right] \right\} d\boldsymbol{z} \\ &+ \sum_{n=1}^{N} \lambda_n \left[\hat{y}_n - \int_{\mathcal{D}} \alpha(\boldsymbol{z}) \kappa(\boldsymbol{z}, \boldsymbol{x}_n) d\boldsymbol{z} \right] \\ &+ \sum_{n=1}^{N} \mu_n \left\{ (y_n - \hat{y}_n)^2 - \epsilon \right\}. \end{aligned} \tag{3}$$

Thus, its dual function is given by

$$g(\lambda_n, \mu_n) = \min_{\alpha \in L_2,\, \hat{y}_n} \mathcal{L}(\alpha, \hat{y}_n, \lambda_n, \mu_n) \tag{4}$$

and its dual problem can be written as

$$\underset{\mu_n \geq 0}{\text{maximize}} \quad g(\lambda_n, \mu_n). \tag{DIII}$$

To show how a solution of (PIII) can be obtained by solving (DIII), we leverage the following result:

Theorem 1. *Suppose that κ has no point masses (Dirac deltas) and that Slater's condition holds for* (PIII)*. Then,* (PIII) *has null duality gap, i.e., $P = D$ for P, the optimal value of* (PIII)*, and D, the optimal value of* (DIII)*.*

Proof. See [16]. ∎

Theorem 1 provides an efficient way of solving (PIII). Indeed, let $\lambda_n^\star, \mu_n^\star$ be solutions of (DIII) and $\alpha^\star, \hat{y}_n^\star$ be solutions of (PIII). Then, it holds that

$$(\alpha^\star, \hat{y}_n^\star) = \operatorname*{argmin}_{\alpha \in L_2,\, \hat{y}_n} \mathcal{L}(\alpha, \hat{y}_n, \lambda_n^\star, \mu_n^\star). \tag{5}$$

The equality in (5) comes from the fact that the Lagrangian (3) is strongly convex, so that the argmin set is a singleton [14]. Naturally, the efficiency of this solution depends on being able to efficiently evaluate the minimum in (4) and (5). The following proposition shows that this is indeed the case.

Proposition 1. *Consider the Lagrangian in* (3)*. Then, the minimum in* (4) *is achieved for*

$$\alpha_d(\boldsymbol{z}, \lambda_n) = \begin{cases} \sum_{n=1}^{N} \lambda_n \kappa(\boldsymbol{x}_n, \boldsymbol{z}) & \sum_{n=1}^{N} \lambda_n \kappa(\boldsymbol{x}_n, \boldsymbol{z}) > \sqrt{2\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$\hat{y}_{n,d}(\lambda_n, \mu_n) = y_n - \frac{\lambda_n}{2\mu_n} \tag{7}$$

Proof. Start by noticing that the joint minimization in (4) can be separated into

$$\begin{aligned} g(\lambda_n, \mu_n) = &\min_{\alpha \in L_2} \int_{\mathcal{D}} F\left[\alpha(\boldsymbol{z}), \boldsymbol{z}\right] d\boldsymbol{z} \\ &+ \min_{\hat{y}_n} \sum_{n=1}^{N} \left[\lambda_n \hat{y}_n + \mu_n (y_n - \hat{y}_n)^2 \right] - \epsilon \sum_{n=1}^{N} \mu_n, \end{aligned} \tag{8}$$

with $F(a, \boldsymbol{z}) = a^2/2 + \gamma\, \mathbb{I}(a \neq 0) - \sum_n \lambda_n a \kappa(\boldsymbol{x}_n, \boldsymbol{z})$. The second minimization in (8) is a simple quadratic program whose close form solution is (7).

The first minimization, on the other hand, is a non-convex functional problem. It can, however, be solved efficiently by leveraging the separability F. To do so, we use the following lemma:

Algorithm 1 Stochastic dual ascent for (PIII)

$\lambda_n^{(0)} = 0$, $\mu_n^{(0)} = 1$
for $t = 1, \ldots, T$
  Draw $\boldsymbol{z}_j$, $j = 1, \ldots, J$, uniformly at random from $\mathcal{D}$
  $\hat{\partial}_{\lambda_n} = y_n - \dfrac{\lambda_n^{(t-1)}}{2\mu_n^{(t-1)}} - \sum_{m=1}^{N} \lambda_m^{(t-1)} \dfrac{1}{J} \sum_{j=1}^{J} \kappa(\boldsymbol{x}_m, \boldsymbol{z}_j) \kappa(\boldsymbol{z}_j, \boldsymbol{x}_n)$
  $\partial_{\mu_n} = \left(\dfrac{\lambda_n^{(t-1)}}{2\mu_n^{(t-1)}} \right)^2 - \epsilon$
  $\lambda_n^{(t)} = \lambda_n^{(t-1)} + \eta_\lambda \hat{\partial}_{\lambda_n}$
  $\mu_n^{(t)} = \mu_n^{(t-1)} + \eta_\mu \partial_{\mu_n}$
  Compute $\alpha^{(t)}$ using (6) with $\lambda_n^{(t)}$
end
$\alpha^\star = \dfrac{1}{T} \sum_{t=1}^{T} \alpha^{(t)}$

Lemma 1. *Let $F(a, \boldsymbol{z})$ be a normal integrand, i.e., continuous in a for all fixed $\boldsymbol{z}$ and measurable in $\boldsymbol{z}$ for all fixed a. Then,*

$$\inf_{\alpha \in L_2} \int_{\mathcal{D}} F\left[\alpha(\boldsymbol{z}), \boldsymbol{z}\right] d\boldsymbol{z} = \int_{\mathcal{D}} \inf_{a \in \mathbb{R}} F(a, \boldsymbol{z}) d\boldsymbol{z}. \tag{9}$$

Proof. See [17, Thm. 3A]. ∎

Theorem 1 implies that the minimization with respect to α in (8) can be solved individually for each $\boldsymbol{z}$. The problem then becomes a scalar optimization problem involving a quadratic program followed by thresholding as in (6). ∎

In the sequel, we propose an algorithm to obtain a solution of (PIII) by solving its dual problem (DIII).

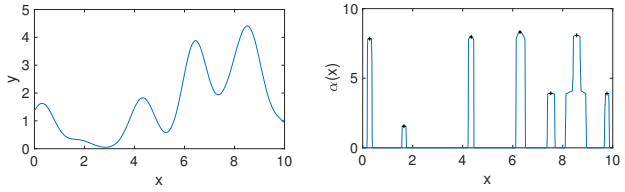
3.2. Solving the dual problem

In this section, we present an algorithm based on stochastic subgradient ascent that obtains a solution of (PIII) $\alpha^\star$ by solving (DIII). First, recall that the constraint slacks evaluated at the dual minimizers are subgradients of the dual problem [14]. Explicitly, using Proposition 1 we obtain

$$\begin{aligned} \partial_{\lambda_n} g(\lambda_n, \mu_n) = &\, y_n - \frac{\lambda_n}{2\mu_n} \\ &- \sum_{m=1}^{N} \lambda_m \int_{\mathcal{S}} \kappa(\boldsymbol{x}_m, \boldsymbol{z}) \kappa(\boldsymbol{z}, \boldsymbol{x}_n) d\boldsymbol{z} \end{aligned} \tag{10}$$

$$\partial_{\mu_n} g(\lambda_n, \mu_n) = \frac{\lambda_n^2}{4\mu_n^2} - \epsilon \tag{11}$$

where $\mathcal{S} = \{\boldsymbol{z} \in \mathcal{D} \mid \sum_n \lambda_n \kappa(\boldsymbol{x}_n, \boldsymbol{z}) > \sqrt{2\gamma}\}$. Observe that evaluating (10) involves computing an integral over a set $\mathcal{S}$ that depends on the dual variables. Although typical numerical integration methods such as Simpson's method could be used, estimating (10) using Monte Carlo integration leads to the stochastic subgradient method described in Algorithm 1, where $\eta_\lambda, \eta_\mu > 0$ are the update step sizes and J is the mini-batch size. Taking $J = 1$ recovers the classical stochastic subgradient method. Since Monte Carlo integration yields an unbiased estimator of the integral, $\hat{\partial}$ is an unbiased estimator of (10). Hence, typical convergence guarantees hold for Algorithm 1 [18, 19].



(a) True function (b) $\alpha^\star(x)$

Fig. 2. A function and its corresponding $\alpha^\star$.

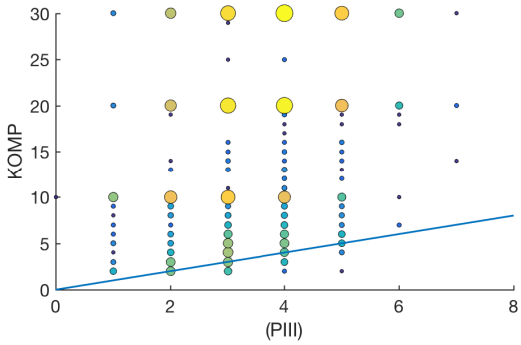


Fig. 3. Number of kernels needed to achieve error level for a function made of $m = 5$ kernels.

4. NUMERICAL EXPERIMENTS

To illustrate the performance of the proposed functional approach, we use a superposition of m Gaussian kernel with bandwidth $\sigma = 0.5$ to simulate functions f^o used to generate the training set. Throughout the experiments, the center $\bar{x}_i$ of the Gaussian kernels and their coefficients a_i were drawn uniformly at random over $\mathcal{D} = [0, 10]$ and $[1, 2]$ respectively. The data pairs (x_n, y_n) are constructed by selecting x_n randomly from $\mathcal{D}$ and taking

$$y_n = f^o(x_n) + v_n, \tag{12}$$

where $\{v_n\}$ are independent zero-mean Gaussian random variables with variance $\mathbb{E}[v_n^2] = 10^{-3}$. We used the generalization MSE as a figure of merit, which is evaluated over a separate test set containing 500 data points of the form (12).

We compare the performance of (PIII) with that of KOMP [11], a commonly used backward greedy selection method that iteratively removes kernel centers until a preset estimation error is obtained. For (PIII), we compute $\alpha^\star$ using Algorithm 1 with $\gamma = 30$, $\epsilon = 10^{-3}$, and $T = 1000$ and extract the kernel centers from its peaks (see Fig. 2). The a_i are evaluated by solving a subsequent least squares problem. Both methods used the true kernel bandwidth σ.

Figures 3 and 4 compares the number of kernels required by KOMP to achieve the same generalization MSE as the solution obtained by solving (PIII) for 1000 realizations of f^o and data set of sizes $N = \{2m, 4m, 6m\}$. First, notice that the functional approach rarely uses more than the actual number of kernels m in f^o. In contrast, KOMP often over estimates the number of kernels required to describe f^o. This is related to the fact that KOMP only places kernels on the training samples. On the other hand, the current functional approach can estimate the correct kernel centers. Finally, observe that KOMP found a smaller number of kernels than (PIII) only 4% and 0.4% of the realizations for $m = 5$ (Fig. 3) and $m = 10$ (Fig. 4)

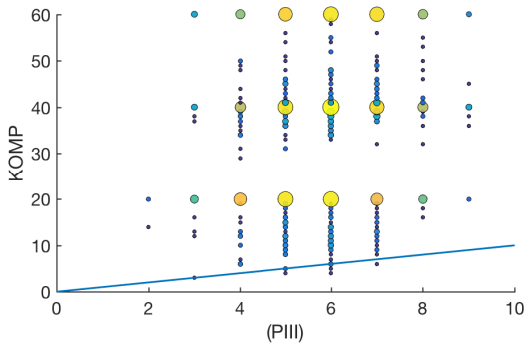


Fig. 4. Number of kernels needed to achieve error level for a function made of $m = 10$ kernels.

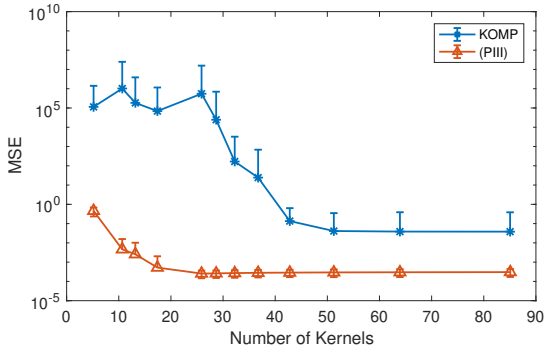


Fig. 5. Generalization MSE as a function of number of kernels.

kernels respectively. This occured due to the choice of regularization parameter γ and early stopping of Algorithm 1.

Figure 5 shows the evolution of the generalization MSE as the number of kernels used increases for $m = 10$ and a training set of $N = 100$ data points. Once again, the ability of (PIII) of moving the kernel centers beyond the training set allows it to achieve significantly lower errors than KOMP. This disparity decreases the number of kernels used increases. Observe that after placing approximately 25 kernels, the performance of (PIII) plateaus. This is only 2.5 times more than the actual number of kernels in the function. For KOMP, a plateau is only reached after approximately 50 kernels are used.

5. CONCLUSION

We tackled the problem of finding a function in an RKHSs that fits the data and can be represented with as few kernels as possible. To do so, we formulated the RKHS model as a sparse, continuous combination of atoms from an overparametrized dictionary containing the value of the kernel evaluated at every point of the function domain. Despite the infinite dimensionality and non-convexity of this problem, we proved that it has null duality gap and can therefor be solved exactly and efficiently through its dual problem. We proposed a stochastic subgradient ascent algorithm to solve this problem and illustrated its performance in numerical experiments. We foresee that locally adapting the kernel as well as the kernel center would allow for even more parsimonious models and enable us to account for more challenging applications, such as those involving functions with varying degrees of smoothness.

6. REFERENCES

[1] R. Rosipal and L.J. Trejo, "Kernel partial least squares regression in reproducing kernel Hilbert space," *Journal of Machine Learning Research*, vol. 2, no. Dec, pp. 97–123, 2001.

[2] Christopher M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[3] M. Yuan and T.T. Cai, "A reproducing kernel Hilbert space approach to functional linear regression," *The Annals of Statistics*, vol. 38, no. 6, pp. 3412–3444, 2010.

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

[5] J. Arenas-Garcia, K.B. Petersen, G. Camps-Valls, and L.K. Hansen, "Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods," vol. 30[4], pp. 16–29, 2013.

[6] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *Journal of mathematical analysis and applications*, vol. 33, no. 1, pp. 82–95, 1971.

[7] B. Schölkopf, R. Herbrich, and A.J. Smola, "A generalized representer theorem," in *International conference on computational learning theory*. Springer, 2001, pp. 416–426.

[8] A.J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *ICML*, 2000, pp. 911–918.

[9] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," in *ICASSP*. IEEE, 2017, pp. 4671–4675.

[10] E.J. Candés, M.B. Wakin, and S.P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[11] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, no. 1-3, pp. 165–187, 2002.

[12] F.R. Bach and M.I. Jordan, "Predictive low-rank decomposition for kernel methods," in *ICML*, 2005, pp. 33–40.

[13] C.J.C. Burges B. Schölkopf and A.J. Smola, *Advances in Kernel Methods: Support Vector Learning*, MIT Press, 1998.

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

[15] A. Shapiro, "On duality theory of convex semi-infinite programming," *Optimization*, vol. 54[6], pp. 535–543, 2006.

[16] L.F.O. Chamon, Y. C. Eldar, and A. Ribeiro, "Strong duality of sparse functional optimization," in *ICASSP*, 2017, pp. 4739–4743.

[17] R. T. Rockafellar, *Integral functionals, normal integrands and measurable selections*, Springer, 1976.

[18] A. Ruszczyński and W. Syski, "On convergence of the stochastic subgradient method with on-line stepsize rules," *Journal of Mathematical Analysis and Applications*, vol. 114, no. 2, pp. 512–527, 1986.

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