Learning Positive Functions in a Hilbert Space

J. Andrew Bagnell
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
dbagnell@ri.cmu.edu

Amir-massoud Farahmand
Mitsubishi Electric Research Laboratories
Cambridge, MA, USA
farahmand@merl.com

Abstract

We develop a method for learning positive functions by optimizing over SoS_K, a reproducing kernel Hilbert space subject to a Sum-of-Squares (SoS) constraint. This constraint ensures that only nonnegative functions are learned. We establish a new representer theorem that demonstrates that the regularized convex loss minimization subject to the SoS constraint has a unique solution and moreover, its solution lies on a finite dimensional subspace of an RKHS that is defined by data. Furthermore, we show how this optimization problem can be formulated as a semidefinite program. We conclude with examples of learning such functions.

1 Introduction

The goal of this paper is to introduce a new framework for learning functions that fits data by minimizing a convex loss while guaranteeing that the estimated function is positive (or nonnegative, to be more precise). Even though there are many methods for learning a function under a convex loss criteria in the machine learning and statistics literature, they do not guarantee the positiveness of the estimate as a built-in feature of the method. Of course one can always truncated the estimator's output to make it nonnegative, but one may argue that this is not an elegant approach: the positiveness condition is not an intrinsic part of the method.

To design an estimator that guarantees the positivity of its output, we bring together two different family of concepts and tools. The first one is the concept of Sum of Squares (SoS) and positive polynomials, cf. Nesterov [8], Parrilo [9], Lasserre [5], Parrilo [10], Ghasemi [3]. SoS has already several applications, e.g., in control theory [9], but as far as we know it has rarely used for machine learning problems (one exception is by Magnani et al. [7], who use SoS polynomials to fit a convex polynomial to a set of data points). The second one is the concept of reproducing kernel Hilbert spaces (RKHS), which is quite familiar to a machine learning audience, e.g., Aronszajn [1], Schölkopf and Smola [13], Steinwart and Christmann [15].

To understand the basic concept behind SoS, we briefly review the positive semidefinite (PSD) (or nonnegative) polynomials, and its relation to SoS. Consider a univariate real polynomial $p(x)$ with $x \in \mathbb{R}$. It is called positive semidefinite or nonnegative if $p(x) \geq 0$ for all x in its domain. This polynomial is a sum of squares if there exist some other polynomials $q_1(x), q_2(x), \ldots$ such that $p(x) = \sum_i q_i^2(x)$. For the univariate polynomials the condition of being a PSD polynomial is equivalent to the condition of being an SoS polynomial. More generally for multivariate polynomials, being an SoS implies that the polynomial is PSD. The converse, however, is not generally true as there are PSD polynomials that do not have an SoS representation (Hilbert characterizes the conditions when a PSD polynomial can be written as an SoS polynomial; cf. Ghasemi [3] for a review). Nevertheless, SoS polynomials define an important subset of PSD polynomials. One particular reason is that verifying that a polynomial is an SoS is computationally feasible as it can be done by solving a semidefinite program (SDP), but verifying the positivity of a polynomial is NP-hard. Furthermore, there are also some results regarding the denseness of the space of SoS poly-

nomials within the space of PSD polynomials, e.g., cf. Lasserre and Netzer [6], Ghasemi [3], which may justify them as a rich enough subset of positive polynomials.

To make the connection of the PSD polynomials and SoS more concrete, suppose that $p(x)$ is of degree $2d$, so $p(x) = \sum_{i=0}^{2d} w_i x^i$ (this discussion is borrowed from the lecture notes of Parrilo [11]). Define the vector of monomials $\phi(x) = \left[1, x, \ldots, x^d\right]^\top$. Let $p(x)$ have the SoS representation of $p(x) = \sum_{i=1}^{m} q_i^2(x)$. We can write

$$\bar{q}(x) = \begin{bmatrix} q_1(x) \\ \vdots \\ q_m(x) \end{bmatrix} = V\phi(x),$$

with V being an $m \times (d+1)$ matrix whose i-th row corresponds to the coefficients of q_i. Therefore, $p(x) = \sum_{i=1}^{m} q_i^2(x) = \bar{q}^\top(x)\bar{q}(x) = \phi^\top(x)V^\top V\phi(x)$. Let us define $Q = V^\top V$, which is $d + 1 \times d + 1$ matrix. By construction, Q is a PSD matrix. Conversely for a PSD $Q \succeq 0$, one can decompose it as $Q = V^\top V$ and obtain an SoS representation. Therefore, for a polynomial $p(x) = \phi^\top(x)Q\phi(x)$, having $Q \succeq 0$ is a sufficient condition to ensure that the polynomial is nonnegative (but it is not a necessary condition).

The previous discussion focused on polynomials. This is common in the discussion of SoS in the algebraic geometry context as polynomials define a communicative ring, so one can use algebraic tools to study and analyze them. For our problem of learning a positive function from data, we do not want to be limited to the space of polynomials. In fact, the vector of features ϕ, defined above, does not need to be a vector consisting of monomials. One can see ϕ as a feature of x, for which the set of monomials is a particular case. It is easy to see that no matter what ϕ we choose, the condition on $Q \succeq 0$ leads to a function $p(x)$, not necessarily a polynomial anymore, be nonnegative.

Seeing ϕ as features opens up the possibility of using a variety of features, which is a common practice in machine learning. But this also leads to the problem of how to choose a proper set of features such that the target function can be represented well enough. Our approach is to consider ϕ as the feature from x to a reproducing kernel Hilbert space (RKHS). Thanks to the reproducing property of an RKHS, we have a representer theorem for an RKHS that allows us to design efficient algorithms that do not need an explicit representation for ϕ. Unfortunately the usual representer theorems for RKHS (e.g., Schölkopf et al. [14], Steinwart and Christmann [15]) are not suitable for our problem of learning with the additional constraint of being an SoS function.

In this paper we formulate learning with positive functions in an RKHS with SoS constraint. We prove that the optimization problem required for learning a PSD function from data has a unique solution. We also present a representer theorem stating that the optimal solution has a finite representation that depends on data (Section 2.3). We also show that the optimization problem can be cast as a semidefinite program, so it can be solved efficiently (Section 3). Finally we illustrate the algorithm on a simple problem (Section 4).

2 Learning Positive Functions in an RKHS

2.1 Losses

We review a few standard definitions from Chapter 2 of Steinwart and Christmann [15]. Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ be the output space. Denote the pointwise loss function by $l : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to [0, \infty)$. A pointwise loss function is called (strictly) convex if $l(x, y, \cdot) : \mathbb{R} \to \mathbb{R}$ is (strictly) convex for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Let $\mu \in \mathcal{M}(\mathcal{X} \times \mathcal{Y})$ be a probability distribution on $\mathcal{X} \times \mathcal{Y}$. For a measurable function $f : \mathcal{X} \to \mathbb{R}$, we define the expected loss as $L(f) = \mathbb{E}_\mu \left[l(X, Y, f(X))\right] = \int_{\mathcal{X}\times\mathcal{Y}} l(x, y, f(x)) \, \mathrm{d}\mu(x, y)$. Given a dataset $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$ with $(X_i, Y_i) \sim \mu$, the empirical loss is defined as $L_n(f) = \frac{1}{n}\sum_{i=1}^n l(X_i, Y_i, f(X_i))$. It is easy to see that the convexity of l as a function of its third argument implies the convexity of $L(\cdot)$ and $L_n(\cdot)$.

The problem of learning a positive function that fits data well can be formulated as the following optimization problem:

$$\hat{f} \leftarrow \operatorname*{argmin}_{f \in \mathcal{F}, f \geq 0} \; L_n(f) + \lambda J(f), \tag{1}$$

in which $\mathcal{F}$ is a function space and $J(f)$ is a properly defined regularizer (penalty) that controls the complexity of the estimate in the function space $\mathcal{F}$. From the statistical point of view, the goal can be seen as ensuring that the estimate $\hat{f}$ has a small excess loss $L(\hat{f}) - \min_{f \in \mathcal{F}, f \geq 0} L(f)$. The proper choices of $\mathcal{F}$ and J can lead to such a guarantee, but we do not discuss the statistical properties any further in this paper.

We have to specify $\mathcal{F}$ and J. We focus on the choice of $\mathcal{F}$ that is closely related to an RKHS with kernel κ.

2.2 Kernels, RKHS, and the Space of Sum of Squares functions in an RKHS

We consider a bounded measurable kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and its corresponding RKHS $\mathcal{H}$. This RKHS has an associated feature map $\phi : \mathcal{X} \to \mathcal{H}$, defined as $\phi(x) = (\phi_i(x))_{i \in \mathcal{I}}$ with $\phi_i : \mathcal{X} \to \mathbb{R}$ with $\mathcal{I}$ being an index set, e.g., $\mathcal{I} = \{1, 2, 3, \dots\}$. We set $d = |\mathcal{I}|$ with the understanding that the index set might be countably infinite, in which case we set $d = \infty$.

The space of Sum of Squares (SoS) w.r.t. ϕ is defined as

$$\mathcal{S} \triangleq \left\{ x \mapsto \phi^\top(x) Q \phi(x) \ : \ Q \succeq 0 \right\}.$$

Here $Q \succeq 0$ means that the matrix Q is positive semidefinite (PSD). Evidently, any function $f \in \mathcal{S}$ is nonnegative. We may use $\mathcal{S}_\phi$ or $\mathcal{S}_\kappa$ to make the connection to the feature map or the kernel explicit.

For further development, we would like to have the RKHS machinery at our disposal. But notice that $\mathcal{S}$ is not a subspace of $\mathcal{H}$. We may, however, define another RKHS in which $\mathcal{S}$ is a subspace. We start by defining a new feature map $\psi : \mathcal{X} \to \mathcal{H}'_0$ as

$$\psi(x) = (\phi_i(x) \cdot \phi_j(x))_{i,j \in \mathcal{I}}.$$

Given this feature map, we define a kernel $\kappa' : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as usual:

$$\begin{aligned} \kappa'(x, y) \triangleq \langle \psi(x), \psi(y) \rangle_{\mathcal{H}'_0} &= \sum_{i,j \in \mathcal{I}} \phi_i(x)\phi_j(x)\ \phi_i(y)\phi_j(y) = \\ &= \sum_{i \in \mathcal{I}} \phi_i(x)\phi_i(y) \sum_{j \in \mathcal{I}} \phi_j(x)\phi_j(y) = \langle \phi(x), \phi(y) \rangle^2 = \kappa^2(x, y). \end{aligned} \tag{2}$$

Therefore there exists a unique RKHS $\mathcal{H}'$ for which κ' is a reproducing kernel. In tensor notation, $\mathcal{H}' = \mathcal{H} \otimes \mathcal{H}$.

Let $I(i, j)$ be the mapping from $\mathcal{I} \times \mathcal{I}$ to the corresponding index of ψ, that is, $\psi_{I(i,j)} = \phi_i \phi_j$. We then have

$$\begin{aligned} \mathcal{S} = \left\{ x \mapsto \phi^\top(x) Q \phi(x) \ : \ Q \succeq 0 \right\} &= \left\{ x \mapsto \sum_{i,j \in \mathcal{I}} Q_{ij} \phi_i(x)\phi_j(x) \ : \ Q \succeq 0 \right\} \\ &= \left\{ x \mapsto \sum_{i,j \in \mathcal{I}} Q_{ij} \psi_{I(i,j)}(x) \ : \ Q \succeq 0 \right\} \\ &\subset \left\{ x \mapsto \sum_{i,j \in \mathcal{I}} Q_{ij} \psi_{I(i,j)}(x) \ : \ Q \in \mathbb{R}^{d \times d} \right\} = \mathcal{H}'. \end{aligned} \tag{3}$$

The function space $\mathcal{S}$ is the space of SoS functions defined using the feature map ϕ corresponding to the RKHS κ, and is a subset of $\mathcal{H}'$. We call it a Kernel SoS space. Therefore, we state our goal of solving the optimization problem (1) as this particular instance of that general optimization:

$$\hat{f} \leftarrow \operatorname*{argmin}_{f \in \mathcal{S}} L_n(f) + \lambda \|f\|^2_{\mathcal{H}'}. \tag{4}$$

When d is moderately small, we can explicitly construct $\mathcal{S}$ as it is a subset of d^2-dimensional linear space defined by features ψ. When d is large, however, we use a representer theorem, to be proved in the next section, to provide a computationally feasible algorithm.

2.3 Representer Theorem for Kernel SoS

In this section we state a representer theorem for Kernel SoS. Let us first define some function spaces. For a particular set $\{X_i\}_{i=1}^n$, we define $\mathcal{H}_n$, $\mathcal{H}'_n$ and $\mathcal{S}_n$:

$$\mathcal{H}_n = \left\{ x \mapsto \sum_{l=1}^{n} \alpha_l \mathrm{K}(x, X_i) \;:\; \alpha \in \mathbb{R}^n \right\},$$
$$\mathcal{H}'_n = \left\{ x \mapsto \sum_{l=1}^{n} \alpha_l \mathrm{K}'(x, X_i) \;:\; \alpha \in \mathbb{R}^n \right\},$$
$$\mathcal{S}_n = \mathcal{H}'_n \cap \mathcal{S}.$$

The following result is similar in spirit to Theorem 5.5 (Representer Theorem) of Steinwart and Christmann [15], but is modified for Kernel SoS. Its proof requires appropriate modifications of Lemma 5.1, Theorem 5.2, and Theorem 5.5 of Steinwart and Christmann [15].

Theorem 1 (Representer Theorem)**.** *Let L_n be a convex empirical loss function as defined before. Then for all $\lambda > 0$, there exists a unique solution $\hat{f} \in \mathcal{S}$ satisfying*

$$L_n(\hat{f}) + \lambda \left\| \hat{f} \right\|_{\mathcal{H}'}^2 = \inf_{f \in \mathcal{S}} L_n(f) + \lambda \|f\|_{\mathcal{H}'}^2 .$$

Moreover, $\hat{f} \in \mathcal{S}_n$.

Proof. (Existence) By a argument similar to the first paragraph of the proof of Theorem 5.2 of Steinwart and Christmann [15], one can show that the mapping $f \mapsto g_f \triangleq L_n(f) + \lambda \|f\|_{\mathcal{H}}^2$ is convex and continuous.

Define the set

$$\mathcal{A} = \left\{ f \in \mathcal{S} \;:\; L_n(f) + \lambda \|f\|_{\mathcal{H}'}^2 \leq \underbrace{L_n(0) + \lambda \|0\|_{\mathcal{H}'}^2}_{= L_n(0)} \right\}.$$

Because of the continuity of the mapping g_f, the set $\mathcal{A}$ is closed. (Moreover, $\mathcal{S}$ is a convex cone and $\left\{ f \in \mathcal{H}' \;:\; L_n(f) + \lambda \|f\|_{\mathcal{H}'}^2 \leq L_n(0) \right\}$ is a convex set, so their intersection, which is $\mathcal{A}$, is convex too. Also $f = 0 \in \mathcal{A}$ (the zero function belongs to $\mathcal{S}$), so $\mathcal{A}$ is non-empty.

From the definition of this set, we have that for any $f \in \mathcal{A}$, it holds that $\|f\|_{\mathcal{H}'}^2 \leq \frac{L_n(0)}{\lambda}$. Therefore, $\mathcal{A}$ is a subset of a closed ball in $\mathcal{H}'$ with radius $\sqrt{\frac{L_n(0)}{\lambda}}$, i.e., $\mathcal{A} \subset B_{\mathcal{H}'}\left(\sqrt{\frac{L_n(0)}{\lambda}}\right)$. Clearly this ball is bounded. To summarize, $\mathcal{A}$ is a closed, convex, non-empty, and bounded set.

As $0 \in \mathcal{S}$, we can benefit from the optimizer property to have

$$\inf_{f \in \mathcal{S}} L_n(f) + \lambda \|f\|_{\mathcal{H}'}^2 \leq L_n(0) + \lambda \|0\|_{\mathcal{H}'}^2 = L_n(0).$$

The right-hand side (RHS) is the same as the upper bound constraint in the definition of $\mathcal{A}$. So restricting the search for the infimum from $\mathcal{S}$ to its intersection with $\mathcal{A}$, that is $\mathcal{S} \cap \mathcal{A} = \mathcal{A}$, does not change the value of the infimum. Therefore,

$$\inf_{f \in \mathcal{S}} L_n(f) + \lambda \|f\|_{\mathcal{H}'}^2 = \inf_{f \in \mathcal{A}} L_n(f) + \lambda \|f\|_{\mathcal{H}'}^2 . \tag{5}$$

As $\mathcal{A}$ is a non-empty closed bounded convex set, by Theorem 8.4 of Rockafellar [12], the recession cone $0^+\mathcal{A}$ consists of zero vector only.

The mapping g_f is a closed (because it is continuous, cf. p. 52 of Rockafellar [12]) proper convex function. As the direction of recession of $\mathcal{A}$ is only zero, Theorem 27.3 of Rockafellar [12] shows that $\inf_{f \in \mathcal{A}} L_n(f) + \lambda \|f\|_{\mathcal{H}'}^2$ attains its infimum over $\mathcal{A}$. Let us call it $\hat{f} \in \mathcal{A}$. The equality (5) indicates that $\hat{f}$ is an infimum of g_f over $\mathcal{S}$ too, so we have proven the existence of a solution.

We can also define $\mathcal{A}_n = \left\{ f \in \mathcal{S}_n \,:\, L_n(f) + \lambda \|f\|^2_{\mathcal{H}'} \leq L_n(0) \right\}$ and follow the same arguments to show that $\inf_{f\in\mathcal{S}_n} L_n(f) + \lambda \|f\|^2_{\mathcal{H}}$ has a solution that is attained in $\mathcal{A}_n$.

(Uniqueness) Suppose that there exist $f_1, f_2 \in \mathcal{S}$ that are two distinct minimizers of $\inf_{f\in\mathcal{S}} L_n(f) + \lambda \|f\|^2_{\mathcal{H}'}$. Define $\bar{f} = \frac{1}{2}(f_1 + f_2)$. Note that all $f_1, f_2, \bar{f} \in \mathcal{H}'$. By Lemma A.5.9 of Steinwart and Christmann [15], it holds that

$$\frac{1}{2}(\|f_1\|^2_{\mathcal{H}'} + \|f_2\|^2_{\mathcal{H}'}) = \left\| \frac{f_1 + f_2}{2} \right\|^2_{\mathcal{H}'} + \left\| \frac{f_1 - f_2}{2} \right\|^2_{\mathcal{H}'}.$$

As $f_1 \neq f_2$, by the property of the norm it holds that $\|\frac{f_1 - f_2}{2}\|^2_{\mathcal{H}'} > 0$, so

$$\left\|\bar{f}\right\|^2_{\mathcal{H}'} < \frac{1}{2}\left[\|f_1\|^2_{\mathcal{H}'} + \|f_2\|^2_{\mathcal{H}'}\right]. \tag{6}$$

By the convexity of the loss L_n, we also have

$$L_n(\bar{f}) \leq \frac{1}{2}[L_n(f_1) + L_n(f_2)]. \tag{7}$$

Moreover, as $\mathcal{S}$ is convex, $\bar{f}$ is in $\mathcal{S}$ too, so $\bar{f}$ *can* be a minimizer. If that is the case, by (6) and (7) we have

$$L_n(\bar{f}) + \lambda \left\|\bar{f}\right\|^2_{\mathcal{H}'} < \frac{1}{2}\left[L_n(f_1) + L_n(f_2)\right] + \frac{1}{2}\left[\|f_1\|^2_{\mathcal{H}'} + \|f_2\|^2_{\mathcal{H}'}\right] = L_n(f_1) + \|f_1\|^2_{\mathcal{H}'},$$

which is a contrary to the fact that f_1 is a minimizer. Therefore, there cannot be more than one minimizer.

(Finite Representation) Let $\Pi_{\mathcal{S}_n} : \mathcal{S} \to \mathcal{S}_n$ be the projection operator from $\mathcal{S}$ to $\mathcal{S}_n$ w.r.t. the norm of $\mathcal{H}$. First we show that for any $f \in \mathcal{S}$, we have $L_n(f) = L_n(\Pi_{\mathcal{S}_n} f)$.

For any $f \in \mathcal{S} \subset \mathcal{H}'$, we can decompose it as $f = f_1 + f_2$ with $f_1 \in \mathcal{H}'_n$ and $f_2 \in \mathcal{H}'^{\perp}_n$, the orthogonal complement of $\mathcal{H}'_n$ in $\mathcal{H}'$.

For any $X_l \in \{X_1, \ldots, X_n\}$, by the reproducing property of $\mathcal{H}'$, we have

$$f_2(X_l) = \langle\, f_2\,,\, \kappa'(\cdot, X_l)\,\rangle = 0. \tag{8}$$

The terms contributing to $L_n(f)$ are in the form of $l(\cdot, f(X_l)) = l(\cdot, f_1(X_l) + f_2(X_l)) = l(\cdot, f_1(X_l))$, so only functions in the span of $\{\kappa'(X_l, \cdot)\}_{l=1}^n$ contribute to the loss L_n.

By construction of $\mathcal{S}$ (3), any function $f \in \mathcal{S}$ has a representation $v^\top Q v$ for some $v \in \mathcal{H}$ and $Q \succeq 0$. The projection operator $\Pi_{\mathcal{H}'_n} : \mathcal{H}' \to \mathcal{H}'_n = \mathcal{H}_n \otimes \mathcal{H}_n$ takes $w \in \mathcal{H}' = \mathcal{H} \otimes \mathcal{H}$ and returns a $w_n = v_n \otimes v_n \in \mathcal{H}_n \otimes \mathcal{H}_n$. Since v_n is the projection of v onto $\mathcal{H}_n$, one may write it as $v_n = Pv$ for an appropriate projection operator $P : \mathcal{H} \to \mathcal{H}_n$. So after projection of f, we have $\Pi_{\mathcal{H}'_n} f = v_n^\top Q v_n = v^\top P^\top Q P v$. Since $Q \succeq 0$, the $n \times n$ matrix $P^\top Q P$ is also PSD, so the projection $\Pi_{\mathcal{H}'_n} f$ also belongs to the PSD cone $\mathcal{S}_n = \mathcal{H}'_n \cap S$.

One can see that $f_n = \Pi_{\mathcal{H}'_n} f$ is the same as $f'_n = \Pi_{\mathcal{S}_n} f$. We just showed that f_n belongs to $\mathcal{S}_n$. Also by definition, f'_n belongs to $\mathcal{S}_n$ too. Suppose that $f_n \neq f'_n$. Since $\Pi_{\mathcal{H}'_n}$ is the orthogonal projection onto $\mathcal{H}'_n$, the value of $\|f_n - f\|_{\mathcal{H}'}$ is minimal. If $f'_n \neq f_n$, due to the uniqueness of orthogonal projection onto a linear subspace, we would have $\|f'_n - f\|_{\mathcal{H}'} > \|f_n - f\|_{\mathcal{H}'}$, which would contradict the assumption that f'_n is the projection of f onto $\mathcal{S}_n$.

Therefore for any $f \in \mathcal{S}$, $\Pi_{\mathcal{S}_n} f = \Pi_{\mathcal{H}'_n} f$. This alongside the fact that functions in the orthogonal complement of $\mathcal{H}'_n$ do not contribute to the loss L_n (8) show that

$$L_n(\Pi_{\mathcal{S}_n} f) = L_n(\Pi_{\mathcal{H}'_n} f) = L_n(f). \tag{9}$$

Since for any $f \in \mathcal{S}$ we have $\Pi_{\mathcal{S}_n} f = \Pi_{\mathcal{H}'_n} f$ and because the projection onto $\mathcal{H}'_n$ does not increase the norm (i.e., $\|\Pi_{\mathcal{H}'_n} f\|_{\mathcal{H}'} \leq \|f\|_{\mathcal{H}'}$), we have

$$\|\Pi_{\mathcal{S}_n} f\|_{\mathcal{H}'} \leq \|f\|_{\mathcal{H}'}. \tag{10}$$

Therefore, we get

$$\inf_{f\in\mathcal{S}} L_n(f) + \lambda \|f\|^2_{\mathcal{H}'} \leq \inf_{f\in\mathcal{S}_n} L_n(f) + \lambda \|f\|^2_{\mathcal{H}'} = \inf_{f\in\mathcal{S}} L_n(\Pi_{\mathcal{S}_n} f) + \lambda \|\Pi_{\mathcal{S}_n} f\|^2_{\mathcal{H}'} \leq \inf_{f\in\mathcal{S}} L_n(f) + \lambda \|f\|^2_{\mathcal{H}'}.$$

The first inequality is because the infimum in $\mathcal{S}_n$ cannot be smaller than the infimum in $\mathcal{S}$ as $\mathcal{S}_n \subset \mathcal{S}$. The first equality is due to the definition of the projection operator. The second inequality is the result of (9) and (10).

Since the left-hand side (LHS) is the same as the RHS, it must be $\inf_{f\in\mathcal{S}} L_n(f) + \lambda \|f\|^2_{\mathcal{H}'} = \inf_{f\in\mathcal{S}_n} L_n(f) + \lambda \|f\|^2_{\mathcal{H}'}$, as desired. □

3 Algorithm

In this section we develop a computationally efficient approach to solve (4). From Theorem 1, we know that the solution can be written as $f(x) = \sum_{l=1}^n \alpha_l \mathrm{K}'(X_i, x)$ under the condition that the function has an SoS representation, that is, $f(x) = \phi(x)^\top Q \phi(x)$ for some $Q \succeq 0$. We have

$$\sum_{l=1}^n \alpha_l \mathrm{K}'(X_i, x) = \sum_{l=1}^n \alpha_l \left\langle \Psi(X_l), \Psi(x) \right\rangle = \sum_{l=1}^n \alpha_l \sum_{i,j\in\mathcal{I}} \phi_i(X_l)\phi_j(X_l)\ \phi_i(x)\phi_j(x)$$

$$= \sum_{i,j\in\mathcal{I}} \phi_i(x)\phi_j(x) \sum_{l=1}^n \alpha_l \phi_i(X_l)\phi_j(X_l) = \sum_{i,j\in\mathcal{I}} Q_{ij}\phi_i(x)\phi_j(x),$$

with $Q_{ij} = \sum_{l=1}^n \alpha_l\, \phi_i(X_l)\phi_j(X_l)$. We require that Q, which is a function of α, to be PSD.

The matrix $Q = [Q]_{ij}$ is $d \times d$, so the explicit computation of Q might not be feasible. The rank of Q, however, is at most n, the number of data points used in the optimization. We shortly see that one can enforce the same condition by requiring the positive semi-definiteness of a potentially much smaller $n \times n$ matrix.

Define a $d \times n$ matrix $\Phi = [\phi(X_1) \cdots \phi(X_n)]$ and an $n \times n$ diagonal matrix $A = \mathrm{diag}(\alpha_1, \ldots, \alpha_n)$. The matrix Q, defined above, can be written as $Q = \Phi A \Phi^\top$. The condition of Q being PSD is that all its eigenvalues should be nonnegative. For a square matrix B, denote $\mathrm{eig}(B)$ as the set of its non-zero eigenvalues. Because $\mathrm{eig}(BB^\top) = \mathrm{eig}(B^\top B)$, we have

$$\mathrm{eig}(Q) = \mathrm{eig}(\Phi A \Phi^\top) = \mathrm{eig}(\Phi \sqrt{A}\sqrt{A}\Phi^\top) = \mathrm{eig}(\sqrt{A}\underbrace{\Phi^\top\Phi}_{\triangleq G}\sqrt{A}) = \mathrm{eig}(GA).$$

Here $G = \Phi^\top\Phi$ is the Grammian matrix. We have $\Phi_{ij} = \sum_{k\in\mathcal{I}} \phi_k(X_i)\phi_k(X_j) = \left\langle \phi(X_i), \phi(X_j) \right\rangle_{\mathcal{H}} = \mathrm{K}(X_i, X_j)$. This means that even if the features are infinite dimensional, as long as we know their corresponding kernel function K, we can construct G.

Also note that for $f = \sum_{l=1}^n \alpha_l \mathrm{K}'(X_i, x)$, we have $\|f\|^2_{\mathcal{H}'} = \alpha^\top K' \alpha$ with $K'_{ij} = \mathrm{K}'(X_i, K_j) = \mathrm{K}^2(X_i, X_j)$ (2). In other words, $K' = G \odot G$, in which $\odot$ indicates entrywise (or Hadamard) matrix product. Therefore we get that

$$\inf_{f\in\mathcal{S}} L_n(f) + \lambda \|f\|^2_{\mathcal{H}'} = \inf_{\alpha\in\mathbb{R}^n} L_n\left(\sum_{l=1}^n \mathrm{K}'(X_l, \cdot)\alpha_l\right) + \lambda \alpha^\top K' \alpha.$$
$$\text{s.t.} \qquad G\,\mathrm{diag}(\alpha) \succeq 0$$

We next show how to formulate this optimization problem as a semidefinite program (SDP) (e.g., cf. Vandenberghe and Boyd [16]) when the loss function is the squared loss. If we denote the vector of target values by $Y = [Y_1, \ldots, Y_n]^\top$, we can write the optimization as

$$\inf_{\alpha\in\mathbb{R}^n} (K'\alpha - Y)^\top (K'\alpha - Y) + \lambda \alpha^\top K' \alpha$$
$$\text{s.t.} \qquad G\,\mathrm{diag}(\alpha) \succeq 0$$

First note that $G\,\mathrm{diag}(\alpha)$ is not symmetric. To convert this condition to the standard SDP formulation, which requires that the semi-definiteness condition to be imposed on symmetric matrices, note that $\mathrm{eig}(GA) = \mathrm{eig}(A^\top G^\top)$, and since both A and G are symmetric, it is equal to $\mathrm{eig}(AG)$. Therefore, $GA \succeq 0$ if and only if $GA + AG \succeq 0$.

Let us write the objective as $\alpha^\top M\alpha + 2\alpha^\top N$ with $M = K'^\top K' + \lambda K'$ and $N = -2K'^\top Y$ (we ignore the $Y^\top Y$ term, which does not affect the minimizer). Let L be such that $L^\top L = M$, e.g., its Cholesky factorization. Note that $\min_\alpha \alpha^\top M\alpha + 2\alpha^\top N$ is equivalent to

$$\begin{aligned} &\min_{t,\alpha\in\mathbb{R}^n} \; t \\ &\text{s.t. } \alpha^\top M\alpha + 2\alpha^\top N - t \leq 0. \end{aligned}$$

Moreover, having $t + 2\alpha^\top N - \alpha^\top M\alpha \geq 0$, by the Schur complement condition, is equivalent to requiring the positive semi-definiteness of

$$\begin{bmatrix} \mathbf{I}_{n\times n} & L\alpha \\ \alpha^\top L^\top & t + 2\alpha^\top N \end{bmatrix}.$$

Altogether we obtain the following SDP:

$$\begin{aligned} &\min_{t,\alpha\in\mathbb{R}^n} \; t \\ &\text{s.t. } \begin{bmatrix} \mathbf{I}_{n\times n} & L\alpha & 0_{n\times n} \\ \alpha^\top L^\top & t + 2\alpha^\top K'^\top Y & 0_{1\times n} \\ 0_{n\times n} & 0_{n\times 1} & G\,\mathrm{diag}(\alpha) + \mathrm{diag}(\alpha)G \end{bmatrix}_{2n+1\times 2n+1} \succeq 0. \end{aligned} \tag{11}$$

4 Illustrations

We illustrate the proposed algorithm through some simple illustrations. To solve the SDP formulation (11), we use CVX, a package for solving convex programs [2, 4].

We choose the function $f_c(x) = \sin(x) + \cos(3x) - (\frac{x}{2})^2 + c$ defined on $\mathcal{X} = [-5, +5]$ with the value of c to be specified. For the input of the training data, we use a uniform grid of 50 data points on $\mathcal{X}$. For its output, we first consider the noiseless case, i.e., $Y_i = f_c(X_i)$. Later we study the effect of noise too.

We first study learning a positive function using noiseless samples (Figure 1). By the choice of $c = 8$, the function $f_8(x)$ is positive on its domain. We choose the kernel function to be the mixture of squared exponential (or Gaussian) with the bandwidth of σ and the Dirac's delta function:[1]

$$\kappa(x_1, x_2) = \exp\left(-\frac{|x_1 - x_2|^2}{2\sigma^2}\right) + 0.01\delta(x_1 - x_2).$$

We present the results with $\sigma \in \{0.25, 0.75\}$. These values are chosen to show two different types of behaviour; they are not chosen to optimize the loss. In all presented results, the regularization coefficient is $\lambda = 0.01$. One can see from Figure 1 that the estimator with $\sigma = 0.25$ closely overlaps the target function. On the other hand, the estimator with $\sigma = 0.75$ is slightly oversmoothing.

We also tried smaller values of σ. When σ becomes too small (smaller than the minimum distance between two adjacent points, which is 0.2 in our experiments), the estimator severely overfits to data points. This is reflected by having a Grammian matrix with off-diagonal terms that are much smaller than the diagonal terms. Of course this is expected from kernel-based methods.

Figure 2 shows the result of learning function f_3, which takes both positive and negative values in its domain. As before, the output data is noiseless. The learned functions are regularized L_2-projection of the target function onto SoS_κ spaces. They approximate the positive parts of f_3, but fade away whenever f_3 becomes negative. Noticeably the shapes of the learned functions are not

[1] The main reason for having the delta function in the used kernel was numerical stability. The squared exponential kernels with large value of σ (even $\sigma \approx 0.5$ in our experiments) lead to Grammian matrices that are too ill-conditioned for CVX to converge.

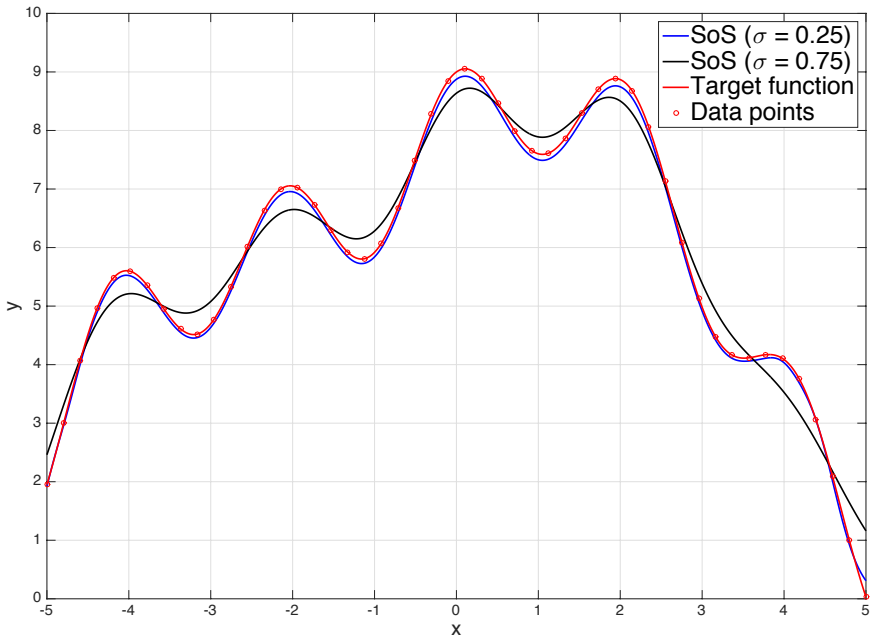

Figure 1: Learning a positive function from 50 noiseless samples. The estimator that uses SoS_K with $\sigma = 0.25$ (blue curve) almost overlaps the target function (red curve). The estimator with $\sigma = 0.75$ is oversmoothed.

the same as those learned in the previous illustration (with the shifting down and truncation at the level of 0). This difference is more prominent for $\sigma = 0.75$. Whereas the estimator with the kernel parameter $\sigma = 0.75$ in Figure 1 is oversmoothed only slightly, the same kernel parameter leads to gross oversmoothing in this case.

Figure 3 depicts the result of learning f_3 with noisy output data. In this experiment, $Y_i = f_3(X_i) + \varepsilon_i$ with ε_i being an i.i.d. sample from a zero-mean normal distribution with standard deviation of 0.5. We can see that as in the previous illustration, the estimator with $\sigma = 0.75$ is oversmoothed, while the estimator with $\sigma = 0.25$ is slightly overfitting. One can perform some kind of model selection to choose the best value of σ and λ.

5 Future Work

Several interesting questions remain to be answered. One of them is studying function approximation properties of Kernel SoS. We know that RKHS with universal kernels are dense in the space of continuous function w.r.t. the supremum norm. Can we show a similar result for Kernel SoS in the space of positive functions? We have not studies the statistical properties of such an estimator. Proving an estimation error upper bound is a future research topic. Designing more computationally efficient algorithms to solve (11) than using a generic SDP solver is necessary to make this algorithm practical.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337 – 404, May 1950. 1

[2] CVX Research, Inc. CVX: Matlab software for disciplined convex programming, version 2.0. http://cvxr.com/cvx, August 2012. 7

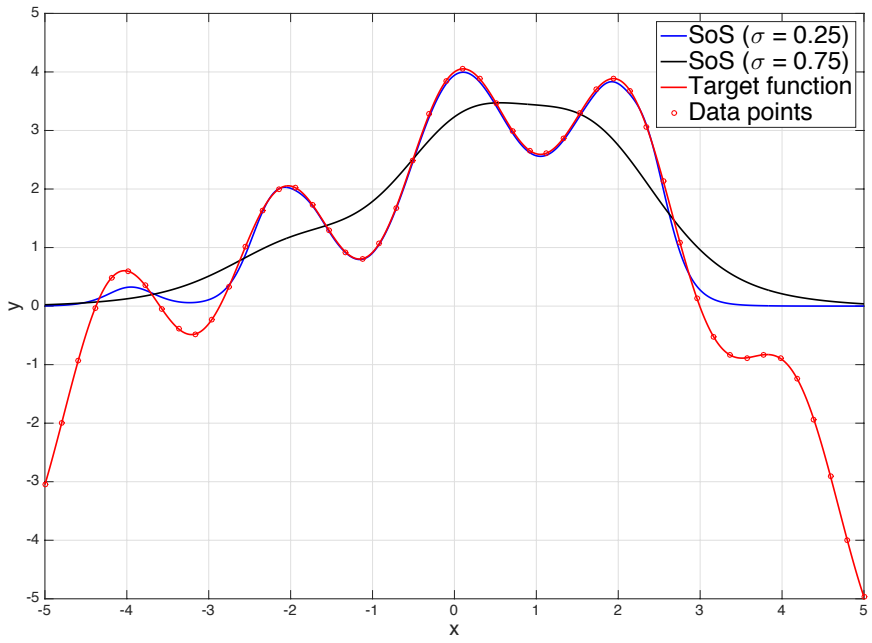

Figure 2: Learning a function that has both positive and negative regions from 50 noiseless samples.

[3] Mehdi Ghasemi. *Polynomial Optimization and the Moment Problem*. PhD thesis, Department of Mathematics and Statistics, University of Saskatchewan, 2012. 1, 2

[4] Michael C. Grant and Stephen P. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. 7

[5] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001. 1

[6] Jean B Lasserre and Tim Netzer. SOS approximations of nonnegative polynomials via simple high degree perturbations. *Mathematische Zeitschrift*, 256(1):99–112, 2007. 2

[7] Alessandro Magnani, Sanjay Lall, and Stephen Boyd. Tractable fitting with convex polynomials via sum-of-squares. In *IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC)*, pages 1672–1677. IEEE, 2005. 1

[8] Yurii Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, pages 405–440. Springer, 2000. 1

[9] Pablo A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000. 1

[10] Pablo A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003. 1

[11] Pablo A. Parrilo. MIT 6.256 - algebraic techniques and semidefinite optimization. Course Notes, March 2014. 2

[12] R Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1997. 4

[13] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002. 1

[14] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *COLT '01/EuroCOLT '01: Proceedings of the 14th Annual Conference on Computational*

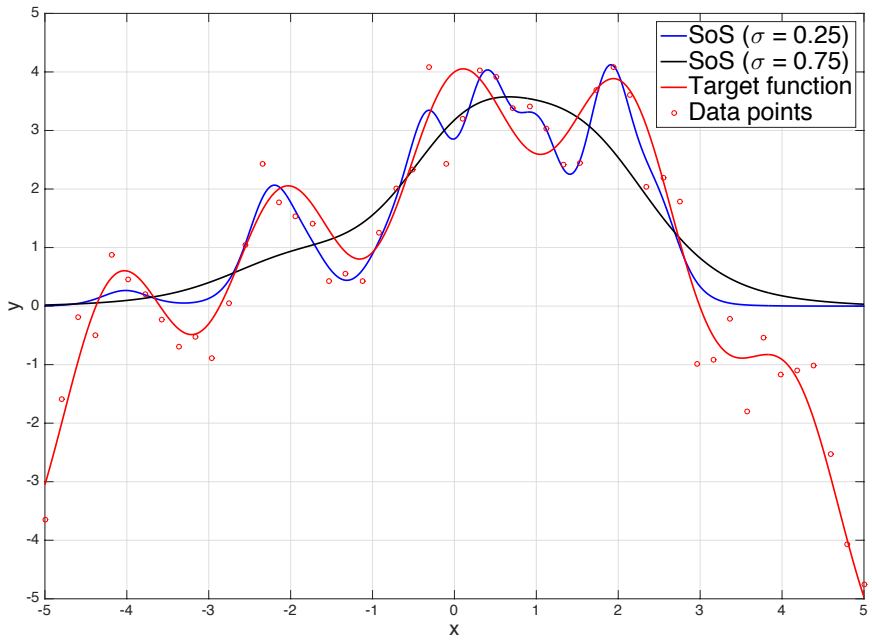

Figure 3: Learning a function that takes both positive and negative values from 50 noisy samples.

Learning Theory and and 5th European Conference on Computational Learning Theory, pages 416–426. Springer-Verlag, 2001. 2

[15] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008. 1, 2, 4, 5

[16] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1): 49–95, 1996. 6