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 (RKHS) 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 an example 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 [7], Parrilo [8], Lasserre [5], Parrilo [9], Ghasemi [3]. SoS has already several applications, e.g., in control theory [8], but as far as we know it has rarely used for machine learning problems (one exception is the work by Magnani et al. [6], 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 [10], Steinwart and Christmann [11].

2 Learning Positive Functions in an RKHS

We are given a dataset $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$ with $(X_i, Y_i) \sim \mu$. Denote the pointwise loss function by $l : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to [0, \infty)$. We consider convex losses. Given this pointwise loss l, for a function $f : \mathcal{X} \to \mathbb{R}$, we can define the expected loss $L(f) = \mathbb{E}_\mu \left[l(X, Y, f(X)) \right] = \int_{\mathcal{X} \times \mathcal{Y}} l(x, y, f(x)) \, \mathrm{d}\mu(x, y)$ and the empirical loss $L_n(f) = \frac{1}{n} \sum_{i=1}^n l(X_i, Y_i, f(X_i))$. Refer to Chapter 2 of Steinwart and Christmann [11] for more details.

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 $\phi_i : \mathcal{X} \to \mathbb{R}$

and $\mathcal{I}$ 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:

$$\begin{aligned} \kappa'(x, y) \triangleq \langle \psi(x), \psi(y) \rangle_{\mathcal{H}'_0} &= \sum_{i,j \in \mathcal{I}} \phi_i(x)\phi_j(x)\ \phi_i(y)\phi_j(y) = \\ &= \sum_{i \in \mathcal{I}} \phi_i(x)\phi_i(y) \sum_{j \in \mathcal{I}} \phi_j(x)\phi_j(y) = \langle \phi(x), \phi(y) \rangle^2 = \kappa^2(x, y). \end{aligned} \tag{1}$$

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

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

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.

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 stated in the next section, to provide a computationally feasible algorithm.

2.1 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 $\mathcal{S}_n$:

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

The following result is similar in spirit to Theorem 5.5 (Representer Theorem) of Steinwart and Christmann [11], 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 [11]. Due to space limitation, we defer the proof to the longer version of this paper.

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

3 Algorithm

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

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

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

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

We next show how to formulate this optimization problem as a semidefinite program (SDP) (e.g., cf. Vandenberghe and Boyd [12]) 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

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

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 $GA + AG \succeq 0$.

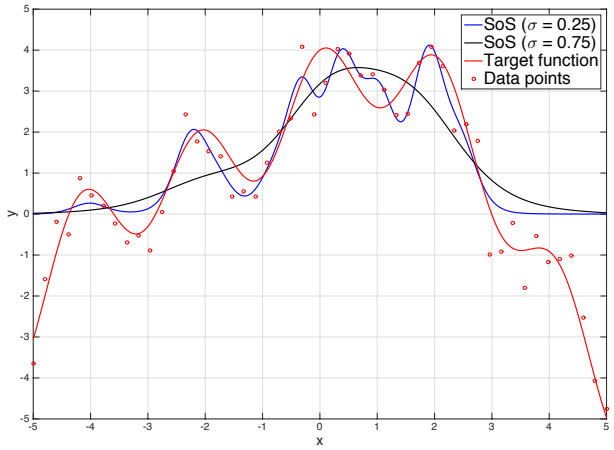

Figure 1: Learning a function that takes both positive and negative values from 50 noisy samples.

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:

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

4 Illustrations

We briefly illustrate the proposed algorithm in the task of learning $f_c(x) = \sin(x)+\cos(3x)-(\frac{x}{2})^2+3$ defined on $\mathcal{X} = [-5, +5]$. This function takes both positive and negative values in its domain. We sample 50 noisy data points and solve the SDP formulation (2) by CVX [2, 4]. We choose $\kappa(x_1, x_2) = \exp(-\frac{|x_1-x_2|^2}{2\sigma^2}) + 0.01\delta(x_1 - x_2)$. We present the results for both $\sigma \in \{0.25, 0.75\}$ and with the choice of regularization coefficient $\lambda = 0.01$. More details are available in the longer version of this paper.

Figure 1 depicts the result when the noise in the output variables has a zero-mean normal distribution with the standard deviation of 0.5. We can see that the estimator with $\sigma = 0.75$ is oversmoothed, while the estimator with $\sigma = 0.25$ is slightly overfitting. Both of them, however, are non-negative, as expected. Finally note that one can perform a standard 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 (2) 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. 4

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

[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. 4

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

[6] 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

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

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

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

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

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

[12] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1): 49–95, 1996. 3