POLYNOMIAL ARGMIN FOR RECOVERY AND APPROXIMATION OF MULTIVARIATE DISCONTINUOUS FUNCTIONS

Didier Henrion, Milan Korda, Jean-Bernard Lasserre

POLYNOMIAL ARGMIN FOR RECOVERY AND APPROXIMATION OF MULTIVARIATE DISCONTINUOUS FUNCTIONS

DIDIER HENRION[1,2], MILAN KORDA[1,2], JEAN BERNARD LASSERRE[1,3]

Abstract. We propose to approximate a (possibly discontinuous) multivariate function $f(\mathbf{x})$ on a compact set by the partial minimizer $\arg\min_y p(\mathbf{x}, y)$ of an appropriate polynomial p whose construction can be cast in a univariate sum of squares (SOS) framework, resulting in a highly structured convex semidefinite program. In a number of non-trivial cases (e.g. when f is a piecewise polynomial) we prove that the approximation is exact with a low-degree polynomial p. Our approach has three distinguishing features: (i) It is mesh-free and does not require the knowledge of the discontinuity locations. (ii) It is model-free in the sense that we only assume that the function to be approximated is available through samples (point evaluations). (iii) The size of the semidefinite program is independent of the ambient dimension and depends linearly on the number of samples. We also analyze the sample complexity of the approach, proving a generalization error bound in a probabilistic setting. This allows for a comparison with machine learning approaches.

1. Introduction

Approximation of discontinuous functions in multiple dimensions is a notoriously difficult problem and a scientific challenge. A common strategy (e.g. described in [23]) which works well in the univariate setting (and is implemented for example in the `chebfun` package [8]) consists of the following steps : 1) detect the discontinuity locations and split the domain into a disjoint union of regions where the function is continuous, and 2) construct approximations of the continuous pieces on each region. However, in the multivariate case this strategy is very challenging to implement since the discontinuity set may have a positive dimension (see, e.g, [13] where an algorithm for detecting discontinuities in two dimensions is proposed). Numerical difficulties faced with approximating multivariate functions are illustrated in the example sections of the paper.

A typical and important application is concerned with classification in data analysis and supervised learning, where powerful deep learning methods have obtained impressive results and success stories. However, such powerful methods still have some limitations (even for learning continuous functions, let alone discontinuous functions). Indeed for instance and quoting [3], *" Despite many results that establish the existence of Neural Nets (NNs) with excellent approximation properties, algorithms that can compute these NNs only exist in specific cases."* That is, no training algorithm can obtain them in the general case. For an interesting discussion about such limits (instability, accuracy, etc.) the interested reader is referred to [2, 3] and references therein. For learning discontinuous functions by neural networks, [12] proposes a tailored architecture; however this approach requires the knowledge of the discontinuities locations and is limited to univariate problems.

[1]CNRS; LAAS; Université de Toulouse, 7 avenue du colonel Roche, F-31400 Toulouse, France.

[2]Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, CZ-16626 Prague, Czechia.

[3]Institute of Mathematics; Université de Toulouse, 118 route de Narbonne, F-31062 Toulouse, France.

The research of M. Korda and J . B. Lasserre research is partly supported by AI Interdisciplinary Institute ANITI funding, through the French "Investing for the Future PIA3" program under the Grant agreement n°ANR-19-PI3A-0004. This research is also part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. J.B. Lasserre also acknowledges support from ANR-NuSCAP-20-CE48-0014.

In this paper we provide an alternative approximation technique aimed at dealing with such discontinuities and the Gibbs phenomenon, which are large oscillations of the approximation near the discontinuity points, see e.g. [21, Chapter 9]. In particular, we show that our class of approximants can model exactly multivariate piecewise polynomial functions and can approximate with arbitrary accuracy other discontinuous functions.

Prior work. The idea behind our new approximant is a non-trivial extension of our previous work [15] that has provided an approximant which is the argument of the partial minimum (argmin), of a sum of squares (SOS) of polynomials, in fact the reciprocal of the Christoffel function of a measure, ideally supported on the graph of the function to recover. As explained in [15], this recovery procedure can be seen as a non-standard application of the Christoffel-Darboux kernel. Importantly, being in a class of functions much larger than polynomials, such an approximant is able to approximate some discontinuous functions much better than polynomials can.

Remarkably, in non-trivial examples, recovery is possible without oscillations and Gibbs phenomenon usually encountered in several more standard approaches. In this respect the reader is referred to the detailed discussion in [22] on kernel variants (e.g Féjer or Jackson kernels) to attenuate the Gibbs phenomenon encountered with polynomial approximations.

This paper can be considered both as a follow-up and a non trivial extension of [15].

Contribution. Inspired by [15] we introduce a new class of approximants for a possibly discontinuous function f from $\mathbf{X} \subset \mathbb{R}^n$ to $\mathbf{Y} \subset \mathbb{R}$. We propose to approximate f by the *polynomial argmin*

$$\mathbf{x} \mapsto \hat{f}(\mathbf{x}) := \min\{ \arg\min_{y \in \mathbf{Y}} p(\mathbf{x}, y) \}, \quad \mathbf{x} \in \mathbf{X}, \tag{1.1}$$

where $\mathbf{Y} \supset f(\mathbf{X})$ and $p \in \mathbb{R}[\mathbf{x}, y]$ is a polynomial in $(\mathbf{x}, y)$.

The main features of our approach can be summarized as follows.

- The approach is mesh-free and does not require the knowledge of the discontinuities locations.
- The approach is not limited to univariate functions or tensor products thereof.
- The approach is model-free, working only with the samples of the unknown function.
- The approximant is constructed using convex optimization (semidefinite programming) with size dependent (linearly) on the number of samples but not on the ambient dimension.
- The approximant is simple to evaluate as it is the argmin of a univariate polynomial.
- We provide a generalization error analysis in a probabilistic setting.

We believe that these features make the approach a unique and extremely promising tool with a wide variety of applications in data analysis. This is corroborated by a numerical evidence where we observe a remarkable performance on a range of examples. We also provide a solid theoretical underpinning of the method but leave some questions open, including the optimal rate of convergence of the argmin approximant.

A novelty and distinguishing feature with respect to [15] is that the polynomial p in (1.1) is *not* restricted to be the reciprocal of the Christoffel function associated with the measure supported on the graph of f. Indeed, our set of potential candidate polynomials p is now a suitably chosen subspace of $\mathbb{R}[\mathbf{x}, y]$, which is much larger than the set considered in [15].

While our original motivation for introducing the polynomial argmin approximant was to recover the solution of a nonlinear partial differential equation from the knowledge of its approximate moments [14], this strategy was made rigorous and generalized to graph recovery from moments [15]. Then it was extended to cope with partial moment information [11]. Following our initial work, this argmin strategy was called "implicit model" and used in robotics applications [10] where

the polynomial p in (1.1) is replaced by a continuous function computed by training, e.g. a neural network.

Outline. In the motivational Section 2 we show that our polynomial argmin strategy is already efficient in some non-trivial cases. For instance, it allows *exact recovery* when f is a polynomial, or an algebraic function, or a piecewise polynomial. In particular it can model exactly the indicator function of the unit disk, sharply contrasting with classical approximants.

However, exact recovery by a polynomial argmin cannot be guaranteed in general, and in Section 3 we provide a numerical scheme to obtain the polynomial p in (1.1) when knowledge on f is only though its finitely many values on a sample of points $(\mathbf{x}(i))_{i\in I} \subset \mathbf{X}$ (without a priori knowledge on its distribution). We reformulate our polynomial argmin strategy in the framework of sum of squares (SOS) positivity certificates so that finding p amounts to solving a semidefinite optimization problem whose size is controlled by the degree of p in y and the sample size.

Finally, in Section 4 we also provide a set of numerical experiments to evaluate the efficiency of our proposed polynomial argmin strategy on a sample of problems. We observe that it performs remarkably well to approximate challenging discontinuous functions already tested in [15], as well as non trival two-dimensional examples of functions whose set of discontinuities has positive dimension. We have also compared with machine learning methods based on neural networks. In all our experiments, the proposed method achieves far better accuracy with simpler representation of the approximant.

2. Motivating examples

The purpose of this section is to demonstrate the expressive power of the polynomial argmin. We do so by showing that for several classes of known functions, an exact (and sometimes obvious) representation is possible. Subsequently, in Section 3, we show how a polynomial argmin approximation can be found using convex optimization, provided that some finite sample of values of f is available. Importantly, all polynomials of the illustrative examples below are indeed optimal solutions of the convex program alluded to, which supports the claim that in some sense the optimization problem is well-founded as natural obvious solutions of those illustrative examples can be recovered.

2.1. Polynomials.

When $f \in \mathbb{R}[\mathbf{x}]$ is a given polynomial, we can choose in (1.1)

$$p(\mathbf{x}, y) := \frac{1}{2}y^2 - f(x)y.$$

Indeed, we observe that $\frac{dp}{dy} = y - f(x)$ and hence $y = f(x)$ is a stationary point. Since p is strictly convex in y, it follows that $y = f(x)$ is the global minimizer. The degree of p in x is equal to the degree of f whereas the degree in y is equal to two irrespective of f.

2.2. Algebraic functions.

An *algebraic function* f is such that $q_k(\mathbf{x}, f(\mathbf{x})) = 0$ for some given polynomials $q_k \in \mathbb{R}[\mathbf{x}, y]$, $k = 1, \ldots, m$.

If for each $\mathbf{x}$, $(\mathbf{x}, f(\mathbf{x}))$ is the unique common zero of the q_k then we can readily choose

$$p(\mathbf{x}, y) := \sum_{k=1}^{m} q_k(\mathbf{x}, y)^2$$

and observe that the degree of p in $\mathbf{x}$ and y is twice the maximal degree of the q_k in these variables.

	$(-\infty,-1)$	$(-1,-1/3)$	$(-1/3,0)$	$(0,1/3)$	$(1/3,1)$	$(1,\infty)$
p_1	max	max	min	min	min	min
p_2	min	min	max	max	max	max
p_3	min	min	max	max	min	min
argmin	p_3	p_2	p_2	p_1	p_1	p_3

TABLE 1. Nature of critical points for different intervals of variation of $g(\mathbf{x})$.

Note that *semi-algebraic functions*[1] can also be modeled like that, provided that the inequalities are incorporated in the definitions of the domain $\mathbf{X}$ and image sets Y.

Example 1. *The absolute value function can be expressed as*

$$|x| = \arg\min_{y\in Y}(x^2-y^2)^2$$

with $\mathbf{Y} := [0,1]$*, for all* $x \in \mathbf{X} := [-1,1]$*. Note that a more complicated degree 8 polynomial argmin model for the absolute value function was already described in* [15, Example 2].

2.3. **Piecewise polynomials.** Let $g, p_1, p_2 \in \mathbb{R}[\mathbf{x}]$ and consider the piecewise polynomial function

$$f(\mathbf{x}) := \begin{cases} p_1(\mathbf{x}) & \text{if} \quad g(\mathbf{x}) \in (0,1) \\ p_2(\mathbf{x}) & \text{if} \quad g(\mathbf{x}) \in (-1,0) \end{cases}$$

for $\mathbf{x} \in \mathbf{X} := \{\mathbf{x} : g(\mathbf{x}) \in (-1,0)\} \cup \{\mathbf{x} : g(\mathbf{x}) \in (0,1)\}$.

Lemma 2.1. *There exists* $p \in \mathbb{R}[\mathbf{x},y]$ *such that* $f(\mathbf{x}) = \arg\min_{y\in\mathbf{Y}} p(\mathbf{x},y)$ *for all* $\mathbf{x} \in \mathbf{X}$*, with* $\mathbf{Y} = \mathbb{R}$.

Proof: With $r \in \mathbb{R}[\mathbf{x}]$ and $q \in \mathbb{R}[\mathbf{x},y]$, let $p(\mathbf{x},y) := (y-p_1(\mathbf{x}))^2(y-p_2(\mathbf{x}))^2 + r(\mathbf{x})q(\mathbf{x},y)$ be such that $\partial q(\mathbf{x},y)/\partial y = 6(y-p_1(\mathbf{x}))(y-p_2(\mathbf{x}))$. For instance $q(\mathbf{x},y) := 2y^3 - 3(p_1(\mathbf{x}) + p_2(\mathbf{x}))y^2 + 6p_1(\mathbf{x})p_2(\mathbf{x})y$. Now observe that for each given $x \in \mathbf{X}$, $y \mapsto p(\mathbf{x},y)$ is a coercive quartic univariate polynomial, and hence it has at most two local minima. Letting $r(\mathbf{x}) := g(\mathbf{x})\,(p_1(\mathbf{x}) - p_2(\mathbf{x}))$, the gradient $\partial p(\mathbf{x},y)/\partial y$ vanishes at the critical points $p_1(\mathbf{x})$ resp. $p_2(\mathbf{x})$ resp. $p_3(\mathbf{x}) := (p_1(\mathbf{x})(1-3g(\mathbf{x})) + p_2(\mathbf{x})(1+3g(\mathbf{x}))/2$. At the critical points the Hessian $\partial^2 p(\mathbf{x},y)/\partial y^2$ is equal to $2(p_1(\mathbf{x})-p_2(\mathbf{x}))^2(1+3g(\mathbf{x}))$ resp. $2(p_1(\mathbf{x})-p_2(\mathbf{x}))^2(1-3g(\mathbf{x}))$ resp. $(p_1(\mathbf{x})-p_2(\mathbf{x}))^2(-1+3g(\mathbf{x}))(1+3g(\mathbf{x}))$. To compare the values at the critical points, we evaluate the differences $p(\mathbf{x},p_2(\mathbf{x})) - p(\mathbf{x},p_1(\mathbf{x})) = g(\mathbf{x})(p_1(\mathbf{x})-p_2(\mathbf{x}))^4$, $p(\mathbf{x},p_3(\mathbf{x})) - p(\mathbf{x},p_2(\mathbf{x})) = (1+g(\mathbf{x}))(1-3g(\mathbf{x}))^3(p_1(\mathbf{x}) - p_2(\mathbf{x}))^4/16$, $p(\mathbf{x},p_3(\mathbf{x})) - p(\mathbf{x},p_1(\mathbf{x})) = (-1+g(\mathbf{x}))(1+3g(\mathbf{x}))^3(p_1(\mathbf{x}) - p_2(\mathbf{x}))^4/16$. The proof follows by evaluating the signs of the Hessian and the differences at the critical points for $g(\mathbf{x})$ within the intervals $(-\infty,-1)$, $(-1,-1/3)$, $(-1/3,0)$, $(0,1/3)$, $(1/3,1)$, $(1,\infty)$, see Table 1. Note that p_3 is an auxiliary polynomial used in the proof rather than the value of f on a subset of $\mathbf{X}$.□

Example 2. *In* [15, Example 1] *a degree 8 polynomial argmin model was described for the sign function* $f(x)$ *which is equal to* -1 *if* $x \in (-1,0)$ *and* $+1$ *if* $x \in (0,1)$*. In this case* $g(x) = x$*,* $p_1(x) = +1$ *and* $p_2(x) = -1$ *and the construction of the proof of Lemma 2.1 yields the degree 4 polynomial argmin model* $p(x,y) = (y+1)^2(y-1)^2 + 4xy(y^2-3)$ *on* $\mathbf{X} = [-1,1]$*, see Figure 1.*

Note that an even simpler argmin model of the sign function on $\mathbf{X} = (-1,1)$ *is* $p(x,y) = -xy$ *for* $\mathbf{Y} = [-1,1]$*. Similarly to Example 1, the choice of domain and image sets* $\mathbf{X},\mathbf{Y}$ *plays here a key role.*

Example 3. *The indicator function of the bivariate unit disk* $\{\mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 \le 1\}$ *can be modeled exactly on* $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 \le 2\}$ *with the degree 5 polynomial argmin of* $p(\mathbf{x},y) = y^2((y-1)^2 + (1-x_1^2-x_2^2)(3-2y))$ *obtained by letting* $p_1(\mathbf{x}) = 0$*,* $p_2(\mathbf{x}) = 1$*,* $g(\mathbf{x}) = 1 - x_1^2 - x_2^2$ *in the construction of Lemma 2.1.*

[1]A semi-algebraic function is such that its graph is described by a finite union of a finite intersection of sets defined by polynomial equations and inequalities.

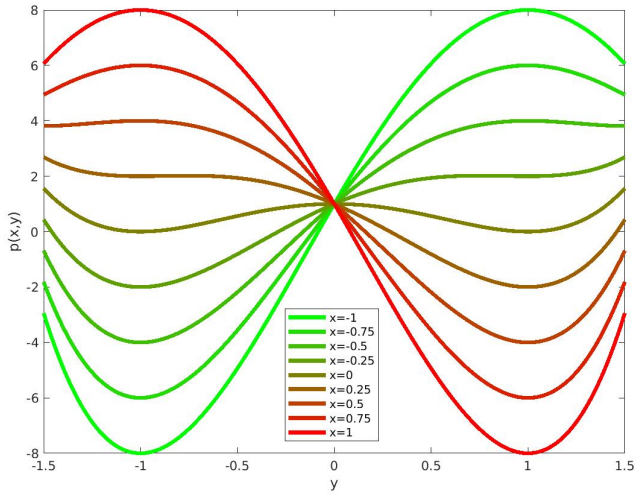

FIGURE 1. Degree 4 polynomial $p(x, y)$ whose argmin w.r.t. y models the sign function. Represented are univariate polynomials $y \mapsto p(x, y)$ for various given values of x. We observe that for positive values of x, the argmin is precisely $+1$ whereas for negative values of x, the argmin is precisely -1 as required in order to represent the function $\mathrm{sign}(x)$.

3. Sample-based formulation

3.1. **Notation and definitions.** Let $\mathbb{R}[\mathbf{x}, y]$ denote the ring of polynomials in the variables $\mathbf{x} \in \mathbb{R}^n$ and $y \in \mathbb{R}$, and $\mathbb{R}[\mathbf{x}, y]_d$ its subset of polynomials of total degree at most d. A polynomial p is a sum of squares (SOS) if it can be written as $\sum_k p_k^2$ for finitely many polynomials p_k. The convex cone of all SOS polynomials of degree at most d in the variable $\mathbf{x}$ is denoted by $\Sigma_d[\mathbf{x}]$.

3.2. **Nonnegativity of univariate polynomials on an interval.** Certificates of non-negativity of univariate polynomials will play a key role in our approach. The following result due to F. Lukács is classical (see, e.g., [20] for a discussion).

Theorem 3.1. *If $q \in \mathbb{R}[y]$ is a univariate polynomial of degree d nonnegative on the interval $[a, b] \subset \mathbb{R}$, then*

$$\begin{cases} q = \sigma_0 + \sigma_1(b-y)(y-a), & \sigma_0 \in \Sigma_d[y], \quad \sigma_1 \in \Sigma_{d-2}[y] \quad d \text{ even}, \\ q = \sigma_0(y-a) + \sigma_1(b-y), & \sigma_0 \in \Sigma_{d-1}[y], \ \sigma_1 \in \Sigma_{d-1}[y] \quad d \text{ odd}. \end{cases}$$

It is a simple observation that a polynomial σ belongs to Σ_d (for d even) if and only if there exists a positive semi-definite matrix $W \succeq 0$ such that $\sigma(y) = v_{d/2}(y) W v_{d/2}(y)$, where $v_{d/2}$ is a basis of $\mathbb{R}[y]_{d/2}$, e.g., $v_{d/2} = [1, y, y^2, \ldots, y^{d/2}]$. Therefore the nonnegativity of $q \in \mathbb{R}[y]$ on $[a, b]$ is *equivalent* to the feasibility of a semidefinite programming (SDP) problem. Observing that the conditions in Theorem 3.1 are affine in the coefficients of q, we conclude that one can also *optimize* over the set of polynomials (of fixed degree) nonnegative over $[a, b]$ using SDP.

3.3. **A numerical scheme.** Given a function $f : \mathbb{R}^n \to \mathbf{Y}$ with $\mathbf{Y} = [a, b] \subset \mathbb{R}$ sampled at points $(\mathbf{x}_i, y_i)_{i=1}^N$ with $y_i = f(\mathbf{x}_i)$, we wish to construct an approximation $\hat{f}_d$ of the form

$$\hat{f}(\mathbf{x}) \ = \ \arg\min_{y \in \mathbf{Y}} p(\mathbf{x}, y), \tag{3.1}$$

where $p \in \mathbb{R}[\mathbf{x}, y]$ is a polynomial to be determined. Throughout this section we assume that the argmin is unique. If this is not the case, a tiebreaker rule has to be applied (e.g., one can consider the min of the argmin). Since monomials not containing y do not influence the argmin, we use the parametrization of p as

$$p(\mathbf{x}, y) = \sum_{k=1}^{d_y} h_k(\mathbf{x}) y^k, \tag{3.2}$$

where $h_k \in \mathbb{R}[\mathbf{x}]_{d_x}$ are polynomials of total degree at most d_x to be determined. In order to find h_k, we propose to solve the following convex optimization problem parametrized by $d_x, d_y, d_\gamma \in \mathbb{N}$ and $\alpha > 0$:

(3.3)

$$\begin{aligned} \min_{\gamma \in \mathbb{R}[\mathbf{x},y]_{d_\gamma},\, (h_k \in \mathbb{R}[\mathbf{x}]_{d_x})_{k=1}^{d_y}} \quad & \int_{\mathbf{X}\times[a,b]} \gamma(x,y)\, d\mathbf{x}dy \\ \text{s.t.} \quad & p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2 \geq 0\,, \forall y \in [a,b], \quad i = 1, \ldots, N \\ & \gamma(\mathbf{x}_i, y_i) \geq 0\,, \quad i = 1, \ldots, N, \end{aligned}$$

where p is parametrized using h_k as in (3.2). The rationale behind (3.3) is as follows: If $\gamma(\mathbf{x}_i, y_i) = 0$, then $p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) \geq \alpha(y - y_i)^2$ for all $y \in \mathbf{Y} = [a, b]$ and hence $y_i = \arg\min_{y \in \mathbf{Y}} p(\mathbf{x}_i, y)$, which means $\hat{f}(\mathbf{x}_i) = y_i$. Therefore, if $\gamma(\mathbf{x}_i, y_i) = 0$ for all $i = 1, \ldots, N$ we get an exact interpolation of all data points. This, however, cannot be achieved in general in which case $\gamma(\mathbf{x}_i, y_i) > 0$ for some i; the polynomial γ therefore acts as a slack variable and is minimized in the objective function. An alternative to using a polynomial slack variable is to assign one slack variable $\gamma_i \in \mathbb{R}_+$ to each of the constraints; here we chose to use the polynomial slack variable in order to make the number of decision variables of (3.3) independent of the number of samples N, which facilitates the analysis of the generalization error in Section 3.4.

We also note that if the objective function cannot be evaluated in closed form (e.g., if the moments of the Lebesgue measure over $\mathbf{X} \times [a, b]$ are not known or too costly to compute), the integral can be replaced by an approximation computed from the available data

$$\frac{1}{N} \sum_{i=1}^{N} \gamma(\mathbf{x}_i, y_i). \tag{3.4}$$

The optimization problem (3.3) translates to SDP by equivalently reformulating the first constraint using Theorem 3.1. For brevity, we state it explicitly only for d_y even (the difference for d_y odd is the same as in Theorem 3.1):

(3.5)

$$\begin{aligned} \min_{\gamma \in \mathbb{R}[\mathbf{x},y]_{d_\gamma},\, (h_k \in \mathbb{R}[\mathbf{x}]_{d_x})_{k=1}^{d_y},\, (\sigma_{0,i}, \sigma_{1,i})_{i=1}^{N}} \quad & \int_{\mathbf{X}\times[a,b]} \gamma(x,y)\, d\mathbf{x}dy \\ & p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2 = \sigma_{0,i} + \sigma_{1,i}(b - y)(y - a), \quad i = 1, \ldots, N \\ & \sigma_{0,i} \in \Sigma_{d_y}[y], \sigma_{1,i} \in \Sigma_{d_y - 2}[y] \\ & \gamma(\mathbf{x}_i, \mathbf{y}_i) \geq 0, \quad i = 1, \ldots, N. \end{aligned}$$

The problem (3.5) readily translates to SDP and can be solved using off-the-shelf solvers such as MOSEK or SeDuMi.

Remark 3.2 (Range of f)**.** *If no information on the range of f is available, the first constraint of* (3.5) *can be replaced by*

$$p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2 = \sigma_{0,i} \quad i = 1, \ldots, N,$$

with $\sigma_{0,i} \in \Sigma_{d_y}$. This is equivalent to $p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2$ being nonnegative on $\mathbb{R}$ for each $i = 1, \ldots, N$.

We note that, up to rescaling of p and γ, the problems (3.3) and(3.5) are invariant with respect to the choice of the parameter $\alpha > 0$. We decided to include this parameter as a simple way to control the scaling of the coefficients of p and γ in the numerical implementation.

Importantly, in all the examples in Sections 2.1, 2.3 and 2.3 where an obvious solution p exists such that $f(\mathbf{x}) = \arg\min_{y \in \mathbf{Y}} p(\mathbf{x}, y)$, this solution p is an optimal solution of (3.3) and(3.5) with $\alpha = 0$. For some examples (e.g., the example of Section 2.1, Example 2 and Example 3), a rescaling of p is optimal with any positive value of α. In other words, our blind data driven approach is able to recover exactly obvious optimal solutions in nontrivial cases for which a Gibbs phenomenon would occur with more classical approaches.

The following result bounds the error on data points of any feasible solution to (3.3) and (3.5).

Lemma 3.3 (Error on data points)**.** *Let p, γ and α be feasible in (3.3) or (3.5) and let*

$$z_i \in \arg\min_{y \in [a,b]} p(\mathbf{x}_i, y).$$

Then we have

$$|z_i - y_i| \leq \sqrt{\frac{\gamma(\mathbf{x}_i, y_i)}{\alpha}}$$

and therefore

$$|\hat{f}(\mathbf{x}_i) - f(\mathbf{x}_i)| \leq \sqrt{\frac{\gamma(\mathbf{x}_i, y_i)}{\alpha}}$$

for $i = 1, \ldots, N$.

Proof. Let $(\mathbf{x}_i, y_i)$ be fixed. Observe that for any $z_i \in \arg\min_{y \in [a,b]} p(\mathbf{x}_i, y)$, we have $p(\mathbf{x}_i, z_i) - p(\mathbf{x}_i, y_i) \leq 0$. Therefore we have by the first constraint of (3.3) or (3.5)

$$\gamma(\mathbf{x}_i, y_i) - \alpha(z_i - y_i)^2 \geq p(\mathbf{x}_i, z_i) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(z_i - y_i)^2 \geq 0.$$

Therefore

$$|z_i - y_i| \leq \sqrt{\frac{\gamma(\mathbf{x}_i, y_i)}{\alpha}}.$$

The last statement of the lemma follows by the facts that $\hat{f}(\mathbf{x}_i) \in \arg\min_{y \in [a,b]} p(\mathbf{x}_i, y)$ with p optimal in (3.3) and (3.5) and $y_i = f(\mathbf{x}_i)$. □

3.4. **Generalization error.** In this section we study the generalization error of the argmin estimator in a probabilistic setting. We assume that the samples $\mathbf{x}_i$, $i = 1, \ldots, N$, are independent identically distributed, drawn from a probability distribution $\mathbb{P}$ on $\mathbf{X}$ that is unknown to us. We study the generalization error using the tools of scenario optimization, which allows for analysis with minimal underlying assumptions on f. The generalization bounds obtained have no explicit dependence on the dimension of the ambient space n and on regularity of f. They depend only the number of decision variables in (3.3), which, however, may depend implicitly on n.

We first observe that Problem 3.3 can be equivalently rewritten in the form

$$\begin{aligned} \min_{\theta \in \mathbb{R}^{n_\theta}} \quad & c^\top \theta \\ \text{s.t.} \quad & \inf_{y \in [a,b]} \{p_\theta(\mathbf{x}_i, y) - p_\theta(\mathbf{x}_i, y_i) + \gamma_\theta(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2\} \geq 0, \quad i = 1, \ldots, N \\ & \gamma(\mathbf{x}_i, y_i) \geq 0, \quad i = 1, \ldots, N, \end{aligned} \tag{3.6}$$

where $\theta \in \mathbb{R}^{n_\theta}$ gathers all the decision variables of (3.3), i.e., the coefficients of $(h_k)_{k=1}^{d_y}$ and γ, and $c \in \mathbb{R}^{n_\theta}$ is a constant vector such that $c^\top \theta = \int \gamma \, d\mathbf{x} dy$.

Problem (3.6) can be rewritten as

$$\begin{aligned} \min_{\theta\in\mathbb{R}^{n_\theta}} \quad & \theta^\top c \\ \text{s.t.} \quad & \theta \in \bigcap_{i=1}^{N} \Theta_i, \end{aligned} \tag{3.7}$$

where the set Θ_i is defined by

$$\Theta_i := \Big\{\theta \mid \inf_{y\in[a,b]} \{p_\theta(\mathbf{x}_i, y) - p_\theta(\mathbf{x}_i, y_i) + \gamma_\theta(\mathbf{x}_i, y_i) - \alpha(y-y_i)^2 \geq 0\},\ \gamma_\theta(\mathbf{x}_i, y_i) \geq 0\Big\}.$$

We also define

$$\Theta_{\mathbf{x}} := \Big\{\theta \mid \inf_{y\in[a,b]} \{p_\theta(\mathbf{x}, y) - p_\theta(\mathbf{x}, f(\mathbf{x})) + \gamma_\theta(\mathbf{x}, f(\mathbf{x})) - \alpha(y-f(\mathbf{x}))^2 \geq 0\},\ \gamma_\theta(\mathbf{x}, f(\mathbf{x})) \geq 0\Big\}.$$

We observe that (3.7) is the so-called scenario counterpart of the robust optimization problem

$$\begin{aligned} \min_{\theta\in\mathbb{R}^{n_\theta}} \quad & \theta^\top c \\ \text{s.t.} \quad & \theta \in \bigcap_{\mathbf{x}\in\mathbf{X}} \Theta_{\mathbf{x}}. \end{aligned} \tag{3.8}$$

In other words, the feasible set in (3.7) is a sub-sampled version of the feasible set of (3.8), where only N constraints, drawn independently, are enforced. Crucially, we remark that both Θ_i and $\Theta_{\mathbf{x}}$ are convex and so are the feasible sets of (3.7) and (3.8). With these notations, we can state the following theorem:

Theorem 3.4. *Let n_θ denote the number of decision variables in (3.7). Suppose that $\epsilon \in (0,1)$, $\delta \in (0,1)$ and $N \in \mathbb{N}$ are chosen such that*

$$\sum_{k=0}^{n_\theta-1} \binom{N}{k} \epsilon^k (1-\epsilon)^{N-k} \leq \delta \tag{3.9}$$

or

$$N \geq \frac{1}{\epsilon}\left(n_\theta - 1 + \ln\frac{1}{\delta} + \sqrt{2(n_\theta-1)\ln\frac{1}{\delta}}\right) \tag{3.10}$$

hold ((3.10) is a sufficient condition for (3.9)). Let (p,γ) be an optimal solution to (3.5) with N iid samples from a probability distribution $\mathbb{P}$ on $\mathbf{X}$ and denote

$$\gamma_{\max} := \sup_{(\mathbf{x},y)\in\mathbf{X}\times[a,b]} \gamma(\mathbf{x}, y).$$

Then with probability at least $1-\delta$ (taken over the sample $\mathbf{x}_1, \ldots, \mathbf{x}_N$ with joint distribution $\underbrace{\mathbb{P}\otimes\ldots\otimes\mathbb{P}}_{N \text{ times}}$), we have

$$\mathbb{P}\Big(\{\mathbf{x}\in\mathbf{X} : |\hat{f}(\mathbf{x}) - f(\mathbf{x})| \leq \sqrt{\frac{\gamma_{\max}}{\alpha}}\}\Big) > 1-\epsilon, \tag{3.11}$$

where $\hat{f}(\mathbf{x})$ is any measurable selection satisfying $\hat{f}(\mathbf{x}) \in \operatorname{Argmin}_{y\in\mathbf{Y}} p(\mathbf{x}, y)$.

Remark 3.5. *We remark that if the points $\mathbf{x}_i$ are sampled uniformly in $\mathbf{X}$, the probability in (3.11) is nothing but the normalized Lebesgue measure on $\mathbf{X}$.*

Proof of Theorem 3.4. Any optimal solution (p,γ) of (3.5) induces an optimal solution θ to (3.7) and vice-versa. Given such an optimal solution, [5, Theorem 1] asserts that if (3.9) holds, then with probability at least $1-\delta$

$$\mathbb{P}(\{\mathbf{x} \mid \theta \notin \Theta_{\mathbf{x}}\}) \leq \epsilon$$

or equivalently

$$\mathbb{P}(\{\mathbf{x} \mid \theta \in \Theta_{\mathbf{x}}\}) > 1 - \epsilon.$$

Using the definition of $\Theta_{\mathbf{x}}$, this implies that

$$\mathbb{P}(A) > 1 - \epsilon,$$

where

$$A := \{\mathbf{x} \in \mathbf{X} \mid \inf_{y \in [a,b]} \{p(\mathbf{x}, y) - p(\mathbf{x}, f(\mathbf{x})) + \gamma(\mathbf{x}, f(\mathbf{x})) - \alpha(y - f(\mathbf{x}))^2 \geq 0\}.$$

Given any $\mathbf{x} \in A$ and taking $\hat{f}(\mathbf{x}) \in \arg\min_{y \in [a,b]} p(\mathbf{x}, y)$, we have

$$\gamma(\mathbf{x}, f(\mathbf{x})) - \alpha(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \geq p(\mathbf{x}, \hat{f}(\mathbf{x})) - p(\mathbf{x}, f(\mathbf{x})) + \gamma(\mathbf{x}, f(\mathbf{x})) - \alpha(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \geq 0,$$

where we used the fact that $p(\mathbf{x}, \hat{f}(\mathbf{x})) - p(\mathbf{x}, f(\mathbf{x})) \leq 0$, $\hat{f}(\mathbf{x}) \in [a, b]$ and that $\mathbf{x} \in A$. It follows that

$$(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \leq \frac{\gamma(\mathbf{x}, f(\mathbf{x}))}{\alpha} \leq \frac{\gamma_{\max}}{\alpha}$$

for all $\mathbf{x} \in A$. Taking square roots and recalling that $P(A) > 1 - \epsilon$, we obtain the result. The condition (3.10) is a sufficient condition for (3.9) derived in [1, Corollary 1]. □

Remark 3.6. *When the integral in the objective function 3.3 is replaced by its empirical average* (3.4) *computed from the same data set that is used to enforce the constraints, it is currently unknown whether the generalization bound of Theorem 3.4 remains valid* [6]*. A simple remedy is to use an independent sample for the objective and for the constraints, for example by splitting the data set in two.*

3.5. **Beyond polynomials.** In this section we briefly discuss how the proposed method extends to approximants of the form

$$\hat{f}(\mathbf{x}) = \arg\min_{y \in \mathbf{Y}} p(\mathbf{x}, y),$$

where p is not necessarily a polynomial. The key observation to make is that when parametrizing p as

$$p(\mathbf{x}, y) = \sum_{k=1}^{d_y} h_k(\mathbf{x}) y^k, \tag{3.12}$$

the functions h_k appear in (3.3) and (3.5) only via their evaluations at the data points $\mathbf{x}_i$. Therefore, parametrizing each h_k as $h_k(\mathbf{x}) = \sum_{i=1}^{n_k} c_{k,i} \beta_{k,i}(\mathbf{x})$, where $\beta_{k,i}$ are possibly non-polynomial basis functions, the optimization problem (3.5) remains a semidefinite programming problem with the decision variables $c_{i,k} \in \mathbb{R}$ and the coefficients of γ. The function γ can be parametrized by non-polynomial basis functions in $(\mathbf{x}, y)$ since only evaluations of γ at the samples $(\mathbf{x}_i, y_i)$ appear in (3.5).

If a non-polynomial parametrization of p in y was sought, one would have to resort to certificates of nonnegativity for the given function class akin to Theorem 3.1. Currently, this is well understood for trigonometric polynomials [9] but we envision broader function classes may be considered, given the univariate nature of the nonnegativity certificate required in (3.5) which is significantly less challenging than its multivariate counterpart.

4. Numerical examples

In this section we present several examples demonstrating the effectiveness of the polynomial argmin data structure for regression of functions possessing discontinuities. All examples were solved on a MacBook Air 1.2 GHz Quad-Core Intel Core i7 with 16GB RAM, MOSEK SDP solver. The problems were modeled using Yalmip [18]. The range of all functions was normalized to $\mathbf{Y} = [-1, 1]$ and the parameter α was taken to be 0.01 in all examples. The parameters that vary in the examples

are d_x and d_y in the parametrization of p in (3.2) (i.e., d_x and d_y are the degrees of p in $\mathbf{x}$ and y respectively).

4.1. **Univariate: Approximation of discontinuous functions.** We start by showing the effectiveness of the method on four discontinuous functions depicted in Figure 2 alongside their polynomial argmin approximations. The functions are

$$f_1 = \begin{cases} -1, & x \in [-0.75, 0.75] \\ 1, & x \in [-1, -0.75) \cup (0.75, 1], \end{cases} \quad f_2 = \begin{cases} -1, & x \in [-0.75, -0.25] \cup [0.25, 0.75] \\ 1, & x \in [-1, -0.75) \cup (-0.25, 0.25) \cup (0.75, 1]. \end{cases}$$

$$f_3 = x^2 f_1, \quad f_4 = \sin(2x) f_1.$$

In each case we used 200 data points sampled uniformly at random in $[-1, 1]$ and solved (3.5). We observe a remarkably precise recovery of the discontinuous functions. In (3.2), the minimal degrees d_x and d_y required to obtain an approximation of this accuracy are reported in the figure. For comparison, in Figure 3 we depict the performance on the same task with a neural network with five hidden layers with 20 neurons per layer with the hyperbolic tangent activation functions, trained using Matlab's neural network toolbox with gradient descent; these parameters were selected by manual hyperparameter tuning.

4.2. **Univariate: Parameter dependence.** Here we investigate the dependence of the approximation quality on d_x which is the degree of the polynomials h_k parametrizing the polynomial p in (3.2). We do so on the function from Eq. (66) in [16] that possesses 7 discontinuities. For data, we use two hundred equidistantly spaced samples in the interval $[-1, 1]$. The results are depicted in Figure 4. As expected the approximation quality improves as the degree of h_k increases, obtaining a very precise accuracy for $\deg h_k = 7$. For comparison, Figure 5 depicts also the results with a neural network approximation.

4.3. **Univariate: Challenging continuous functions.** For completeness we briefly report results for approximation of continuous functions. We do so on two functions. The first one is $f(x) = \sqrt{|\sin x|}$ which is a transcendental function with Hölder exponent $1/2$ whose derivative grows unbounded near the origin. The second one is the Runge function $f(x) = (1+25x^2)^{-1}$ which is a smooth function that exhibits the Runge phenomenon (oscillations near the boundary) when approximated by polynomials through interpolation. The results are depicted in Figure 6. As in the previous examples, we observe an accurate fit and no oscillations with low degrees of p in x and y.

4.4. **Bivariate: Approximation of discontinuous functions.** On Figure 7 we represent the `chebfun2` approximation to the indicator function[2] of a bivariate disk

$$\mathbb{I}_{\{x \in \mathbb{R}^2 : \|x\| \le 1/2\}}$$

obtained with the following `chebfun` [8] commands:

```
[x1,x2]=meshgrid(linspace(-1,1,100));
plot(chebfun2(double(x1.^2+x2.^2<=1/4)));
```

We observe that the approximation is corrupted by the typical Gibbs phenomenon encountered when approximating a discontinuous function with polynomials [21, Chapter 9], namely large oscillations near the discontinuity set.

On Figure 8 we demonstrate the performance of our argmin approximation on this indicator function. For data, we chose one thousand randomly sampled points in $[-1, 1]^2$. Contrary to the Chebyshev polynomial approximation in Figure 7 we observe no Gibbs phenomenon and far better accuracy of the approximation despite a more parsimonious parametrization. Indeed, we used

[2]The indicator function of a set is equal to one on the set and zero outside.

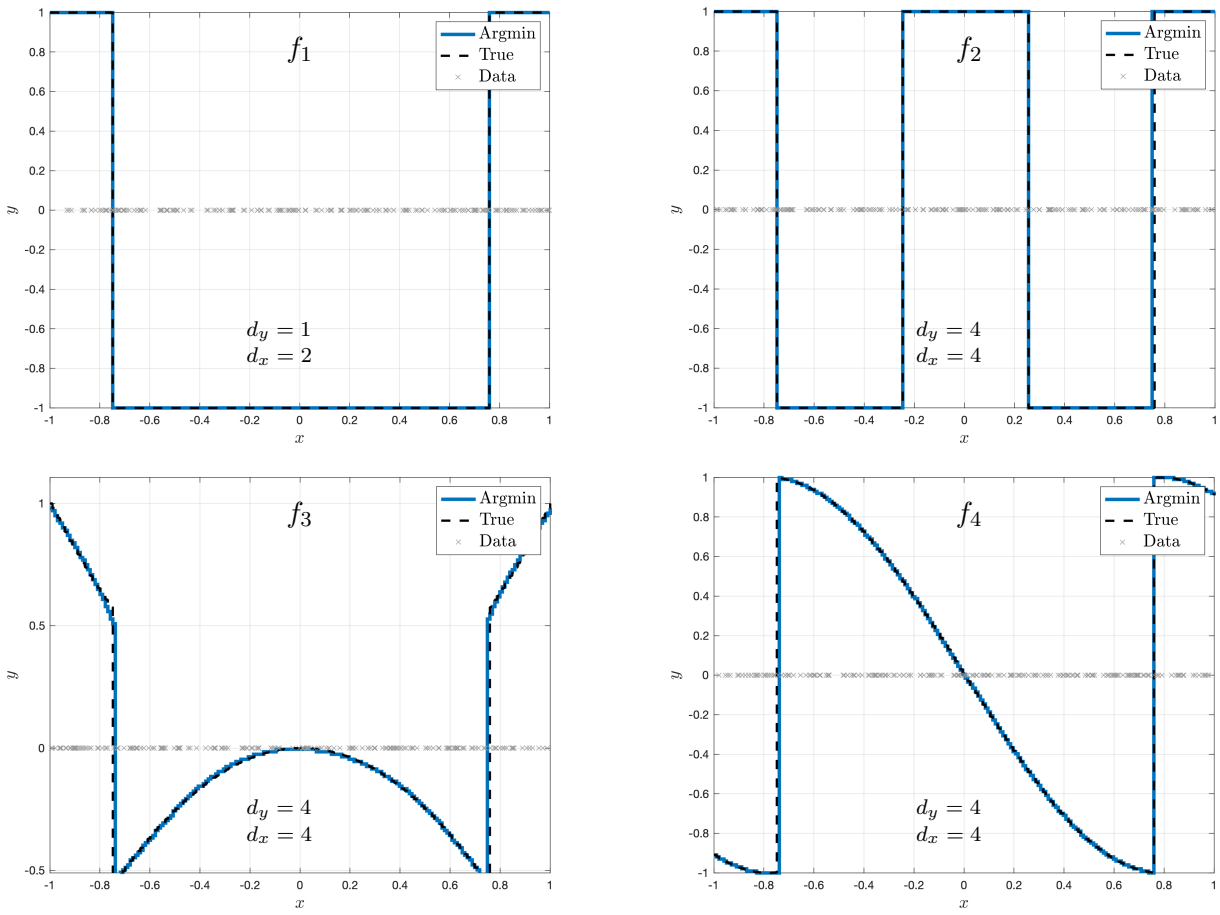

FIGURE 2. Polynomial argmin approximations of different functions with discontinuities.

degrees $d_y = 6$ and $\deg h_k = 6$ whereas the Chebyshev polynomial approximation in Figure 7 is of degree 100.

On Figure 9, we approximate the more challenging weighted sum of indicator functions of three disks

$$\mathbb{I}_{\{\|x\| \leq 1/2\}} + \frac{1}{2}\mathbb{I}_{\{\|x-[3/4,3/4]\| \leq 1/4\}} + \frac{1}{2}\mathbb{I}_{\{\|x+[3/4,3/4]\| \leq 1/4\}}.$$

With the same parameters as in the single disk example we observe a perfect match of our polynomial argmin approximation.

5. Discussion and conclusion

We have presented a simple method based on the argmin of a polynomial for approximation of discontinuous functions. The approach is model-free and mesh-free in the sense of requiring no prior knowledge about the function being approximated, working only with its samples. The approach is grounded in powerful tools from sum of squares optimization, but very simple to use, based only on convex semidefinite programming. The approach shows a great promise in numerical examples and we believe that it may become an invaluable tool in data analysis. We proved the possibility of exact recovery on certain examples of discontinuous functions and provided theoretical analysis of in-sample and out-of-sample error in a probabilistic setting.

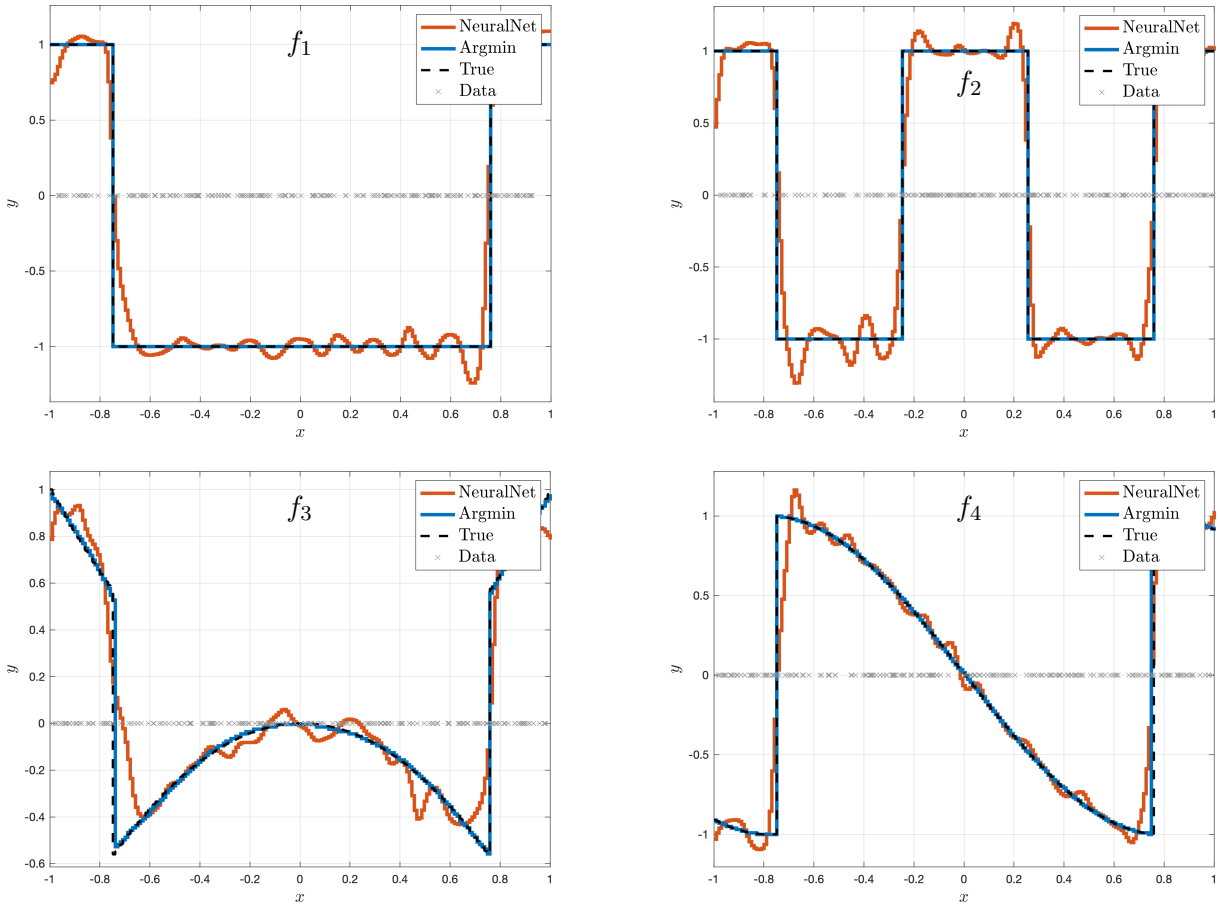

FIGURE 3. Polynomial argmin approximation versus neural network.

While we used general purpose semidefinite solvers to construct our argmin approximants, more efficient approaches can be envisioned. Indeed our formulation boils down to optimization over the cone of univariate non-negative polynomials, a very specific class of semidefinite optimization problems. For example, non-symmetric solvers may perform faster on these problems [19]. Another option could be to bypass numerical optimization and use tailored numerical linear algebra as in [17].

An additional interesting feature of the approximant is that its evaluation at a given point $\mathbf{x} \in \mathbf{X}$ reduces to finding the global minimum of a univariate polynomial on an interval, which can be done efficiently e.g. by matrix eigenvalue computation. A numerically stable algorithm is described in [4, Section 7] and implemented in the `roots` function of the `chebfun` package [8]. It is based on the application of the QR algorithm for finding the eigenvalues of a balanced companion matrix constructed by evaluating the polynomial at Chebyshev points.

This paper is a first step that introduces the argmin approximant and illustrates its promising potential on non trivial numerical examples. We hope that it could inspire some further developments. In particular, we have left open the question of optimal rates of convergence of the argmin approximant or more generally its worst-case performance when considering pre-defined classes of functions to approximate, e.g. in terms of the *manifold width* discussed in [7], which is a generalization of the classical Kolmogorov width. Based on Section 2.1, it is clear that the rates are at least as good as those of polynomial approximation whenever the degree of p in y is at least two. However, we conjecture that the rates are better for discontinuous functions.

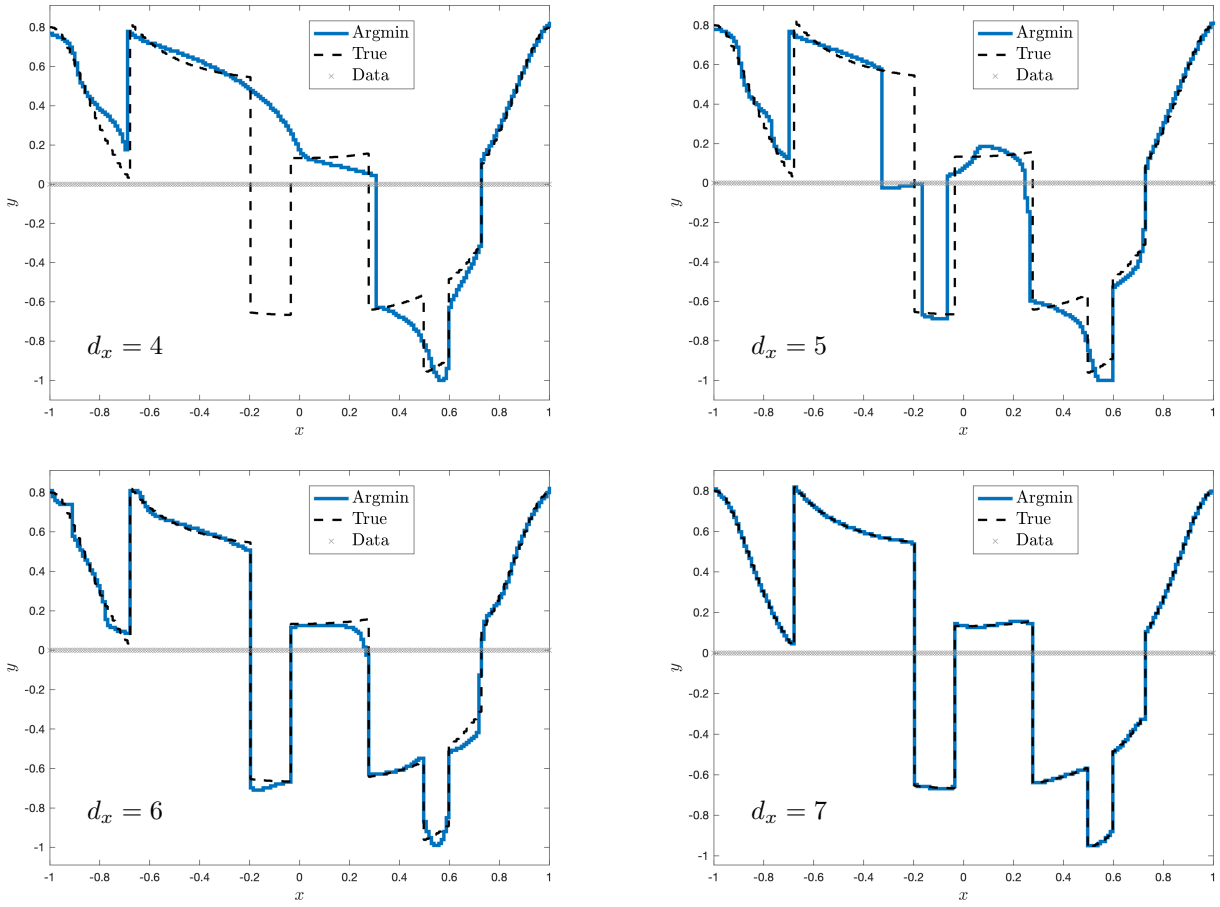

FIGURE 4. Polynomial argmin approximation on a function with 7 discontinuities with $d_y = 6$ for different values of d_x.

REFERENCES

[1] T. Alamo, R. Tempo, A. Luque, D. R. Ramirez. Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. Automatica 52:160-172, 2015.

[2] V. Antun, N. M. Gottschling, A. C. Hansen, B. Adcock. Deep learning in scientific computing: understanding the instability mystery. SIAM News 54:2, 2021.

[3] V. Antun, M. J. Colbrook, A. C. Hansen. Proving existence is not enough: mathematical paradoxes unravel the limits of neural networks in artificial intelligence. SIAM News 55:4, 2022.

[4] Z. Battles, L. N. Trefethen. An extension of Matlab to continuous functions and operators. SIAM J. Sci. Comp. 25(5):1743–1770, 2004.

[5] M. C. Campi, S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. SIAM Journal on Optimization 19(3):1211-1230, 2008.

[6] M. Campi. Personal communication.

[7] A. Cohen, R. DeVore, G. Petrova, P. Wojtaszczyk. Optimal Stable Nonlinear Approximation. Foundations of Computational Mathematics, 22:607–648, 2022.

[8] T. A. Driscoll, N. Hale, L. N. Trefethen (editors). Chebfun Guide. Pafnuty Publications, 2014.

[9] B. Dumitrescu. Positive trigonometric polynomials and signal processing applications. Springer, 2007.

[10] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, J. Tompson. Implicit behavioral cloning. `arXiv:2109.00137`, 2021.

[11] D. Henrion, J. B. Lasserre. Graph recovery from incomplete moment information. Constructive Approximation 56:165-187, 2022.

[12] B. Llanas, S. Lantarón, F. J. Sáinz. Constructive approximation of discontinuous functions by neural networks. Neural Processing Letters 27:209-226, 2008.

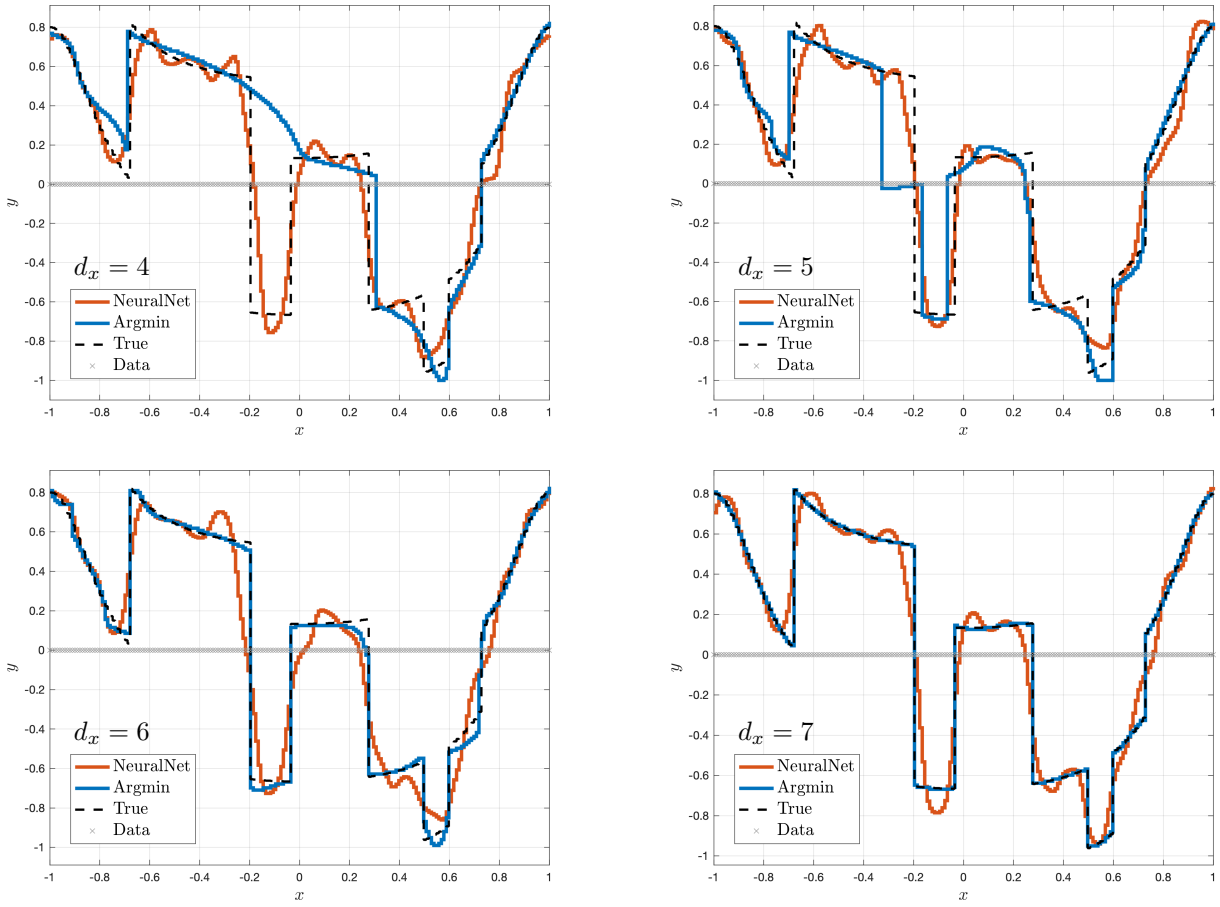

FIGURE 5. Polynomial argmin approximation of a function with 7 discontinuities for different values of the degree of h_k, $k = 1, \ldots, 6$ in comparison with a neural network.

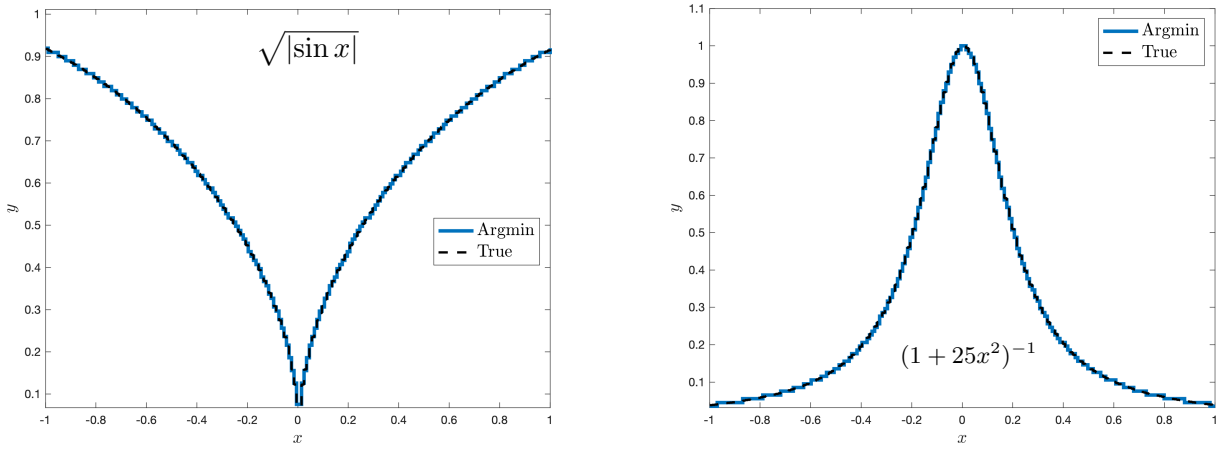

FIGURE 6. Polynomial argmin approximations of two continuous functions with $d_x = 4$, $d_y = 4$. Left: transcendental function with unbounded derivatives near the origin. Right: Runge function.

[13] M. Bozzini, M. Rossini. The detection and recovery of discontinuity curves from scattered data. J. Computational and Applied Mathematics 240:148-162, 2013.

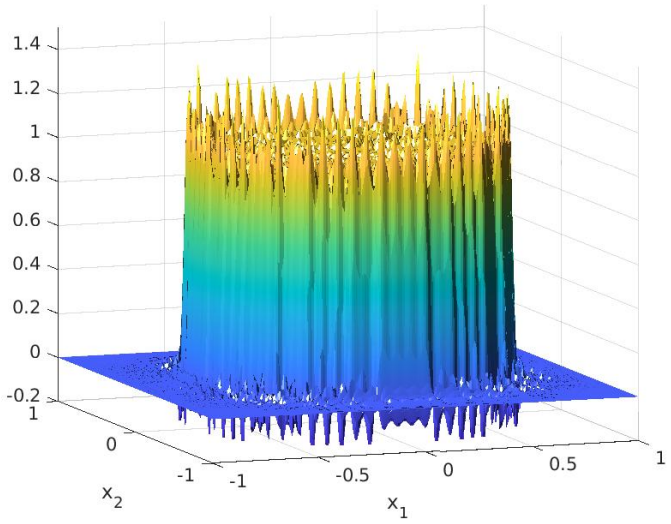

FIGURE 7. Chebyshev polynomial approximation of the indicator function of a bivariate disk obtained by `chebfun2`.

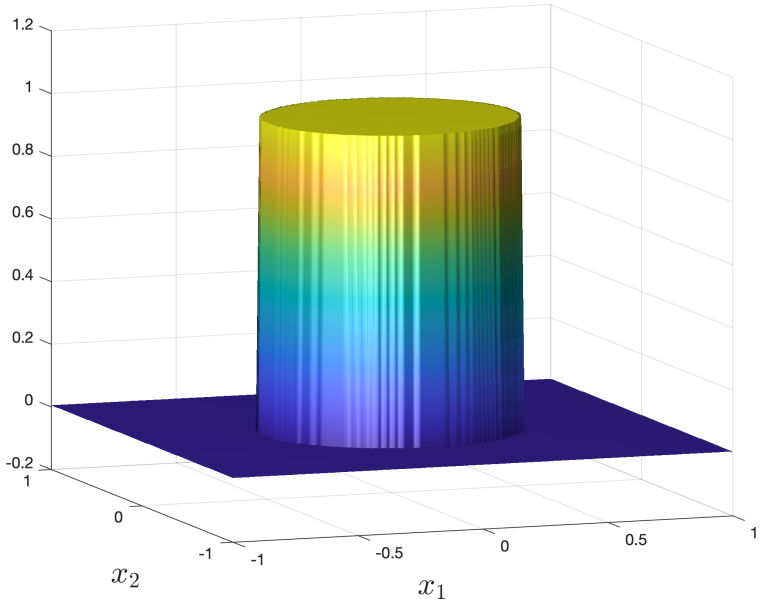

FIGURE 8. Polynomial argmin approximation of the indicator function of a bivariate disk.

[14] S. Marx, T. Weisser, D. Henrion, J. B. Lasserre. A moment approach for entropy solutions to nonlinear hyperbolic PDEs. Mathematical Control and Related Fields, 10(1):13-140, 2020.
[15] S. Marx, E. Pauwels, T. Weisser, D. Henrion, J. B. Lasserre. Semi-algebraic approximation using Christoffel-Darboux kernel. Constructive Approximation 54:391–429, 2021.
[16] K. S. Eckhoff. Accurate and efficient reconstruction of discontinuous functions from truncated series expansions. Math. Comput. 61(204):745-763, 1993.
[17] S.-I. Filip. A robust and scalable implementation of the Parks-McClellan algorithm for designing FIR filters. ACM Trans. Math. Software 43(1), 7:1-24, 2016.
[18] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. IEEE Symp. Computer-Aided Control Design (CACSD), Taiwan, 2004.
[19] D. Papp, S. Yildiz. Sum-of-squares optimization without semidefinite programming. SIAM J. Optim. 29(1):822-851, 2019.

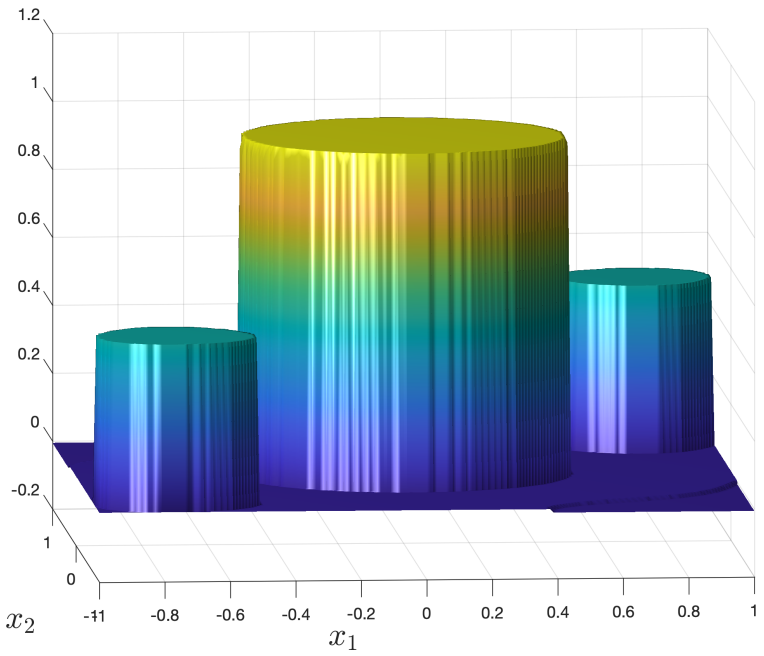

FIGURE 9. Polynomial argmin approximation of the weighted sum of three indicator functions of disks.

[20] V. Powers, B. Reznick. Polynomials that are positive on an interval. Trans. Amer. Math. Soc. 352(10):4677-4692, 2000.
[21] L. N. Trefethen. Approximation Theory and Aproximation Practice. SIAM, 2013.
[22] A. Weisse, G. Wellein, A. Alvermann, H. Fehske. The kernel polynomial method. Reviews of Modern Physics 78(1), 2006.
[23] E. Tadmor. Filters, mollifiers and the computation of the Gibbs phenomenon. Acta Numerica 16:305-378, 2007.