Computing an Efficient Exploration Basis for Learning with Univariate Polynomial Features

Abstract

Barycentric spanners have been used as an efficient exploration basis in online linear optimisation problems in a bandit framework. We characterise the barycentric spanner for decision problems in which the cost (or reward) is a polynomial in a single decision variable. Our characterisation of the barycentric spanner is two-fold: we show that the barycentric spanner under a polynomial cost function is the unique solution to a set of nonlinear algebraic equations, as well as the solution to a convex optimisation problem. We provide numerical results to show that the computation of barycentric spanners in the polynomial case by using our method is significantly faster than the only other known algorithm for the purpose. As an application, we consider a dynamic pricing problem in which the revenue is an unknown polynomial function of the price. We then empirically show that the use of a barycentric spanner to initialise the covariance matrix updates in a Thompson sampling setting leads to lower cumulative regret as compared to standard initialisations. We also illustrate the importance of barycentric spanners in adversarial settings by showing, both theoretically and empirically, that a barycentric spanner achieves the minimax value in a static adversarial linear regression problem where the learner selects the training points while an adversary selects the testing points and influences the noise during training.

Introduction

Background

Many sequential decision-making problems can be cast as an online optimization problem, where a decision-maker, or learner, chooses an action from a decision space D at each round, and receives feedback in the form of a cost from the environment. Well known examples include online routing (Awerbuch and Kleinberg 2008) and dynamic pricing (Keskin and Zeevi 2014). The goal of the decision-maker, or learner, is to learn the best decision over multiple rounds, where "best" is defined in terms of a suitable notion of regret. In the stochastic version of such a problem, the costs are assumed to be generated by a stochastic model, while in the adversarial version, one allows for the possibility that the cost functions may be chosen by an adversary.

Online linear optimization problems form a special class of online optimization problems where the decision set D is a subset (usually compact and convex) of a d-dimensional real vector space, and the costs are linear functions on $\mathbb{R}^d$. In the case of full-information or transparent feedback, the entire cost function is revealed to the learner after each round. On the other hand, in the bandit version of the problem, only the cost of the last decision made by the learner is revealed after each round.

While the well known strategy Follow-the-Perturbed-Leader (FPL) yields a simple and efficient low-regret algorithm for adversarial online linear optimization under full information (Hannan 1957; Kalai and Vempala 2005), the harder bandit version requires more elaborate algorithms that strike a delicate balance between 1) exploration aimed at learning the unknown cost functions, and 2) exploitation that uses a full-information algorithm like FPL on the cost function estimates obtained during exploration (Awerbuch and Kleinberg 2008; McMahan and Blum 2004; Dani and Hayes 2006; Bubeck, Cesa-Bianchi, and Kakade 2012; Abernethy, Hazan, and Rakhlin 2012; Hazan and Karnin 2016).

Exploration Basis

The exploration in many of the algorithms cited above is based on the intuitive idea that the value of a *linear* function at any point can be predicted if the values of the function are known at a set of basis elements. The exploration steps in all these algorithms therefore involve sampling decisions from a carefully chosen subset of the decision set called an *exploration basis*. The choice of the exploration basis is crucial, as a wrong choice can "amplify" the effect of errors or noise that might be present in the function values observed at the basis elements.

To understand this, suppose we wish to estimate a linear function $x \mapsto \mu^{\mathrm{T}} x$ based on noisy measurements $y_i = \mu^{\mathrm{T}} x_i + \epsilon_i$, $i = 1, \ldots, d$, of the linear function on elements of an exploration basis $\{x_1, \ldots, x_d\} \subset \mathbb{R}^d$, with ϵ_i being the noise sample at the ith measurement. Assuming the basis elements to be linearly independent, a simple estimate of μ is given by $\hat{\mu} = (X^{-1})^{\mathrm{T}} y$, where X is the matrix having $x_1, \ldots, x_d$ as its columns. The error that results if we use our estimate $\hat{\mu}$ to predict the value of the function at a new point $x \in \mathbb{R}^d$ is easily seen to be $\hat{\mu}^{\mathrm{T}} x - \mu^{\mathrm{T}} x = \epsilon^{\mathrm{T}} c(x)$, where $c(x) = X^{-1} x$ is the vector of coefficients required to write x as a linear combination of the basis elements. It is evident that the error in predicting the function at a general point x

depends on the "size" of the coefficients needed to express x in terms of the basis elements. For a geometric explanation of the same point, see Awerbuch and Kleinberg (2008). The preceding discussion suggests that the exploration basis must be chosen such that all elements in the decision space can be written as a linear combination of the basis elements using coefficients that are, in some suitable sense, small.

Hazan and Karnin (2016) use the L_2 norm of the coefficient vector as a measure of smallness for defining an efficient, low variance exploration basis. They define a volumetric spanner as a set of elements of the decision set such that every decision vector can be written as a linear combination of the basis elements with coefficients whose Euclidean norm does not exceed 1. The algorithm for the adversarial setting given by Hazan and Karnin (2016) uses a volumetric spanner for a low variance exploration basis. Alternative mechanisms for exploration based on convex analysis were used by Abernethy, Hazan, and Rakhlin (2012) and Bubeck, Cesa-Bianchi, and Kakade (2012). However, the first notion of an exploration basis in the context of online bandit linear optimization was that of a *barycentric spanner*, and appeared in the seminal work of Awerbuch and Kleinberg (2008).

Barycentric Spanner

A *barycentric spanner* for a given $D \subset \mathbb{R}^d$ is a finite subset of D such that every element in D can be expressed as a linear combination of elements of the subset using coefficients in $[-1, 1]$. If the coefficients are allowed to lie in $[-C, C]$ for some $C > 1$, the corresponding set of elements is called a C-approximate barycentric spanner. Barycentric spanners or C-approximate barycentric spanners have been used as part of bandit linear optimization algorithms for the adversarial setting in Awerbuch and Kleinberg (2008); Bartlett et al. (2008); Dani and Hayes (2006); McMahan and Blum (2004); Dani, Kakade, and Hayes (2008), and for the stochastic setting in Dani, Hayes, and Kakade (2008). In a different application, Chen and Moitra (2019) used barycentric spanners to estimate a mixture of binary product distributions from a sample drawn from the mixture.

Awerbuch and Kleinberg (2008) show that a compact decision set $D \subset \mathbb{R}^d$ always has a barycentric spanner with at most d elements. Furthermore, given $C > 1$, Awerbuch and Kleinberg (2008) give an algorithm that computes a C-approximate barycentric spanner for a general compact set $D \subset \mathbb{R}^d$ with $O(d^2 \log_C d)$ calls to an optimization oracle for performing linear optimization over D. While it is preferable for C to be closer to 1, the complexity bound for the algorithm given by Awerbuch and Kleinberg (2008) diverges as C approaches 1. Moreover, the optimization step in the algorithm has to be implemented afresh for different instances of the decision set D.

Present Work

In this paper, we consider the problem of computing a barycentric spanner for the special case where the decision set D is the set D_n defined by $D_n = \{[1, p, p^2, \ldots, p^n]^\mathrm{T} \in \mathbb{R}^{n+1} : p \in [p_{\min}, p_{\max}]\}$ for some integer n. In the context of online optimization problems, it is natural to consider the decision set D_n in the case where the cost functions are polynomials of degree n in a single decision variable p. Formulating the decision set in this manner permits one to cast an online optimization problem with polynomial costs as an online linear optimization problem. Furthermore, having a barycentric spanner for the set D_n enables the application of adversarial bandit linear optimization algorithms to the case of polynomial cost functions.

The case of a polynomial objective function is of interest in applications such as dynamic pricing of retail products, where the seller of a product would like to sequentially learn the price that elicits the maximum revenue for that product in the case where the market demand curve for the product is unknown, but modeled as a polynomial in the price. We illustrate the role of barycentric spanners in an online setting with the help of a dynamic pricing problem. We cast the problem as a stochastic bandit linear optimization problem, and apply the Thompson sampling algorithm (den Boer 2015).

To clarify the role of barycentric spanners in adversarial settings, we consider a static adversarial linear regression problem in which a learner first selects training points for fitting a linear function from noisy measurements. An adversary observes the learner's choice, selects points for testing the learner's fit, and distributes a given total noise variance across the training points chosen by the learner.

The main contributions of the paper are as follows.

1. We show that the barycentric spanner of the decision set D_n introduced above can be characterized through the unique optimizer of a convex optimization problem or, equivalently, the unique solution of a set of nonlinear equations. Our characterization makes it possible to compute the barycentric spanner of the set D_n efficiently in polynomial time, using either interior point methods for convex optimization (Nesterov and Nemirovskii 1994), or trust region methods for solving nonlinear algebraic equations. We provide empirical run-times of the resulting algorithms which turn out to be significantly faster than the algorithm of Awerbuch and Kleinberg (2008).

2. We show that the barycentric spanner of D_n can be easily constructed from the barycentric spanner for the standard case where the domain of the polynomials is the unit interval. Effectively, this means that the computation of the barycentric spanner is required only once for a given polynomial degree. We also show how symmetry properties can be exploited to further reduce the computations.

3. We empirically show that choosing the covariance of the prior distribution based on a barycentric spanner leads to improved regret performance when compared with standard choices in Thompson sampling applied to an online linear bandit formulation of dynamic pricing. We also present empirical evidence to show that the performance improvement is robust with respect to some features of the unknown demand curve.

4. We show theoretically and empirically, that the learner in the adversarial linear regression setting described above can achieve the lowest worst case expected mean square

error by choosing the training points from the barycentric spanner, where the worst case is over the adversary's choices.

We start by introducing the required definitions and notation in the next section.

Barycentric Spanners

Notations and Definitions

Let $D \in \mathbb{R}^d$, and $C > 0$. A finite-subset $\{x_1, \ldots, x_k\} \subseteq D$ is a C-approximate barycentric spanner for D if, for every $z \in D$, there exist $c_1, \ldots, c_k \in [-C, C]$ such that $z = c_1 x_1 + \cdots + c_k x_k$. A *barycentric spanner* for D is a 1-approximate barycentric spanner for D. Thus, every element of D may be written as a linear combination of elements of a barycentric spanner using coefficients in $[-1, 1]$. If D is compact, then D has a barycentric spanner with at most d elements (Awerbuch and Kleinberg 2008).

For each positive integer n, define $f_n : \mathbb{R} \to \mathbb{R}^{n+1}$ by $f_n(p) = [1, p, p^2, \ldots, p^n]^{\mathrm{T}}$. Given $w = [w_1, \ldots, w_{n+1}]^{\mathrm{T}} \in \mathbb{R}^{n+1}$, $V(w) \overset{\text{def}}{=} [f_n(w_1), \ldots, f_n(w_{n+1})]$ is the $(n+1) \times (n+1)$ Vandermonde matrix formed from the elements of w.

Let $[p_{\min}, p_{\max}]$ be a closed interval of $\mathbb{R}$. In the sequel, we will be concerned with the set $D_n \overset{\text{def}}{=} \{f_n(p) : p \in [p_{\min}, p_{\max}]\} \subset \mathbb{R}^{n+1}$ for some $n \geq 1$. The motivation for considering this particular set follows from the discussion given in the introduction.

To begin, we claim that the set D_n is not contained in a d-dimensional subspace of $\mathbb{R}^{n+1}$ for any $d < n+1$. Indeed, if $z \in \mathbb{R}^{n+1}$ is orthogonal to the subspace spanned by D_n, then $z^{\mathrm{T}} f_n(\cdot)$ is identically zero on $[p_{\min}, p_{\max}]$. Equating the first $n+1$ derivatives of the constant function $z^{\mathrm{T}} f_n(\cdot)$ to zero results in a triangular system of equations which is easily solved to obtain $z = 0$. Thus the orthogonal complement of the span of D_n is the trivial subspace $\{0\}$, and our claim follows.

Main Results

Our main result below gives two characterizations for the barycentric spanner of the set D_n. All the proofs related to this section are given in the supplementary material.

Theorem 1. *Suppose $\mathbf{p} \in \mathbb{R}^{n+1}$ is such that $p_{\min} \leq p_1 \leq \cdots \leq p_{n+1} \leq p_{\max}$. Then the following three statements are equivalent.*

1. *The set $\{f_n(p_1), \ldots, f_n(p_{n+1})\} \subset D_n$ is a barycentric spanner for D_n.*
2. *The vector $\mathbf{p}$ satisfies $p_{\min} = p_1 < p_2 \cdots < p_{n+1} = p_{\max}$ and*
$$\sum_{1 \leq j \leq n+1, j \neq i} \frac{1}{p_i - p_j} = 0, \ i = 2, \ldots, n. \tag{1}$$
3. *The vector $\mathbf{p}$ is the unique global solution of the optimization problem*
$$\max_{\substack{w \in \mathbb{R}^{n+1} \\ p_{\min} = w_1 < \cdots < w_{n+1} = p_{\max}}} \ln|\det V(w)|. \tag{2}$$

The proof of Theorem 1 depends on the following proposition. The proposition states that the optimization problem appearing in 3) of Theorem 1 is a convex optimization problem with a unique global maximizer which is also the unique solution of the set of nonlinear equations (1).

Proposition 1. *Let $a < b$, and define the set $C \overset{\text{def}}{=} \{z \in \mathbb{R}^k : a < z_1 < z_2 < \cdots < z_k < b\}$. Then the set of equations*
$$\frac{1}{z_i - a} + \sum_{j \neq i} \frac{1}{z_i - z_j} + \frac{1}{z_i - b} = 0, \ i = 1, \cdots, k, \tag{3}$$
has a unique solution z^* in the convex set C. Moreover, z^* is the unique global maximizer in C of the strongly concave function $U : C \to \mathbb{R}$ defined by
$$U(z) \overset{\text{def}}{=} \ln \left| \left(\prod_{i=1}^{k} (a - z_i)(b - z_i) \right) \left(\prod_{\substack{i=1,\cdots,k; \\ j > i,}} (z_i - z_j) \right) \right|. \tag{4}$$
Finally, z^* satisfies
$$z_i^* + z_{k-i+1}^* = a + b, \ \ i = 1, \ldots, k. \tag{5}$$

Discussion

Proposition 1 along with Theorem 1 implies that the set D_n has a unique barycentric spanner, and this barycentric spanner can be found either by solving the set of nonlinear equations (1), or by solving the convex optimization problem (2), both of which have unique solutions. More importantly, both problems can be solved efficiently using well known algorithms. For example, equation (1) can be solved using Powell's hybrid method (Powell 1970), while the convex optimization problem (2) can be solved using an interior point method (Nesterov and Nemirovskii 1994). Note that the computational run time of both types of algorithms grows polynomially in the number of variables.

Next, observe that if p_i, $i = 1, \ldots, n+1$, satisfy (1), then so do $ap_i + b$ for all $a, b \in \mathbb{R}$. Since the interval $[p_{\min}, p_{\max}]$ is an image of the unit interval under an affine map, it follows that a barycentric spanner for any given values of $p_{\min}$ and $p_{\max}$ can simply be computed from the barycentric spanner for the canonical case $p_{\min} = 0$ and $p_{\max} = 1$. Effectively, the problems (1) or (2) have to be solved for each value of n only once.

The relations (5) imply that the points $p_i, i = 1, \ldots, n+1$, yielding the barycentric spanner are symmetrically placed about the midpoint $\bar{p} \overset{\text{def}}{=} \frac{1}{2}(p_{\min} + p_{\max})$ of the interval $[p_{\min}, p_{\max}]$. Thus, it is sufficient to find points lying only on one side of the midpoint. This can be essentially achieved by using the symmetry relations (5) to eliminate (roughly) half the variables from (1) and (2). The details are given in the supplementary material.

Empirical Comparison of Run Times

Table 1 provides a comparison of the run times (in seconds) to compute a barycentric spanner for the set D_n for various

Polynomial degree n	A-K			Non linear equations		Convex optimization	
	$C=1$	$C=2$	$C=5$	Full	Reduced	Full	Reduced
2	0.097	0.097	0.097	0.0002	0.00003	0.0209	0.0154
5	4.537	0.372	0.372	0.0007	0.0004	0.0713	0.0478
10	35.185	2.891	2.698	0.0081	0.0025	0.2296	0.1517
13	53.752	5.537	5.467	0.0158	0.0038	0.3678	0.2087
15	65.656	8.163	7.937	0.0316	0.0081	0.4853	0.2691
20	115.45	19.13	18.93	0.0793	0.0241	0.9198	0.4967
25	NA	NA	NA	0.206	0.068	1.933	0.804
30	NA	NA	NA	0.415	0.099	2.126	1.278
45	NA	NA	NA	2.305	0.377	4.656	2.527
60	NA	NA	NA	6.985	1.534	9.618	5.975
80	NA	NA	NA	24.676	3.299	15.636	8.196

Table 1: Time in seconds for computing an exact or approximate barycentric spanner from different methods.

values of n in the canonical case $[p_{\min}, p_{\max}] = [0, 1]$ using the full versions (1) and (2) and reduced versions given by (20)-(23) in the supplementary material. Table 1 also gives the execution time of our implementation of the algorithm provided by Awerbuch and Kleinberg (2008) (referred to as A-K) for computing a C-approximate barycentric spanner with $C = 1, 2$ and 5. As expected, Table 1 shows that computations using the reduced versions of either the nonlinear equations or the convex optimization are faster than with the corresponding full versions.

For higher values of n, our implementation of the A-K algorithm does not give the correct spanner due to numerical inadequacy. We emphasize the fact that we have implemented the A-K algorithm by fully exploiting the structure of our decision space to increase efficiency. Specifically, the search for an initial set of linearly independent vectors from the set D as well as repeated optimization of the determinant of $n+1$ vectors chosen from D in the original A-K algorithm are both implemented after specializing to the polynomial setting.

The nonlinear equations (1) were solved using the *fsolve* function available in *SciPy optimize* Python package, which uses a modified version of Powell's hybrid method. The optimization in (2) was achieved using *CVXPY* Python package (Diamond and Boyd 2016). All computations were performed on an Intel® Core™ i5-7200U CPU with 8GB memory and four cores, each running at 2.50GHz.

Dynamic Pricing

In this section, we illustrate the impact of using barycentric spanners in the context of dynamic pricing which has been widely studied as a bandit optimization problem (see den Boer (2015) for references). More formally, suppose a firm sells a product over a time horizon of T periods. In each period, $t = 1, 2, \ldots, T$, the seller must choose a price p_t from a given feasible set $[p_{\min}, p_{\max}] \in \mathbb{R}$, where $0 \leq p_{\min} < p_{\max} < \infty$. The seller observes the demand d_t according to a linear demand model: $d_t = \alpha - \beta p_t + \epsilon_t$ for $t = 1, 2, \ldots, T$, where $\alpha, \beta > 0$ represent the parameters of the demand model which are unknown to the seller, and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ represents unobserved demand perturbations. The linear demand model is often considered in literature (see Kleinberg (2005)). The seller's single period revenue r_t in period t equals $r_t = d_t p_t$. This leads to a quadratic dependence of r_t on p_t.

More generally, one can consider demand models that lead to a higher degree polynomial dependence of revenue on the price. Hence, we consider a general polynomial for the revenue function $r(p_t) = g(p_t) + \epsilon_t$, where $g(p_t) = \tilde{\mu}_0 + \tilde{\mu}_1 p_t + \tilde{\mu}_2 p_t^2 + \cdots + \tilde{\mu}_n p_t^n$ and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. The firm's goal is to learn the unknown parameters $\tilde{\mu}_0, \tilde{\mu}_1, \ldots, \tilde{\mu}_n$ from noisy observations of price and revenue pairs $\{(p_t, r_t)\}_{t=1}^T$ well enough to reduce the T-period expected regret, defined as $R(T) = \sum_{t=1}^T [r^* - \mathbb{E}(r(p_t))]$, where $r^* = \max_{p \in [p_{\min}, p_{\max}]} g(p)$ is the optimal expected single period revenue. The above formulation of the dynamic pricing problem results in a bandit optimization problem, to which the Thompson sampling (TS) algorithm may be efficiently applied (see Ganti et al. (2018)).

TS begins by putting a prior distribution over the unknown parameters. We choose the prior over the parameter vector $\tilde{\mu} = [\tilde{\mu}_0, \tilde{\mu}_1, \tilde{\mu}_2, \cdots, \tilde{\mu}_n]$ to be multi-variate Gaussian with mean vector μ_0 and co-variance matrix A_0. In this case, the posterior distribution over $\tilde{\mu}$ at time step t is also multi-variate Gaussian with the following update equations for mean vector $\mu_t \in \mathbb{R}^{n+1}$ and co-variance matrix $A_t \in \mathbb{R}^{n \times n}$ (see Bagnell (2005)):

$$\left.\begin{array}{ll} A_{t+1}^{-1} & = A_t^{-1} + \sigma^{-2} x_{t+1} x_{t+1}^{\mathrm{T}}, \\ A_{t+1}^{-1} \mu_{t+1} & = A_t^{-1} \mu_t + \sigma^{-2} r_{t+1} x_{t+1} \end{array}\right\} \quad (6)$$

where $x_t = f_n(p_t)$. The convergence of (6) as well as the regret incurred by any algorithm based on these updates is, expectedly, dependent on the initialization A_0 and μ_0.

We claim that there is a natural way of using a barycentric spanner to initialize A_0 and μ_0, and show through numerical experiments that such an initialization leads to lower regret than the baseline method. Let $\{b_1, b_2, \ldots, b_{n+1}\}$ be a barycentric spanner for the set D. We query the revenue curve at each of these barycentric points once, and perform a least squares fit on the resulting data. Denote $B = [b_1, b_2, \ldots, b_{n+1}]$ and $\epsilon = [\epsilon_1, \epsilon_2, \cdots, \epsilon_{n+1}]$, where $\epsilon_1, \epsilon_2, \ldots, \epsilon_{n+1}$ are the noise samples hidden in the data.

It is easy to show that the least squares fit at points of the barycentric spanner gives $\mu = \tilde{\mu} + (BB^{\mathrm{T}})^{-1}B\epsilon$ (see (7) in the next section). Further, $\mathbb{E}(\mu) = \tilde{\mu}$ and hence the least squares estimate is an unbiased estimator for $\tilde{\mu}$. The covariance matrix of μ is $\mathbb{E}[(\mu - \tilde{\mu})(\mu - \tilde{\mu})^{\mathrm{T}}] = (BB^{\mathrm{T}})^{-1}B\mathbb{E}(\epsilon\epsilon^{\mathrm{T}})B^{\mathrm{T}}(BB^{\mathrm{T}})^{-1} = \sigma^2(BB^{\mathrm{T}})^{-1}$ since $\mathbb{E}(\epsilon\epsilon^{\mathrm{T}}) = \sigma^2 I$. Thus, it makes sense to choose our prior with $\mu_0 = \mu$ and $A_0 = \sigma^2(BB^{\mathrm{T}})^{-1} = \sigma^2 \left(\sum_{i=1}^{n+1} b_i b_i^{\mathrm{T}}\right)^{-1}$ or $A_0^{-1} = \sigma^{-2}\left(\sum_{i=1}^{n+1} b_i b_i^{\mathrm{T}}\right)$ as indicated in Algorithm 1. The confidence-ball-based algorithm for stochastic bandit linear optimization given by Dani, Hayes, and Kakade (2008) also uses updates similar to (6) along with the above initialization for A_0^{-1}.

Baseline method: It is common to choose the covariance matrix A_0 to be a multiple of identity matrix (see Agrawal and Goyal (2013) and Ganti et al. (2018)). We use the baseline method as Thompson Sampling with the covariance matrix initialization $A_0^{-1} = I$ in all our experiments, and compare its performance with Algorithm 1 in the next subsection .

Algorithm 1 Thompson sampling for dynamic pricing

Input: Weight vector $\tilde{\boldsymbol{\mu}} = [\tilde{\mu}_0, \tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_n]$, total_iterations, noise σ^2.
Initialization: Set $D = \{\mathbf{p} := (1, p, p^2, \ldots, p^n), p \in [p_{\min}, p_{\max}]\}$.
Step 1. Find barycentric spanner $b_1, b_2, \ldots, b_{n+1}$ for D.
Step 2. Perform a least square fit to obtain μ_0 by suggesting prices at each of the barycentric spanner points.
Step 3. Compute the initial regret R_0 from performing Step 2. Set $A_0^{-1} = \sigma^{-2}\sum_{i=1}^{n+1} b_i b_i^{\mathrm{T}}$.
Step 4. Sample $\mathbf{w}_0 \sim \mathcal{N}(\mu_0, A_0)$ and set $h_0(p) = \mathbf{w}_0^{\mathrm{T}}\mathbf{p}$. Find $p_0^* = \arg\min_{p_{min} \le p \le p_{max}} h_0(p)$.
Step 5. Set $\mathbf{p_0} = [1, p_0^*, (p_0^*)^2, \ldots, (p_0^*)^n]$, $t = 0$ and $C_0 = R_0$.
while t $\le$ total_iterations **do**
  Simulation: $r_t \leftarrow$ environment(p_t^*); environment represents $\tilde{\boldsymbol{\mu}}^{\boldsymbol{T}}.\mathbf{p} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
  Learning: $A_{t+1}^{-1} = A_t^{-1} + \frac{\mathbf{p_t}\mathbf{p_t}^{\mathrm{T}}}{\sigma^2}, A_{t+1}^{-1}\mu_{t+1} = A_t^{-1}\mu_t + \frac{r_t\mathbf{p_t}}{\sigma^2}$.
  Sampling: $\mathbf{w}_{t+1} \sim \mathcal{N}(\mu_{t+1}, A_{t+1})$ and set $h_{t+1}(p) = \mathbf{w}_{t+1}^{\mathrm{T}}\mathbf{p}$.
  Optimization: Find $p_{t+1}^* = \arg\max_{p_{min} \le p \le p_{max}} h_{t+1}(p)$.
  Regret: $R_t = \tilde{\mu}^{\mathrm{T}}p_{best}^* -$ environment(p_t^*), Cumulative regret $C_t = C_t + R_t$.
  Set $t \leftarrow t + 1$.
end while

Simulation Results

The initialization steps 2-3 in Algorithm 1 query the unknown revenue curve at barycentric points and fit a least squares model. The other initialization steps set the ground for learning the unknown parameters. The algorithm relies on (6) for learning, and uses Thompson sampling for suggesting the new price at each iteration. Specifically, at each round t, the algorithm samples a parameter vector from the posterior distribution, provides the optimal price for the sampled parameter vector to the environment and observes the revenue, and updates the posterior distribution using the observation according to (6). For each price value p suggested by the algorithm, the environment returns a noisy value $r = g(p) + \epsilon$ for the revenue.

We compared the performance of Algorithm 1 with the baseline method in several experiments, and uniformly observed that the Algorithm 1 significantly outperforms the baseline method (see figures 1 and 2). In fact, we experimented with $A_0^{-1} = \lambda I$ for various values of λ as well as various polynomial degrees for revenue function, and observed that Algorithm 1 continues to outperform the baseline method. In the first plot in Figure 1 (which is generated by letting $\lambda = 1$), we consider an environment which returns the revenue $r = -p^4 + 22p^3 - 165p^2 + 480p - 150 + \epsilon$, at the price $p \in [1, 10]$, where ϵ is a zero-mean Gaussian noise sample with $\sigma = 10$. The second plot in Figure 1 shows the regret for a second degree polynomial ($r = 1.1p - 0.5p^2 + \epsilon, \epsilon \sim \mathcal{N}(0, 0.01)$, $p \in [0.75, 2]$). In both plots, the expected cumulative regret is estimated by averaging over 10 sample paths. To close the section, we present the results of some robustness checks performed on Algorithm 1.

1. **Different degree for the polynomial vs model:** We also did a wide range of experiments when the degree of the true revenue function is different from the one assumed in the algorithm. The first plot in Figure 2 shows a typical result.

2. **Non polynomial models:** We also ran the algorithm with radial basis functions as the true revenue function. The second plot in Figure 2 shows the regret comparison when the true revenue curve is the radial basis function $100e^{-(p-5)^2/20} + \epsilon, \epsilon \sim \mathcal{N}(0, 9)$, and $p \in [1, 10]$, but the learned model is a 4th degree polynomial.

Linear Regression: Adversarial Setting

To understand how barycentric spanners help in an adversarial setting, consider a simple linear regression problem with an adversarial twist. A learner selects d training points $x_1, \ldots, x_d \in D \subseteq \mathbb{R}^d$, and observes noisy measurements $y_i = g(x_i) + \epsilon_i, i = 1, \ldots, d$, of an unknown linear function $g(x) = \mu^{\mathrm{T}}x$, with ϵ_i being independent random variables with zero mean and variance σ_i^2. The noise variances are chosen by an adversary with the knowledge of the learner's choice of training points. The adversary's choice of noise variances is subject to the constraint $\sigma_1^2 + \cdots + \sigma_d^2 \le \sigma^2$. The adversary also chooses k test points $z_1, \ldots, z_k \in \mathcal{D}$ at which the linear fit obtained by the learner is tested. The adversary's goal is to force as large a expected mean square testing error as possible by choosing the noise variances, the number of test points k, and the test points themselves.

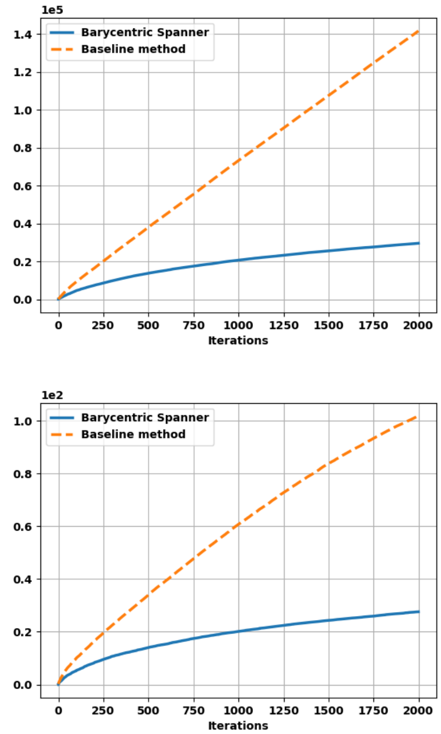

Figure 1: Expected cumulative regret comparison for a fourth (top) and a second (bottom) degree revenue function for the linear demand model considered by Keskin and Zeevi (2014).

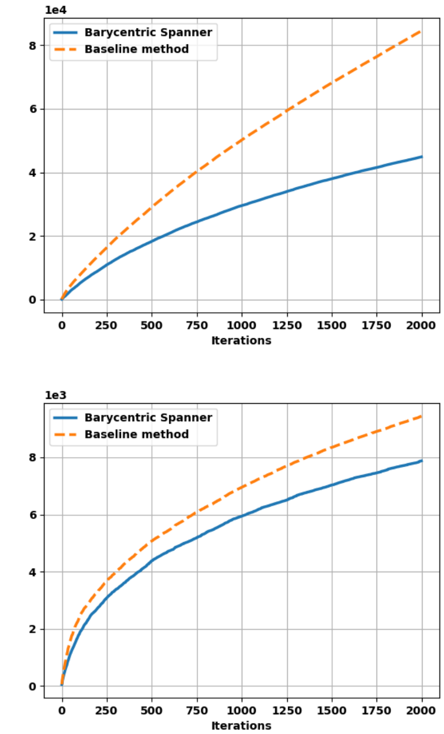

Figure 2: Expected cumulative regret comparison for 4th degree polynomial learnt using a 7th degree model (top) and a radial basis function when learnt by assuming a 4th degree model (bottom).

Let $X \stackrel{\text{def}}{=} [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}$, $y = [y_1, \ldots, y_n]^{\mathrm{T}} \in \mathbb{R}^n$, and $\epsilon = [\epsilon_1, \ldots, \epsilon_n]^{\mathrm{T}}$. Then, $y = X^{\mathrm{T}}\mu + \epsilon$. We assume that $\mathrm{rank}(X) = d$. The least-squares estimate $\hat{\mu}$ of μ is obtained by minimizing the sum of squares of the training errors, that is, $\|y - X^{\mathrm{T}}\hat{\mu}\|_2^2$, and is given by

$$\hat{\mu} = (XX^{\mathrm{T}})^{-1}Xy = \mu + (XX^{\mathrm{T}})^{-1}X\epsilon. \tag{7}$$

The learner's estimate $\hat{g}$ of the function g is then given by $\hat{g}(x) = \hat{\mu}^{\mathrm{T}}x$ for $x \in \mathbb{R}^d$. The learner's goal then is to choose X such that the worst case expected value of the mean square error (MSE), $\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}[\hat{g}(z_i) - g(z_i)]^2$, over the adversary's choice of testing points $z_1, z_2, \ldots, z_k$ and variances σ_i^2, $i = 1, \ldots, d$, is minimized.

Proposition 2. *The expected mean-square testing error is given by*

$$\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}[\hat{g}(z_i) - g(z_i)]^2 = \frac{1}{k}\sum_{j=1}^{k}[\sigma_1^2(e_1^{\mathrm{T}}X^{-1}z_j)^2 + \cdots + \sigma_d^2(e_d^{\mathrm{T}}X^{-1}z_j)^2]. \tag{8}$$

Moreover, the learner can minimize the worst-case (over the adversary's choices) expected MSE by choosing the training points to form a barycentric spanner for the set D.

The proposition, which is proved in the supplementary material, states that using a barycentric spanner for training leads to the best worst case expected MSE. We performed several numerical experiments around this idea for the polynomial case (where X in (8) is the Vandermonde matrix $V(\mathbf{p})$). The observations are reported in the subsection below.

Numerical Illustration

We numerically compare the difference between the worst case expected MSE in (8) when a polynomial regression model is trained using a barycentric spanner (BS) versus a 2-approximate barycentric spanner (2-BS) for various polynomial degrees, and observed that the worst case expected MSE is equal to σ^2 when trained with a BS, and can be significantly higher than σ^2 when trained with a 2-BS.

While we performed several experiments to test the above observation, we restrict ourselves to describing only one of them here. We assume that the learner has access to noisy observations of the following 9th degree polynomial,

$$g(x) = x^9 - 27x^8 + 323x^7 - 2247x^6 + 10017x^5 - 29673x^4 + 58401x^3 - 73629x^2 + 53946x - 17494, \tag{9}$$

for $x \in [2, 4]$, where the noise at each observation is zero-mean Gaussian with $\sigma = 0.1$. We test the correctness of our fit at only one test point, since an adversary can always choose the worst case test point every time in the case of multiple testing opportunities. Letting $k = 1$ in (8) yields the expected MSE at a test point z to be $\sigma^2 \|X^{-1}z\|_\infty^2$.

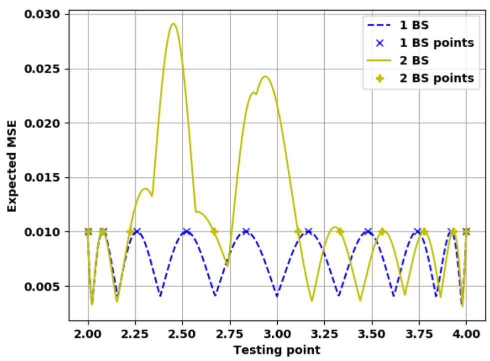

Figure 3: Expected MSE as function of the test point within the feasible range [2,4] that results from training at barycentric spanner (BS) points and 2-approximate barycentric spanner (2 BS) points.

Figure 3 shows a comparison of the expected MSE as a function of the test point z over the entire domain when the training is performed at a barycentric spanner and a 2-approximate barycentric spanner. We observe that the worst case expected MSE (represented by the maximum value of the plot) is 0.01 when training is performed at a barycentric spanner, and 0.0291 when training is performed at a 2-approximate barycentric spanner. The worst case MSE when training at the barycentric spanner is always $\sigma^2 = 0.01$, while the test points that yield the worst case MSE are elements of the barycentric spanner themselves (marked with blue crosses in Figure 3). It is clear from the figure that the worst case expected MSE resulting from training at a 2-approximate barycentric spanner can be nearly 3 times the expected MSE resulting from training at the barycentric spanner.

We also computed the expected MSE in (8) for a fixed set of equidistant testing points for three sets of training points (all of the same cardinality): (1) barycentric spanner (2) a set of equidistant points (3) a fixed set of randomly-chosen points. Across several numerical experiments, it was uniformly observed that *the expected MSE is minimal when the polynomial model is trained at barycentric points.*

Testing → / Training ↓	1000 equidistant points
Barycentric spanner	**0.0090**
10 Equidistant points	0.036
10 Random points	0.3169

Table 2: Expected MSE at 1000 equidistant testing points averaged over 500 trials for the 9th degree polynomial (9)

Table 2 shows a comparison of the MSE averaged over 500 trials when the MSE is computed over 1000 equidistant testing points in the domain of interest, that is, the closed interval $[2, 4]$ for the polynomial (9). The sets of training points chosen for comparison are a barycentric spanner, a set of 10 uniformly spaced points including 2 and 4, and the fixed set of 10 randomly selected training points $\{2.72, 2.64, 2.12, 2.04, 3.44, 2.96, 2.99, 3.96, 2.24, 3.76\}$. The results given in Table 2 show that choosing a barycentric spanner as the set of training points leads to the lowest expected MSE. The same behavior can also be visually noticed in Figure 4, which depicts the polynomials learned in one of the 500 trials summarized in Table 2.

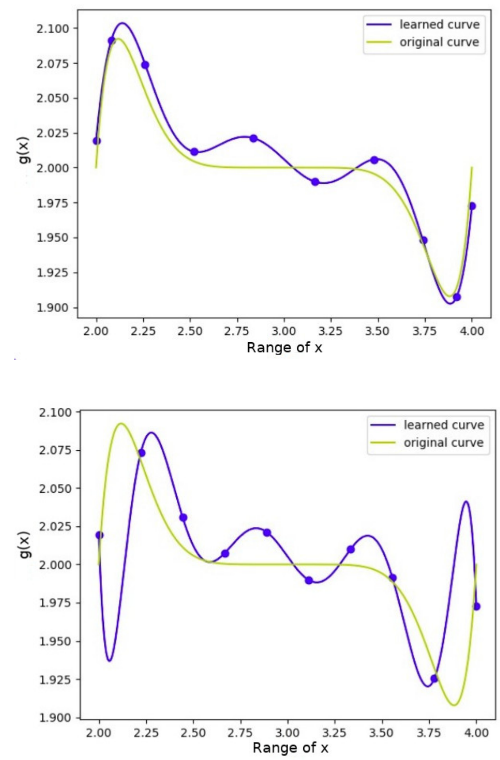

Figure 4: Polynomial regression with training points as barycentric spanner (top) and equidistant points (bottom) in a 9th degree polynomial.

Conclusions

We have shown that the barycentric spanner for a decision space arising from univariate polynomial cost functions can be efficiently computed using convex optimization. We have illustrated the applicability of our results through a dynamic pricing problem involving a polynomial demand curve, and empirically showed that using the barycentric spanner for initializing covariance updates within a Thompson sampling algorithm leads to lower regret. We have also provided theoretical and empirical results to show that a barycentric spanner achieves the least worst case expected MSE in an adversarial setting.

References

Abernethy, J. D.; Hazan, E.; and Rakhlin, A. 2012. Interior-point methods for full-information and bandit online learning. *IEEE Transactions on Information Theory* 58(7): 4164–4175.

Agrawal, S.; and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 127–135.

Awerbuch, B.; and Kleinberg, R. 2008. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences* 74(1): 97–114.

Bagnell, D. 2005. Bayesian Linear Regression Description. http://www.cs.cmu.edu/~16831-f14/notes/F14/16831_lecture20_jhua_dkambam.pdf. Accessed: 2020-06-08.

Bartlett, P. L.; Dani, V.; Hayes, T.; Kakade, S.; Rakhlin, A.; and Tewari, A. 2008. High-probability regret bounds for bandit online linear optimization. In *COLT*.

Bernstein, D. S. 2018. *Scalar, Vector, and Matrix Mathematics: Theory, Facts, and Formulas*. Princeton University Press, revised edition. ISBN 9780691151205. URL http://www.jstor.org/stable/j.ctvc7729t.

Boyd, S.; and Vandenberghe, L. 2004. *Convex Optimization*. Berichte über verteilte messysteme. Cambridge University Press. ISBN 9780521833783.

Bubeck, S.; Cesa-Bianchi, N.; and Kakade, S. M. 2012. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, 41–1.

Chen, S.; and Moitra, A. 2019. Beyond the low-degree algorithm: mixtures of subcubes and their applications. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 869–880.

Dani, V.; and Hayes, T. P. 2006. Robbing the Bandit: Less Regret in Online Geometric Optimization against an Adaptive Adversary. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '06, 937–943. USA: Society for Industrial and Applied Mathematics. ISBN 0898716055.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory*.

Dani, V.; Kakade, S. M.; and Hayes, T. P. 2008. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems*, 345–352.

den Boer, A. 2015. Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in operations research and management science* 20(1): 1–18. doi:10.1016/j.sorms.2015.03.001.

Diamond, S.; and Boyd, S. 2016. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research* URL http://stanford.edu/~boyd/papers/pdf/cvxpy_paper.pdf. To appear.

Ganti, R.; Sustik, M.; Tran, Q.; and Seaman, B. 2018. Thompson sampling for dynamic pricing. *arXiv preprint arXiv:1802.03050* .

Hannan, J. 1957. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games* 3: 97–139.

Hazan, E.; and Karnin, Z. 2016. Volumetric spanners: an efficient exploration basis for learning. *The Journal of Machine Learning Research* 17(1): 4062–4095.

Kalai, A.; and Vempala, S. 2005. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences* 71(3): 291–307.

Keskin, N. B.; and Zeevi, A. 2014. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research* 62(5): 1142–1167.

Kleinberg, R. D. 2005. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, 697–704.

McMahan, H. B.; and Blum, A. 2004. Online Geometric Optimization in the Bandit Setting Against an Adaptive Adversary. In *COLT*.

Nesterov, Y. E.; and Nemirovskii, A. S. 1994. *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA: SIAM.

Powell, M. J. D. 1970. A Hybrid Method for Nonlinear Equations. In Rabinowitz, P., ed., *Numerical Methods for Nonlinear Algebraic Equations*, 87–114. Gordon and Breach.

Supplementary material for 'Computing an Efficient Exploration Basis for Learning with Univariate Polynomial Features'

Proofs

The proof of Theorem 1 uses the following lemma.

Lemma 1. *Let $s \in [p_{\min}, p_{\max}]$ and suppose $p_1, \dots, p_{n+1} \in [p_{\min}, p_{\max}]$ are such that $p_i \neq p_j$ for all $i \neq j$. Then $c_1, \dots, c_{n+1} \in \mathbb{R}$ satisfy*

$$c_1 f_n(p_1) + \cdots + c_{n+1} f_n(p_{n+1}) = f_n(s) \tag{10}$$

if and only if $c_i = l_i(s, \mathbf{p})$ for each $i = 1, \dots, n+1$, where $\mathbf{p} = [p_1, \dots, p_{n+1}]^{\mathrm{T}}$, and $l_i(\cdot, \mathbf{p})$ is the ith Lagrange basis polynomial for the points $\{p_1, p_2, \dots, p_{n+1}\}$ given by

$$l_i(s, \mathbf{p}) \overset{\text{def}}{=} \frac{\prod_{j \neq i} (s - p_j)}{\prod_{j \neq i} (p_i - p_j)}. \tag{11}$$

Proof. Equation (10) may be rewritten as

$$V(\mathbf{p}) c(s) = f_n(s). \tag{12}$$

The determinant of the Vandermonde matrix $V(\mathbf{p})$ equals (see Fact 7.18.5 from Bernstein (2018))

$$\det(V(\mathbf{p})) = \prod_{1 \leq i < j \leq n+1} (p_j - p_i). \tag{13}$$

which is nonzero since $p_i \neq p_j$ for $j \neq i$. Equation (12) thus has a unique solution. Applying Cramer's rule (see Fact 3.16.12 from Bernstein (2018)) gives this solution to be

$$c_i = \frac{\det(V(\mathbf{p}_i^s))}{\det(V(\mathbf{p}))} \tag{14}$$

where $\mathbf{p}_i^s$ is the vector obtained by replacing the ith element of $\mathbf{p}$ by s. Using (13) to expand the determinants of the two Vandermonde matrices in (14) and canceling common terms gives $c_i = l_i(s, \mathbf{p})$. ☐

Proof of Theorem 1. To prove 1) implies 2), suppose the set $\{f_n(p_1), \dots, f_n(p_{n+1})\}$ is a barycentric spanner for D_n for some $\mathbf{p} = [p_1, \dots, p_{n+1}]^{\mathrm{T}} \in [p_{\min}, p_{\max}]^{n+1}$ such that $p_{\min} \leq p_1 \leq \cdots \leq p_{n+1} \leq p_{\max}$. Since the set D_n is not contained in any proper subspace of $\mathbb{R}^{n+1}$, the vectors $\{f_n(p_i)\}_{i=1}^{n+1}$ are all distinct. Hence, $p_i \neq p_j$ for $j \neq i$, and it follows that $p_{\min} \leq p_1 < \cdots < p_{n+1} \leq p_{\max}$.

Next, let $x = f_n(s)$ for some $s \in [p_{\min}, p_{\max}]$. By the definition of barycentric spanner, there exist $c(s) = [c_1(s), \dots, c_{n+1}(s)]^{\mathrm{T}} \in [-1, 1]^{n+1}$ such that $c_1(s) f_n(p_1) + \cdots + c_{n+1}(s) f_n(p_{n+1}) = f_n(s)$. By Lemma 1, $c_i(s) = l_i(s, \mathbf{p})$ for each i.

It now follows from the definition of a barycentric spanner that $|l_i(s, \mathbf{p})| \leq 1$ for all $s \in [p_{\min}, p_{\max}]$ and $i = 1, \dots, n+1$. On the other hand, directly substituting p_i in (11) gives $l_i(p_i, \mathbf{p}) = 1$ for all $i \in \{1, \dots, n+1\}$, showing that p_i is a local maximizer of $l_i(\cdot, \mathbf{p})$ for each i. We have already shown above that the points $p_2, \dots, p_n$ are necessarily interior points of the interval $[p_{\min}, p_{\max}]$. Hence first-order necessary conditions for optimality apply, and give

$$\left. \frac{\partial l_i(s, \mathbf{p})}{\partial s} \right|_{s = p_i} = 0, \; i = 2, \dots, n. \tag{15}$$

Using (11) in (15) directly yields (1). Next, note that

$$\left. \frac{\partial l_1(s, \mathbf{p})}{\partial s} \right|_{s = p_1} = \sum_{j \neq 1} \frac{1}{p_1 - p_j} < 0. \tag{16}$$

Hence, if $p_1 > p_{\min}$, then there exists $\epsilon > 0$ such that $p_1 - \epsilon \in [p_{\min}, p_{\max}]$ and $l_1(p_1 - \epsilon, \mathbf{p}) > l_1(p_1, \mathbf{p}) = 1$ which contradicts our earlier conclusion that $|l_1(s, \mathbf{p})| \leq 1$ for all $s \in [p_{\min}, p_{\max}]$. The contradiction shows that $p_1 = p_{\min}$. A similar argument shows that $p_{n+1} = p_{\max}$. This shows that 1) implies 2).

To show that 2) implies 3), consider a $\mathbf{p} \in \mathbb{R}^{n+1}$ as in the statement 2). On applying Proposition 1 with $k = n-1$, $a = p_1$, and $b = p_{n+1}$, we conclude that $z = [p_2, \dots, p_n]^{\mathrm{T}} \in \mathbb{R}^{n-1}$ is the unique global maximizer of the function U defined by (4). Comparing (13) with (4) shows that $\mathbf{p}$ is a maximizer of $\ln|\det(V(\cdot))|$ among all vectors $w \in \mathbb{R}^{n+1}$ satisfying $p_{\min} = w_1 < w_2 < \cdots < w_n < w_{n+1} = p_{\max}$. It follows that 2) implies 3).

To prove that 3) implies 1), suppose $\mathbf{p}$ is as in statement 3), and consider $s \in (p_{\min}, p_{\max})$. Arguing as in the proof of "1) implies 2)", we see that $c(s) \in \mathbb{R}^{n+1}$ defined by (14) satisfies $f_n(s) = c_1(s) f_n(p_1) + \cdots + c_n(s) f_n(p_{n+1})$. By the global optimality of $\mathbf{p}$, we have $|\det(V(\mathbf{p}_i^s))| \leq |\det(V(\mathbf{p}))|$, that is, $|c_i(s)| \leq 1$ for all i. This completes the proof. ☐

Proof of Proposition 1. First, observe that $C = \{z \in \mathbb{R}^k : a < z_1 < z_2 < \cdots < z_k < b\}$ is an open convex set. Note that the function $x \mapsto \ln|x - r|$ is continuously differentiable at $x \neq r$ with derivative $(x - r)^{-1}$. Using this observation, one can conclude that U is continuously differentiable on C, and calculate

$$\frac{\partial U}{\partial z_i}(z) = \frac{1}{z_i - a} + \sum_{j \neq i} \frac{1}{z_i - z_j} + \frac{1}{z_i - b}, \; i = 1, \dots, k. \tag{17}$$

We can differentiate (17), and further calculate

$$\frac{\partial^2 U}{\partial z_i^2} = \frac{-1}{(z_i - a)^2} - \sum_{j \neq i} \frac{1}{(z_i - z_j)^2} - \frac{1}{(z_i - b)^2}, \tag{18}$$

$$\frac{\partial^2 U}{\partial z_i z_j} = \frac{1}{(z_i - z_j)^2}, \tag{19}$$

for $i, j \in \{1, \dots, k\}$, $j \neq i$. The second-order mixed partial derivatives in (18) and (19) define the Hessian matrix $H(z)$ of U at $z \in C$. Applying the Gershgorin circle theorem (see Fact 6.10.22 from Bernstein (2018)) to $H(z)$ lets us conclude that $H(z)$ is negative definite for each $z \in C$. This implies that U is strictly concave on C.

We first show that U has a unique global maximizer in C. To show this, note that the function U is unbounded below.

For instance, $U \to -\infty$ as $z_1 \to a$. Hence we may choose $K \in \mathbb{R}$ such that the set $F \stackrel{\text{def}}{=} \{x \in C : U(x) \geq K\}$ is nonempty. We claim that F is closed in $\mathbb{R}^k$. To arrive at a contradiction, suppose F is not closed. Then there exists $x \in \mathbb{R}^k \backslash F$ and a sequence $\{x_l\}_{l=1}^{\infty}$ in F converging to x. Since $F \subseteq C$, x belongs to the closure of C. On the other hand, $x \notin C$, since otherwise the continuity of U on C would imply that $K \geq U(x_l) \to U(x)$, and contradict our assumption that $x \notin F$. Thus x lies in the closure of C, but not in C. It follows that x satisfies at least one of the inequalities defining C with equality. However, the definition of U then implies that the sequence $\{U(x_l)\}_{l=1}^{\infty}$ diverges to $-\infty$, contradicting our definition of F. This proves our claim that F is closed.

F is also bounded, and hence compact, as C itself is contained in the bounded set $[a, b]^k$. The continuous function U achieves its maximum over the compact set F at a point, say $z^* \in F$. By the definition of F, we have $U(z^*) \geq K$, while $U(z) < K \leq U(z^*)$ for all $z \in C \backslash F$. Thus we conclude that z^* is a global maximizer of U on C. Being strictly concave, U can have at most one global maximizer (Boyd and Vandenberghe 2004). It follows that z^* is the unique global maximizer of U on C.

Since C is open, first-order necessary conditions for optimality imply that the first-order partial derivatives of U given by (17) vanish at z^*. Thus, z^* is a solution to (3).

If $x \in C$ is any solution of (3), then, by (17), the gradient of U at x is zero, while the Hessian $H(x)$ is negative definite. By second-order sufficient conditions for optimality, x is a local maximizer for U. However, strict concavity implies that x is also a global maximizer of U on C. It now follows from the uniqueness of the global maximizer shown above that $x = z^*$. Thus z^* is the unique solution to (3).

Next, consider the point $x \in \mathbb{R}^k$ defined by setting $x_i = b + a - z^*_{k+i-1}$. It is a simple matter to check that $x \in C$, and verify by direct substitution that x satisfies (3). Since we have already shown that z^* is the unique solution to (3) in C, it follows that $x = z^*$. In other words, (5) holds. This completes the proof.

□

Reduced form of Equations (1) and (2) by exploiting symmetry

The relations (5) imply that the points p_i, $i = 1, \ldots, n+1$, yielding the barycentric spanner are symmetrically placed about the midpoint $\bar{p} \stackrel{\text{def}}{=} \frac{1}{2}(p_{\min} + p_{\max})$ of the interval $[p_{\min}, p_{\max}]$. Thus, it is sufficient to find points lying only on one side of the midpoint. This can be essentially achieved by using the symmetry relations (5) to eliminate (roughly) half the variables from (1) and (2). Next, we describe the reduced versions of (1) and (2) obtained by exploiting the symmetry inherent in (5).

First assume $n = 2l$ for some $l > 0$. Then $p_{l+1} = \bar{p}$ by symmetry, and solving (1) reduces to solving

$$\left[\sum_{\substack{j \neq i \\ 1 \leq j \leq l}} \frac{1}{p_i - p_j} + \frac{1}{p_i + p_j - 2\bar{p}}\right] + \frac{1}{p_i - \bar{p}} = 0, \tag{20}$$

for $i = 2, \ldots, l$. Likewise, optimizing (2) in the case $n = 2l$ reduces to optimizing the function

$$\begin{aligned}\bar{U}(p) = \ln \Bigg| &\left(\prod_{i=2}^{l} (a - p_i)^2 (b - p_i)^2 (p_i - \bar{p})^3\right) \\ &\times \left(\prod_{2 \leq i < j \leq l} (p_i - p_j)^2 (p_i + p_j - 2\bar{p})^2\right)\Bigg|,\end{aligned} \tag{21}$$

on the set $p_{\min} < p_1 < \ldots < p_l < \bar{p}$. Next, assume $n = 2l + 1$ for some $l > 0$. In this case, a solution of (1) can be recovered by solving

$$\sum_{\substack{j \neq i \\ 1 \leq j \leq l+1}} \frac{1}{p_i - p_j} + \frac{1}{p_i + p_j - 2\bar{p}} = 0, \; i = 2, \ldots, l+1, \tag{22}$$

while the optimizer in (2) can be found by optimizing

$$\begin{aligned}\bar{U}(p) = \ln \Bigg| &\left(\prod_{i=2}^{l+1} (a - p_i)^2 (b - p_i)^2 (p_i - \bar{p})\right) \\ &\times \left(\prod_{2 \leq i < j \leq l+1} (p_i - p_j)^2 (p_i + p_j - 2\bar{p})^2\right)\Bigg|,\end{aligned} \tag{23}$$

on the set $p_{\min} < p_1 < \ldots < p_l < \bar{p}$.

Proof of Proposition 2

In order to prove Proposition 2, we first prove the following result.

Proposition 3. *A barycentric spanner for the set D solves the following minmax problem.*

$$\min_{x_1, \ldots, x_d \in D} \max_{z \in D} \|X^{-1} z\|_\infty. \tag{24}$$

Proof. Given a subset $\{x_1, \ldots, x_d\}$ of D and $X = [x_1, \ldots, x_d] \in \mathbb{R}^{d \times d}$, letting $z = x_1$ gives $\|X^{-1}z\|_\infty^2 = \|e_1\|_2^2 = 1$. Thus $\max_{z \in D} \|X^{-1}z\|_\infty^2 \geq 1$ for all choices of X. On the other hand, if $\{x_1, \ldots, x_d\}$ is a barycentric spanner for D, then $\|X^{-1}z\|_\infty^2 \leq 1$ for all $z \in D$. This proves that a barycentric spanner solves (24). □

Proof of Proposition 2: The expected mean-square testing error on the test points is

$$\frac{1}{k} \sum_{i=1}^{k} \mathbb{E}[\hat{g}(z_i) - g(z_i)]^2 = \frac{\sigma^2}{k} \operatorname{tr}(Z^{\mathrm{T}} (X X^{\mathrm{T}})^{-1} Z), \tag{25}$$

where $Z \stackrel{\text{def}}{=} [z_1, \ldots, z_k] \in \mathbb{R}^{d \times k}$. The learner's goal is to choose X such that the worst case value of the expected mean-square testing error in (25) over the adversary's choice of Z is minimized.

Let $\Lambda_1 \in \mathbb{R}^{d \times d}$ denote the diagonal matrix having σ_i as its ith diagonal entry for each i. Note that if $\epsilon = [\epsilon_1, \ldots, \epsilon_d]$, then $\mathbb{E}(\epsilon\epsilon^{\mathrm{T}}) = \Lambda_1^2$. Using this along with (7) gives

$$\begin{aligned}
&\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}[\hat{g}(z_i) - g(z_i)]^2 \\
&= \frac{1}{k}\mathrm{tr}[Z^{\mathrm{T}}(XX^{\mathrm{T}})^{-1}X\Lambda_1^2X^{\mathrm{T}}(XX^{\mathrm{T}})^{-1}Z] \\
&= \frac{1}{k}\mathrm{tr}(Z^{\mathrm{T}}X^{-\mathrm{T}}\Lambda_1^2X^{-1}Z) = \frac{1}{k}\sum_{j=1}^{k}\|\Lambda_1X^{-1}z_j\|_2^2 \\
&= \frac{1}{k}\sum_{j=1}^{k}[\sigma_1^2(e_1^{\mathrm{T}}X^{-1}z_j)^2 + \cdots \\
&\quad + \sigma_d^2(e_d^{\mathrm{T}}X^{-1}z_j)^2]. \qquad (26)
\end{aligned}$$

It is easy to see from (26) that the adversary can ensure the worst case expected mean-square error for a given choice of X by setting $k = 1$, computing $(i^*, z^*) = \mathrm{argmax}_{i,z}|e_i^{\mathrm{T}}X^{-1}z|$, and setting $z_1 = z^*$, $\sigma_{i^*} = \sigma$ and $\sigma_i = 0$ for all $i \neq i^*$. Note that by definition $|e_{i^*}^{\mathrm{T}}X^{-1}z_*| = \max_{z\in D}\|X^{-1}z\|_\infty$. It is now evident from Proposition 3 that the learner can minimize the worst case expected mean-square error forced by the adversary by choosing the training points to form a *barycentric spanner* for the set D. $\square$