Computing an Efficient Exploration Basis for Learning with Univariate Polynomial Features

Chaitanya Amballa,[1] **Manu K. Gupta,**[2*] **Sanjay P. Bhat** [1]
[1]TCS Research, Hyderabad, India
[2]Department of Management Studies, Indian Institute of Technology, Roorkee, India
{chaitanya.amballa, sanjay.bhat}@tcs.com, manu.gupta@ms.iitr.ac.in

Abstract

Barycentric spanners have been used as an efficient exploration basis in online linear optimization 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 optimization problem. We provide numerical results to show that our method computes the barycentric spanner for the polynomial case 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 prior distribution 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 controls the variance of the noise corrupting the training samples.

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. In contrast, only the cost of the last decision made by the learner is revealed after each round in case of the bandit version of the problem.

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 "test" point $z \in \mathbb{R}^d$ is easily seen to be $\hat{\mu}^{\mathrm{T}} z - \mu^{\mathrm{T}} z = \epsilon^{\mathrm{T}} c(z)$, where $c(z) = X^{-1} z$ is the vector of coefficients required

*The second author was with TCS Research when this work was performed.

to write z as a linear combination of the basis elements. It is evident that the error in predicting the function at a general point z depends on the "size" of the coefficients needed to express z 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 in 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 a decision set of the form 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 an application 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. Based on the learner's choice, an adversary selects points for testing the learner's fit, and chooses the variance of the noise corrupting each training sample subject to a constraint on the total variance. The learner's goal is to choose training points to minimize the worst-case expected mean square testing error forced by the adversary's choices.

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 indicate how symmetry properties can be exploited to further reduce the computations.
3. We empirically show that initialising the mean and 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 per-

formance 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 elements of the barycentric spanner as training points, 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 $D \subset \mathbb{R}^d$, 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 $w = [w_1, \ldots, w_{n+1}]^{\mathrm{T}} \in \mathbb{R}^{n+1}$, $V(w) \stackrel{\text{def}}{=} [f_n(w_1), \ldots, f_n(w_{n+1})]$ 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 $D_n \stackrel{\text{def}}{=} \{f_n(p) : p \in [p_{\min}, p_{\max}]\} \subset \mathbb{R}^{n+1}$ for some $n \geq 1$. The motivation for considering this particular set follows from the discussion given in the introduction.

Main Results

Our main result below gives two characterizations for the barycentric spanner of the set D_n. The proofs of all results in this section are given in the supplementary material (see Amballa, Gupta, and Bhat (2020)).

Theorem 1. *Suppose* $\mathbf{p} = [p_1, \ldots, p_{n+1}]^{\mathrm{T}} \in \mathbb{R}^{n+1}$ *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* $C \stackrel{\text{def}}{=} \{z \in \mathbb{R}^k : a < z_1 < z_2 < \cdots < z_k < b\}$. *Then the set of equations*

$$\frac{1}{z_i - a} + \sum_{j \neq i} \frac{1}{z_i - z_j} + \frac{1}{z_i - b} = 0, \; i = 1, \ldots, k, \tag{3}$$

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*

$$U(z) \stackrel{\text{def}}{=} \ln \left| \left(\prod_{i=1}^{k} (a - z_i)(b - z_i) \right) \left(\prod_{\substack{i=1,\ldots,k; \\ j>i,}} (z_i - z_j) \right) \right|. \tag{4}$$

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, (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 only once for any given value of n.

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). The resulting simplified versions of (1) and (2) are given in the supplementary material (see Amballa, Gupta, and Bhat (2020)). Assuming $p_{\min} = 0$ and $p_{\max} = 1$, the simplified equations can be solved analytically to obtain $(p_1, p_2, p_3) = \left(0, \frac{1}{2}, 1\right)$, $(p_1, p_2, p_3, p_4) = \left(0, \frac{1}{2} - \sqrt{\frac{1}{20}}, \frac{1}{2} + \sqrt{\frac{1}{20}}, 1\right)$

and $(p_1, p_2, p_3, p_4, p_5) = \left(0, \frac{1}{2} - \sqrt{\frac{3}{28}}, \frac{1}{2}, \frac{1}{2} + \sqrt{\frac{3}{28}}, 1\right)$ for the cases $n = 2,\ 3$ and 4, respectively.

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 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 (see Amballa, Gupta, and Bhat (2020)). 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 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).

Bandit Formulation

Consider a seller selling a certain 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 $0 \le p_{\min} < p_{\max} < \infty$. The seller observes demand d_t according to the noisy linear demand model given by $d_t = \alpha - \beta p_t + \epsilon_t$ for $t = 1, 2, \ldots, T$, where $\alpha, \beta > 0$ represent the unknown parameters of the demand model, and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ represents unobserved demand perturbations. The linear demand model is often considered in literature (see Keskin and Zeevi (2014)). 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 involve a higher degree polynomial dependence of revenue on the price. Hence, we consider the revenue to be of the general form $r(p_t) = g(p_t) + \epsilon_t$, 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 seller'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)).

Thompson Sampling Algorithm

TS begins by putting a prior distribution over the unknown parameters. We choose the prior over the parameter vector $\tilde{\mu} = [\tilde{\mu}_0, \tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_n]$ to be multi-variate Gaussian with mean vector μ_0 and covariance matrix A_0. In this case, the posterior distribution over $\tilde{\mu}$ at time step $t + 1$ is also multivariate Gaussian with mean vector $\mu_t \in \mathbb{R}^{n+1}$ and covariance matrix $A_t \in \mathbb{R}^{n+1 \times n+1}$ given by the update equations (see Bagnell (2005) and Chapter 3 of Bishop (2006))

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

where $x_{t+1} = f_n(p_{t+1})$. 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_n. 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 $\epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_{n+1}]^{\mathrm{T}}$, where $\epsilon_1, \epsilon_2, \ldots, \epsilon_{n+1}$ are the noise samples hidden in the data. As (7) in the next section shows, performing a least squares fit at points of the barycentric spanner gives $\mu = \tilde{\mu} + (BB^{\mathrm{T}})^{-1} B\epsilon$. 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 mean $\mu_0 = \mu$ and covariance matrix $A_0 = \sigma^2 (BB^{\mathrm{T}})^{-1} = \sigma^2 \left(\sum_{i=1}^{n+1} b_i b_i^{\mathrm{T}}\right)^{-1}$.

Since the value of σ may not be known in applications, we treat it as a parameter and apply the updates (6) as well as the initialization described above with $\sigma = 1$ in Algorithm 1 below. The confidence-ball-based algorithm for stochastic bandit linear optimization given by Dani, Hayes, and Kakade (2008) also uses the same updates as (6) along with the above initialization for A_0^{-1} with $\sigma = 1$.

The initialization steps 2-4 in Algorithm 1 query the unknown revenue curve at barycentric points, fit a least squares model, and initialize the Thompson sampling iterations executed by the while loop. The algorithm relies on (6) for learning, and uses Thompson sampling for suggesting the new price at each iteration. Specifically, at each round t,

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 using different methods.

Algorithm 1 Thompson sampling for dynamic pricing

Input: Model degree n, total_iterations
Initialization:
Step 1. Find the barycentric spanner $b_1, \ldots, b_{n+1}$ for D_n.
Step 2. Suggest each barycentric spanner point as a price, perform a least squares fit on resulting data to obtain μ_0.
Step 3. Set $\sigma = 1$, $A_0^{-1} = \sigma^{-2} \sum_{i=1}^{n+1} b_i b_i^{\mathrm{T}}$.
Step 4. Set $t = 0$.
while $t \leq$ total_iterations **do**
Sampling: Sample $w_t \sim \mathcal{N}(\mu_t, A_t)$ and form the sampled revenue curve $h_t(\cdot) = w_t^{\mathrm{T}} f_n(\cdot)$.
Optimization: Find the optimal price for the sampled revenue curve: $p_t = \arg\max_{p_{\min} \leq p \leq p_{\max}} h_t(p)$.
Decision: Apply price p_t and observe noisy revenue r_t
Learning: $A_{t+1}^{-1} = A_t^{-1} + \sigma^{-2} x_t x_t^{\mathrm{T}}$, $A_{t+1}^{-1}\mu_{t+1} = A_t^{-1}\mu_t + \sigma^{-2} r_t x_t$ with $x_t = f_n(p_t)$.
Set $t \leftarrow t + 1$.
end while

the algorithm samples a parameter vector from the current posterior distribution, computes the optimal price p_t for the sampled parameter vector, suggests the price p_t, observes the noisy revenue r_t returned by the environment, and uses the observation to update the posterior distribution according to (6).

Simulation Results

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

We 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. **Robustness to assumed model degree:** We performed a wide range of experiments in which the degree of the true revenue function was different from the one assumed in the algorithm. The first plot in Figure 2 shows a typical result.
2. **Robustness to polynomial assumption:** 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 model assumed in the algorithm is a fourth degree polynomial, while the true revenue at a price $p \in [1, 10]$ is given by $100e^{-(p-5)^2/20} + \epsilon$, $\epsilon \sim \mathcal{N}(0, 9)$.

Linear Regression: Adversarial Setting

Problem Setting and Main Result

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 subject to the constraint $\sigma_1^2 + \cdots + \sigma_d^2 \leq \sigma^2$ for a given $\sigma > 0$. The adversary also chooses k test points $z_1, \ldots, z_k \in D$ at which the linear fit obtained by the learner is tested. The adversary makes all his choices after observing the training points chosen by the learner. The adversary's goal is to force the highest possible value for the expected mean square testing error by choosing

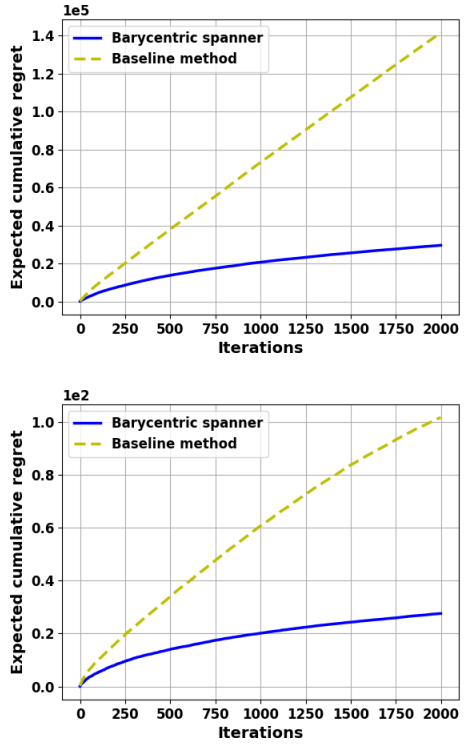

Figure 1: Expected cumulative regret comparison for a fourth (top) and a second (bottom) degree revenue function for the linear demand model considered by Keskin and Zeevi (2014).

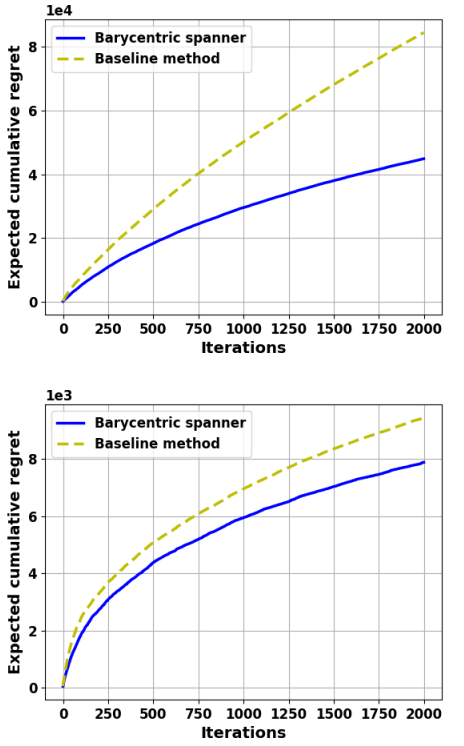

Figure 2: Expected cumulative regret comparison for 4th degree polynomial learnt using a 7th degree model (top) and a radial basis function when learnt by assuming a 4th degree model (bottom).

the noise variances, the number of test points k, and the test points themselves.

Let $X \stackrel{\text{def}}{=} [x_1, \ldots, x_d] \in \mathbb{R}^{d\times d}$, $y = [y_1, \ldots, y_d]^{\mathrm{T}} \in \mathbb{R}^d$, and $\epsilon = [\epsilon_1, \ldots, \epsilon_d]^{\mathrm{T}}$. 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

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

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 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. The following result, whose proof is given in the supplementary material (see Amballa, Gupta, and Bhat (2020)), states that the learner can achieve her goal by choosing the elements of a barycentric spanner as training points.

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

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

where $e_1, \ldots, e_d$ are column vectors of the $d \times d$ identity matrix. Moreover, the learner can minimize the worst-case (over the adversary's choices) expected MSE by choosing elements of a barycentric spanner of D as training points.

We performed several numerical experiments related to Proposition 2 for the polynomial case considered in the previous two sections. 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,

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

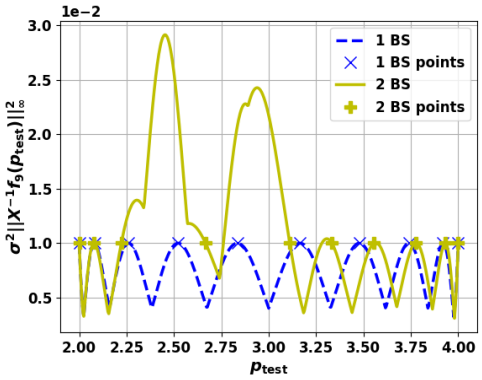

Figure 3: Worst-case value of the right hand side of (8) as a function of the test point $z = f_9(p_{\text{test}})$, $p_{\text{test}} \in [2, 4]$, that results from using barycentric spanner (BS) points and 2-approximate barycentric spanner (2 BS) points for training.

for $p \in [2, 4]$, 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 $p_{\text{test}} \in [2, 4]$, since an adversary can always choose the worst-case test point every time in the case of multiple testing opportunities. It is easy to see that, for $k = 1$, the largest value (over admissible choices of the noise variances) of the expected MSE (8) at a test point z is $\sigma^2\|X^{-1}z\|_\infty^2$. Recall that, in this case, $X = V(\mathbf{p})$ and $z = f_9(p_{\text{test}})$, with $\mathbf{p} \in \mathbb{R}^{10}$ being the vector of training points chosen by the learner.

Figure 3 shows a comparison of $\sigma^2\|X^{-1}f_9(p_{\text{test}})\|_\infty$ as a function of the test point p_{test} 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 value (represented by the maximum value of the plot) is 0.01 when training is performed at a barycentric spanner (marked with blue crosses in Figure 3), and 0.0291 when training is performed at a 2-approximate 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. 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 the 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 among the three choices. The same behavior can also be visually noticed in Figure 4, which depicts plots of the polynomials learned in one of the 500 trials summarized in Table 2.

Training ↓ \ Testing →	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)

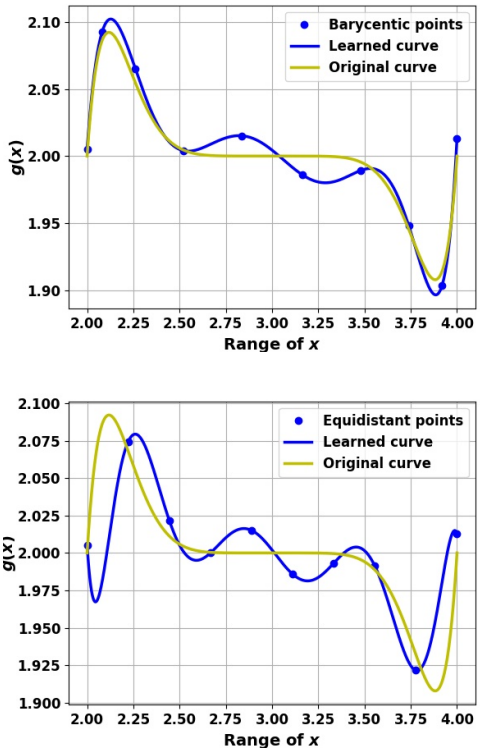

Figure 4: Polynomial regression with training points as barycentric spanner (top) and equidistant points (bottom) for the 9th degree polynomial (9).

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 shown that using the barycentric spanner for initializing the prior distribution 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 linear regression setting. We plan to extend the results to multivariate polynomials and explore applications to multi-product dynamic pricing in the future.

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.

Amballa, C.; Gupta, M. K.; and Bhat, S. P. 2020. Supplementary material for "Computing an Efficient Exploration Basis for Learning with Univariate Polynomial Features". Available at https://manugupta-or.github.io/Slides/DP_spanner.pdf.

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. http://www.cs.cmu.edu/~16831-f14/notes/F14/16831_lecture20_jhua_dkambam.pdf. Accessed: 2020-06-08.

Bartlett, P.; Dani, V.; Hayes, T.; Kakade, S.; Rakhlin, A.; and Tewari, A. 2008. High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory-COLT 2008*, 335–342.

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.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer. ISBN 9780387310732.

Boyd, S.; and Vandenberghe, L. 2004. *Convex Optimization*. 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*. JMLR Workshop and Conference Proceedings.

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)*, volume 6, 937–943.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory-COLT 2008*, 355–366.

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. V. 2015. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science* 20(1): 1–18.

Diamond, S.; and Boyd, S. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research* 17(1): 2909–2913.

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.

McMahan, H. B.; and Blum, A. 2004. Online geometric optimization in the bandit setting against an adaptive adversary. In *International Conference on Computational Learning Theory*, 109–123. Springer.

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.