Fast Algorithms for Monotone Lower Subsets of Kronecker Least Squares Problems

Osman Asif Malik^{*1}, Yiming Xu^{*2}, Nuojin Cheng^{*3}, Stephen Becker³, Alireza Doostan⁴, and Akil Narayan⁵

¹Applied Mathematics & Computational Research Division, Lawrence Berkeley National Laboratory (oamalik@lbl.gov) ²Corporate Model Risk, Wells Fargo (yiming.xu@wellsfargo.com)

³Department of Applied Mathematics, University of Colorado Boulder (nuojin.cheng@colorado.edu,

stephen.becker@colorado.edu)

⁴Smead Aerospace Engineering Sciences Department, University of Colorado Boulder (alireza.doostan@colorado.edu) ⁵Scientific Computing and Imaging Institute, and Department of Mathematics, University of Utah (akil@sci.utah.edu)

Abstract

Approximate solutions to large least squares problems can be computed efficiently using leverage score-based row-sketches, but directly computing the leverage scores, or sampling according to them with naive methods, still requires an expensive manipulation and processing of the design matrix. In this paper we develop efficient leverage score-based sampling methods for matrices with certain Kronecker product-type structure; in particular we consider matrices that are monotone lower column subsets of Kronecker product matrices. Our discussion is general, encompassing least squares problems on infinite domains, in which case matrices formally have infinitely many rows. We briefly survey leverage score-based sampling guarantees from the numerical linear algebra and approximation theory communities, and follow this with efficient algorithms for sampling when the design matrix has Kronecker-type structure. Our numerical examples confirm that sketches based on exact leverage score sampling for our class of structured matrices achieve superior residual compared to approximate leverage score sampling methods.

1 Introduction

Efficient algorithms for computing approximate solutions to large linear least squares problems is a well-studied topic. With $\mathbf{A} \in \mathbb{C}^{M \times N}$ and $\mathbf{b} \in \mathbb{C}^M$, the original, full least-squares problem is

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\arg\min} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2.$$
(1.1)

Algorithms that approximately generate solutions typically seek to overcome either the problems of big data (when M is so large that direct manipulation of \boldsymbol{A} is infeasible) or expensive data (when evaluating every entry of \boldsymbol{b} is too expensive). One standard, general approach involves constructing a sketching operator $\boldsymbol{S} \in \mathbb{C}^{K \times M}$ and solving the corresponding sketched problem,

$$\tilde{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{S}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{S}\boldsymbol{b}\|_2.$$
(1.2)

As long as computing SA and Sb can be done efficiently, then if K < M the resulting problem is smaller; yet if K < M then one cannot deterministically generate an operator S that produces an accurate solution for generic b. On the other hand, *randomly* generating S allows one to state error bounds either in expectation or with high probability.

^{*}Equal contribution.

One well-known approach that fits into the category described above is to construct S as a random row sketch via leverage scores of A, where one has guarantees of the form,

$$K \gtrsim \frac{N \log N}{(\delta + \epsilon)\epsilon} \implies \|A\tilde{x} - b\|_2^2 \le (1 + \epsilon) \|Ax^* - b\|_2^2 \text{ w/ probability } \ge 1 - \delta$$

The major remaining challenge in these cases is to develop algorithms that *efficiently* sample rows according to the leverage scores of A. Direct computation of the leverage scores involves computing an orthonormal basis for the range of A, which is expensive when M is very large. Typically two avenues are available to mitigate this cost: (i) sample from approximate leverage scores that are efficiently computable, e.g., [Dri+12]; or (ii) develop algorithms to compute exact leverage scores efficiently when A has exploitable structure.

Our focus in this paper is on the second approach: the development of algorithms that efficiently sample according to leverage scores when \boldsymbol{A} is a "monotone lower" column subset of a Kronecker product matrix. In particular, if $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(D)}$ are D matrices, with $\boldsymbol{A}^{(d)} \in \mathbb{R}^{M_d \times N_d}$, then we consider \boldsymbol{A} constructed as,

$$\boldsymbol{A}_{\mathrm{kr}} \coloneqq \bigotimes_{d=1}^{D} \boldsymbol{A}^{(d)}, \quad \boldsymbol{A} \text{ is a "monotone lower" column subset of } \boldsymbol{A}_{\mathrm{kr}}, \tag{1.3}$$

where \otimes is the Kronecker product, and "monotone lower" refers to the collection of *D*-dimensional integer indices that index columns of A_{kr} via their coordinate representations in the columns of $A^{(d)}$; see Definition 3.1. This particular structure in A is of considerable interest in high-dimensional function approximation when one builds emulators or surrogates from function spaces, e.g., polynomials [ABW22].

The strategy of leverage score sampling is well-established in the numerical linear algebra community [Mah11; Dri+12; Woo14]. An essentially identical strategy but in the cases when M and N are infinite has been developed in the functional analysis literature with various names: induced sampling, optimal sampling, Christoffel sampling, etc. [HD15; CM17a]. One of our goals in this paper is to present a discussion of leverage score/induced sampling that unifies these concepts and results in different communities.

1.1 Contributions of this article

The main target of this article is efficient computation of solutions to least squares problems that are formulated from data either on finite or un/countably infinite tensor-product domains in D dimensions, which corresponds to the Kronecker product setup mentioned above. In particular, we study such problems when the solution function is sought on a *subspace* of a tensor-product space of functions. When the domain is finite, this amounts to a large but finite matrix-vector least squares problem where the design matrix is a *column subset* of a Kronecker product matrix. Our main approach is a fairly standard one: randomized row sketches. Our contributions are twofold:

- We collect and formalize results guaranteeing near-optimal solutions (in terms of residual) to sparse row-sketched least squares problems.¹ Such results are already known for infinite domains in the approximation theory community, and for finite domains in the theoretical computer science and numerical linear algebra communities. Our presentation is unified, collecting and summarizing results from both communities. In the context of randomized sparse row sketches, the approach we consider is well-known as *optimal* or *induced* sampling in approximation theory, and *leverage score* sampling in the linear algebra community. This unified discussion is the topic of Section 2.
- We present efficient algorithms that can efficiently and *exactly* sample according to the distribution mentioned above for certain types of product-*structured* least squares problems. In the infinite domain case, product structure refers to *D*-dimensional tensor-product constructions, and when the domain is finite product structure refers to *D*-fold Kronecker product structure of the design matrix, cf. (1.3). Direct approaches for this sampling in these cases typically expensive when D > 1, but the approaches we describe have complexity scaling *linearly* in *D*. Our main contribution here

¹"Sparse" here means that we need not access the entire right-hand side data function/vector, instead requiring only a relatively small number of samples. This is in contrast to, e.g., dense Gaussian sketches that can also boast near-optimality but require knowledge of the data \boldsymbol{b} over the entire domain.

is an efficient algorithm when the subspace under consideration is "monotone lower". In the finite domain case, this corresponds to the assumption that the least squares design matrix is a monotone lower column subset of a Kronecker product matrix. Our algorithms in this case are described in Section 4, and in particular Algorithm 5.

An outline of this document is as follows: Section 2 provides our notation and problem setup, with Section 2.1 describing row-sketched least squares. Section 2.2 discusses a well-known $M = \infty$ generalization of leverage scores, and Section 2.3 surveys known results on leverage score-based accuracy of row-sketched least squares. Section 3 specializes our setup to the case of approximating functions of $D \ge 1$ independent variables, which corresponds to the Kronecker product matrix structured described previously. Section 4 discusses algorithms, with our proposed algorithm for monotone lower sets given in Section 4.3.3. Finally, we provide numerical examples in Section 5.

1.2 Related work on structured matrix sketching

A number of previous works develop various random sketches S that can be applied particularly efficiently to matrices A whose columns have Kronecker product structure, i.e., are of the form

$$\boldsymbol{a}_1 \otimes \boldsymbol{a}_2 \otimes \cdots \otimes \boldsymbol{a}_D,$$
 (1.4)

where each $\boldsymbol{a}_d \in \mathbb{R}^{M_d}$.

Row-structured sketches The first class of such sketches impose certain structure on the rows of S which allows it to be applied efficiently to vectors of the form in (1.4). [BBB15] propose such a sketch for which each row of S is the Kronecker product of D row vectors, each containing iid random variables with zero mean and unit variance. [Sun+18] independently propose a sketch with the same structure as well as a variance reduced version. [RR20] take these ideas further by proposing sketches whose rows are vectorized tensors in either canonical polyadic (CP) or tensor train formats and whose factor matrices and cores tensors have iid Gaussian entries with appropriate scaling. The follow-up paper [RR21] further shows that the sketch with tensor train-structured rows also performs well when its core tensor entries are drawn from a Rademacher distribution. [Iwe+21] propose a two-stage sketch which first applies a sketch matrix which is the Kronecker product of smaller sketches, followed by another sketch that further reduces the embedding dimension. More background on these types of sketches is in the review [MT20, Sec. 9.4].

Kronecker fast Johnson–Lindenstrauss transforms The second class of structured sketches impose further structure on the standard fast Johnson–Lindenstrauss transform (FJLT) originally proposed by [AC09]. This additional structure makes applying the sketch to vectors of the form (1.4) even faster. These sketches are referred to as Kronecker fast Johnson–Lindenstrauss transforms (KFJLTs) or tensor subsampled randomized Hadamard transforms (TensorSRHTs) by some authors in the case when the Hadamard transform is used. The KFJLT was first proposed by [BBK18] for use in tensor decomposition algorithms, with further theoretical and empirical work done in [JKW20; MB20; BKW21].

TensorSketch The third type of sketch is the TensorSketch. It can be viewed as a more structured variant of the CountSketch developed in [CW17] that allow efficient application to vectors of the form (1.4). It was developed in a series of papers [Pag13; PP13; ANW14; Dia+18].

Recursive sketches A fourth class of structured sketches rely on recursively applying the sketches discussed above in order to achieve improved theoretical guarantees. [Ahl+20] propose two variants of such a procedure. It first applies either an independent CountSketch [CW17] or OSNAP sketch [NN13] to each vector \mathbf{a}_d in (1.4). In the case when D is a power of 2, the resulting sketched vectors are then recursively combined pairwise in $\log_2(D)$ steps using either TensorSketches or KFJLTs. The case when D is not a power of 2 is handled by appropriately augmenting the product in (1.4) using the canonical basis vector \mathbf{e}_1 .

[Son+21] propose a sketch inspired by [Ahl+20] but specifically adapted for more efficient application in the case when all vectors a_1, \dots, a_D are identical. This situation comes up when computing spectral approximations of the polynomial kernel. The procedure by [Ahl+20] can be illustrated using a binary tree with each node associated with the appropriate sketch. [MS22] generalize this idea by allowing for sketches with arbitrary graph structure, but limit the nodes to being Gaussian sketches.

Sampling-based sketches The fifth and final class of structured sketches are sampling-based. Unlike the sketches discussed above these are *non-oblivious* meaning that they rely on information from the matrix being sketched. These sketches have therefore been developed specifically for matrices with certain structure. Methods for Kronecker product matrices are developed in [Dia+19; FGF21] and methods for the Khatri-Rao product in [Che+16; LK20; WZ20; Mal22; WZ22; Che+20]. The papers [MB21; Mal22] develop methods for the design matrices which arises in algorithms for tensor ring decomposition whose columns are the sum of Kronecker products of vectors.

		When μ and μ_d have infinite support
[D]		The set $\{1, 2, \dots, D\}$ for $d \in \mathbb{N}$
μ,I		product probability measure on \mathbb{R}^D and $I = \operatorname{supp} \mu$
μ_d, I_d	(3.1)	Dimensionwise measures and domain, $\mu = \times_d \mu_d$ and $I = \times_d I_d$
$\ \cdot\ _{oldsymbol{w},p}$	(3.6a)	The <i>w</i> -weighted ℓ^p function for $0 \le p \le \infty$
${\mathcal J}$		A size-N, finite multi-index set in D dimensions, i.e., a finite subset of \mathbb{N}^D
$\mathcal{J}_{\boldsymbol{w},p}(K)$	(3.7)	The 1 -centered ball in \mathbb{N}^D of $\ \cdot\ _{w,p}$ -radius K
$V, V(\mathcal{J})$	(3.5)	N-dimensional subspace of functions in $L^2_\mu(I)$ identified by $\mathcal J$
a_{lpha}	(3.5)	Product basis functions spanning V for $\alpha \in \mathcal{J}$
$V_d, a_j^{(d)}$	(3.2)	One-dimensional function spaces and functions, with $a_{\alpha} = \prod_{d} a_{\alpha_{d}}^{(d)}$
N_d, N	(3.2), (3.3)	$N_d = \dim V_d$, and $\boldsymbol{N} = (N_1, \dots, N_d)$
N	(3.4a)	$N = \dim V = \mathcal{J} $
$oldsymbol{x}^*$	(2.2)	$V\text{-}\mathrm{coefficients}$ for the least squares solution of $L^2_\mu\text{-}\mathrm{projecting}~f$ onto V
au	(2.4)	A finite sketch measure, intended to approximate μ for functions in V
$\widetilde{m{x}}$	(2.8)	Solution to the τ -sketched least squares problem
ν	(2.9)	The (μ, V) -induced measure (used to generate a random sketch τ)
$\nu^{(d)}$	(4.7)	The univariate (μ_d, V_d) -induced measure
		When μ and μ_d have finite support
$y_m^{(d)}, w_m^{(d)}$	(3.11)	The support points and weights, respectively, of μ_d
$oldsymbol{y}_m$		An enumeration of the finite points in I
$oldsymbol{A}^{(d)},oldsymbol{A}_{ m kr},$	(3.14), (3.15)	Dimension- d univariate design matrix, and Kronecker product of univariate matrices
$\boldsymbol{A}, \boldsymbol{b}$	(3.12), (3.16)	The design matrix and right-hand side for the least squares problem
$\ell_m^{(d)} = \nu^{(d)} \left(y_m^{(d)} \right)$		mth leverage score for $A^{(d)}$
$\ell_m = \nu\left(y_m\right)$		mth leverage score for \boldsymbol{A}

Table 1: Notation used throughout this article.

2 Preliminaries for function approximation

We describe least squares problems with unified notation that can be specialized to include projectionbased function approximation and matrix-vector least squares problems. Let $D \in \mathbb{N}$ and μ be a probability measure with closed support $I \subset \mathbb{R}^D$, defining an L^2 space:

$$L^2_{\mu}(I) = \left\{ u: I \to \mathbb{R} \mid \|u\| < \infty \right\}, \qquad \|u\|^2 \coloneqq \langle u, u \rangle, \qquad \langle u, v \rangle = \int_I u(\boldsymbol{y}) v(\boldsymbol{y}) \mathrm{d}\mu(\boldsymbol{y}).$$

Given an N-dimensional subspace $V \subset L^2_{\mu}(I)$ with $N < \infty$, we are abstractly interested in computing the $L^2_{\mu}(I)$ -best approximation from V to a given function $b \in L^2_{\mu}(I)$. Equivalently, we wish to approximate b using least squares:

$$v^* = \underset{v \in V}{\arg\min} \|v - b\|^2.$$
(2.1)

Let $(a_n)_{n \in [N]}$ be a(ny) basis for V. The above problem can also be written in terms of coordinates $(x_n)_{n \in [N]}$ in the basis a_n :

$$\boldsymbol{x}^* = \underset{\boldsymbol{x} \in \mathbb{R}^N}{\operatorname{arg\,min}} \left\| \sum_{n \in [N]} x_n a_n - b \right\|^2.$$
(2.2)

We provide two examples below to illustrate the generality of our problem setup. (See also sections 3.2 and 3.3.)

Example 2.1 (Finite-dimensional least-squares). Let $A \in \mathbb{R}^{M \times N}$ be a matrix with full column-rank, and $b \in \mathbb{R}^{M}$. Then the Euclidean least squares problem

$$\boldsymbol{x}^* = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^N} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2, \tag{2.3}$$

is a special case of the generic least squares problem Eq. (2.1)/(2.2) if we choose D = 1, I = [M], μ the discrete uniform measure on I = [M], i.e., $\mu = \sum_{m \in [M]} w_m \delta_m$ with weights $w_m = 1/M$ to set $a_n(m) = w_m^{-1/2}(\mathbf{A})_{m,n}$ and $b(m) = w_m^{-1/2}(\mathbf{b})_m$, and use the basis $(a_n)_{n \in [N]}$ for V, so that so that $V = \operatorname{col}(\mathbf{A})$.

Our approach generalizes to the case when A and b have an infinite number (countable or uncountable) of rows, in which case we view our problem as that corresponding to function approximation over an infinite domain. Note that such problems are discussed in [SA22] using the language of quasimatrices [TT15]; we approach the same problem here using the equivalent but more standard language of function spaces.

Example 2.2 (1D function approximation with polynomials). Let D = 1, I = [-1, 1] and μ any absolutely continuous measure (with respect to the Lebesgue measure), such as $\mu(y) = \frac{1}{2}$ or $\mu(y) = \frac{1}{\pi} \frac{1}{\sqrt{1-y^2}}$. Let V be the space of N-1 degree polynomials on I, and let (a_n) be any basis for V (such as Legendre or Chebyshev polynomials). In this setting, the generic problem Eq. (2.1)/(2.2) could be thought of as a least squares problem where the matrix **A** has an *uncountable* number of rows.

2.1 Randomly (row-)sketched least squares

In practical problems, the least squares formulation (2.2) (or equivalently (2.1)) cannot be directly solved in a computationally feasible way: The domain I may be un/countably infinite, so that exactly computing (2.2) is not possible. Even when μ is finitely supported at M points, so that we consider (2.3), the function b can be expensive to evaluate, so that collecting data $b(\mathbf{y}_m)$ for all m is too costly. From an algorithmic point of view, assuming $M \geq N$, direct solutions to (2.3) require $\mathcal{O}(MN^2)$ complexity, which one may wish to avoid if $M \gg N$.

In any of the cases above, one strategy to mitigate the cost of forming a least squares solution is to use randomly generated sketches. We will focus in particular on row sketches defined by a random measure τ , where τ is constructed through iid samples of a deterministic measure ν : Let ν be another probability measure on I that is equivalent to μ (i.e., $\nu \ll \mu$ and $\mu \ll \nu$). Define the following random, finitely supported *sketching* measure τ , which is constructed by randomly sampling from ν :

$$\tau = \sum_{k \in [K]} v_k \delta_{\mathbf{Y}_k}, \qquad \qquad v_k = \frac{1}{K} \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(\mathbf{Y}_k), \qquad \qquad \mathbf{Y}_k \stackrel{\mathrm{iid}}{\sim} \nu. \tag{2.4}$$

Above, we have fixed some positive integer K, and the weights v_k are chosen so that integrating with respect to τ is an unbiased estimator for integrating with respect to μ :

$$\mathbb{E}^{\otimes_{K}\nu} \int_{I} f(\boldsymbol{y}) \mathrm{d}\tau(\boldsymbol{y}) = K \mathbb{E}^{\nu} \left(f(\boldsymbol{Y}_{k}) v_{k} \right) = \int_{I} f(\boldsymbol{y}) \mathrm{d}\mu(\boldsymbol{y}).$$
(2.5)

We use τ to define corresponding norms and inner products, which are random:

$$\|u\|_{\tau}^{2} \coloneqq \langle u, u \rangle_{\tau}, \qquad \langle u, v \rangle_{\tau} = \int_{I} u(\boldsymbol{y}) v(\boldsymbol{y}) \mathrm{d}\tau(\boldsymbol{y}).$$

Since K is finite, then evaluating τ -integrals and -norms are computationally feasible and, when μ is finitely supported at M points, more tractable if $K \ll M$. In particular, we can consider a τ -sketched approximation to our least squares problem, which form problems analogous to (2.1) and (2.2):

$$\widetilde{v} \in \underset{v \in V}{\operatorname{arg\,min}} \left\| v - b \right\|_{\tau}^{2}, \qquad \qquad \widetilde{x} \in \underset{x \in \mathbb{R}^{N}}{\operatorname{arg\,min}} \left\| \sum_{n \in [N]} x_{n} a_{n} - b \right\|_{\tau}^{2}, \qquad (2.6)$$

where we have written " \in arg min" to acknowledge that, when replacing μ with the random sketch τ , solutions to the above problem may not be unique. The remainder of this section is concerned with determining when τ -sketched least squares solutions are of comparable quality (in terms of residual) to the full, frequently intractable least squares problem.

Regardless of the size of the support of μ , the procedure described above involves only finitedimensional quantities and is hence computable. To facilitate notation, we define

$$(\widehat{\boldsymbol{A}})_{k,n} = \sqrt{v_k} a_n(\boldsymbol{Y}_k), \qquad (\widehat{\boldsymbol{b}})_k = \sqrt{v_k} b(\boldsymbol{Y}_k), \qquad k \in [K], \ n \in [N].$$
(2.7)

Then \widetilde{x} in (2.6) is the solution to the finite least squares problem,

$$\widetilde{\boldsymbol{x}} \in \underset{\boldsymbol{x} \in \mathbb{R}^{N}}{\operatorname{arg\,min}} \left\| \widetilde{\boldsymbol{A}} \boldsymbol{x} - \widetilde{\boldsymbol{b}} \right\|_{2}^{2}.$$
(2.8)

This procedure is an algorithm, shown in Algorithm 1.

Algorithm 1: Least squares procedure from Section 2.1.			
In	put: Sampling size $K \in \mathbb{N}$, basis a_n for V , function b		
Input: Measure ν , Radon-Nikodym derivative $\frac{d\mu}{d\nu}$			
1	Sample $\boldsymbol{Y}_k \stackrel{\text{iid}}{\sim} \nu$ for $k \in [K]$.;		
2	Evaluate $v_k = \frac{1}{K} \frac{d\mu}{d\nu} (\mathbf{Y}_k)$ for $k \in [K]$.;		
3	Collect data $(\widetilde{\boldsymbol{b}})_k = b(\boldsymbol{Y}_k)$ for $k \in [K]$;		

4 Construct \widehat{A} and \widehat{b} as in (2.7) and solve (2.8).; Output: \widetilde{x} , the coordinates of \widetilde{v} in the basis a_n

When μ is finitely supported, as in Example 2.1, the measure τ is equivalent to constructing a sketching operator $\mathbf{S} \in \mathbb{R}^{K \times M}$; that is, sampling \mathbf{Y}_k according to ν is equivalent to randomly selecting row indices $i_k \in [M]$ according to the measure ν . A sketching matrix \mathbf{S} can be defined as,

$$oldsymbol{S} = \left(egin{array}{ccccc} & \sqrt{v_{i_1}} e_{i_1}^T & ---- \ & \sqrt{v_{i_2}} e_{i_2}^T & ---- \ & & \ddots & & & \ \end{pmatrix},$$

where $e_i \in \mathbb{R}^M$ is the cardinal unit vector in direction *i*. Then \widetilde{A} and \widetilde{b} are directly related to A and b:

$$\widetilde{A} = SA ext{ and } \widetilde{b} = Sb ext{ } \Longrightarrow x^* \in rgmin_x \|SAx - Sb\|_2^2.$$

2.2 The induced distribution: "optimal" least squares and leverage scores

The discussion in the previous section does not reveal how to appropriately choose ν or K. We first discuss the particular choice of ν we make for the remainder of this article, the *induced* distribution.

Definition 2.3. Given (μ, V) , with $V \subset L^2_{\mu}(I)$ and $N = \dim V$, the induced measure $\nu_{\mu,V}$ on I is defined as,

$$d\nu_{\mu,V}(\boldsymbol{y}) = \frac{\sup_{v \in V, \|v\|=1} |v(\boldsymbol{y})|^2}{N} d\mu(\boldsymbol{y}).$$
(2.9)

Loosely speaking, $\nu_{\mu,V}$ is biased toward locations where L^2_{μ} -unit-norm functions from V can have large values. While the definition of (2.9) appears somewhat opaque, a clearer expression can be derived. First, we introduce a(ny) $L^2_{\mu}(I)$ -orthonormal basis for V:

$$\langle u_n, u_\ell \rangle = \delta_{n,\ell}, \quad n, \ell \in [N], \qquad \qquad \text{span}\{u_n\}_{n \in [N]} = V. \tag{2.10}$$

The following result gives an equivalent expression for $\nu_{\mu,V}$.

Proposition 2.4. Fix μ and V. With $\nu = \nu_{\mu,V}$ defined through (2.9), then

$$\frac{\mathrm{d}\nu}{\mathrm{d}\mu}(\boldsymbol{y}) = \frac{1}{N} \sum_{n \in [N]} |u_n(\boldsymbol{y})|^2.$$
(2.11)

Proof. Since $(u_n)_{n \in [N]}$ is an orthonormal basis, then

$$v \in V, \|v\| = 1 \iff \exists \boldsymbol{x} \in \mathbb{R}^N, \|\boldsymbol{x}\| = 1, \text{ such that } v(\boldsymbol{y}) = \sum_{j=1}^N x_j u_j(\boldsymbol{y}).$$
 (2.12)

Then an application of the Cauchy-Schwarz (C-S) inequality finishes the proof:

$$\frac{\mathrm{d}\nu}{\mathrm{d}\mu}(\boldsymbol{y}) = \frac{\sup_{v \in V, \|v\|=1} |v(\boldsymbol{y})|^2}{N} \stackrel{(2.12)}{=} \frac{1}{N} \sup_{\boldsymbol{x} \in \mathbb{R}^N, \|\boldsymbol{x}\|_2 = 1} \left| \sum_{n \in [N]} x_n u_n(\boldsymbol{y}) \right|^2 \stackrel{\mathrm{C-S}}{=} \frac{1}{N} \sum_{n \in [N]} |u_n(\boldsymbol{y})|^2.$$

Note that the choice of orthonormal basis in (2.11) is arbitrary, as that expression remains invariant under any unitary transformation of $(u_n)_{n \in [N]}$. The well-known result (2.11) is one way to see that what we call the induced distribution above has several names in the literature. In functional analysis, the Radon-Nikodym derivative $\frac{d\nu_{\mu,V}}{d\mu}$ is called the (inverse) normalized Christoffel function [Nev86; Xu95], and also coincides with the normalized diagonal of the bivariate reproducing kernel for V in $L^2_{\mu}(I)$ [Sim08]. In pluripotential theory when V is the space of d-variate polynomials up to a fixed degree k, then $\frac{d\nu_{\mu,V}}{d\mu}$ is called the kth Bergman function [Ber09a; Ber09b; BBN11; DMN17]. For least squares problems, sampling from this measure is called "optimal sampling" because it optimizes (minimizes) a matrix Chernoff bound that dictates sample complexity, see [CM17b] and also Section A.1.

When μ is finitely supported, the values of the Radon-Nikodym derivative discussed above are exactly the statistical *leverage scores* of the matrix A [Woo14; Mah11].

Definition 2.5. Given $A \in \mathbb{R}^{M \times N}$, let $U \in col(A)$ be an $M \times r$ matrix, where col(A) denotes the set of all matrices whose columns form an orthonormal basis for range(A) and r = rank(A). Then the *normalized* leverage scores of A are defined as,

$$\ell_m = \frac{1}{r} \sum_{j \in [r]} \left| (U)_{m,j} \right|^2.$$
(2.13)

Using this definition along with Proposition 2.4, we immediately conclude the following.

Corollary 2.6. Consider the setup of Proposition 2.4. If μ is finitely supported on $I = \{\boldsymbol{y}_m\}_{m \in [M]}$, then $\ell_m = \frac{\mathrm{d}\nu_{\mu,V}}{\mathrm{d}\mu}(\boldsymbol{y}_m)$ for all m, i.e., the Radon-Nikodym derivative $\frac{\mathrm{d}\nu_{\mu,V}}{\mathrm{d}\mu}(\boldsymbol{y}_m)$ coincides with the normalized leverage score ℓ_m of \boldsymbol{A} defined in (2.13).

Thus, in this finitely supported setting, $\nu_{\mu,V}$ is sometimes called the leverage distribution, since its probability mass weights equal the leverage scores (of **A**). In such contexts for finite least squares problems, generating samples with respect to ν is called leverage score sampling or natural sampling.

Finally, we note that conditions ensuring that $\nu_{\mu,V}$ is equivalent to μ (required as discussed in section 2.1) can be rephrased given the formula (2.11): $\nu_{\mu,V}$ is equivalent to μ if and only if $\mu(S) = 0$, where $S = V^{(-1)}(0)$ is the zero set of the subspace V, i.e., $\boldsymbol{y} \in S$ if $v(\boldsymbol{y}) = 0$ for every $v \in V$. This property is true if, e.g., μ is Lebesgue measure and V contains only real-analytic functions [Mit20]. We assume this measure equivalence property in all that follows. However, this does exclude some cases, e.g., certain subspaces V of Haar-type wavelets.

2.3 Bounds for row-sketched least squares

In this section we summarize known sufficient conditions to ensure quantitative residual bounds for τ -sketched least squares solutions in (2.6). At their core, these sufficient conditions involve a lower bound for the size of K to be used in Algorithm 1 that depends on a desired probability of failure δ and a target relative accuracy tolerance ϵ . We present two choices for ν below (both induced distributions, but for different subspaces) that achieve useful bounds. All these bounds are essentially well-known in different communities, and so we present here only the statements, and leave the proofs in the appendix.

2.3.1 Sampling with $\nu = \nu_{\mu, V_b}$

We typically assume V is known or can be sampled (as in either Example 2.1 or 2.2), but even in the finite-dimensional case, it may be desirable not to know the whole target function b, such as when each entry of b requires expensive computation. Hence our first algorithm is an idealized one, as it involves a subspace augmented by the target function b:

$$V_b := \operatorname{span}\{V, b\}. \tag{2.14}$$

This is an idealized construction since it, naively, requires access to the full right-hand side function b; one of the common purposes of random sketching is to avoid full sampling of b. Nevertheless, note that if sampling is done only to reduce the flop count of the least-squares problem (when full knowledge of b is often assumed), then this algorithm is not so impractical. Sampling according to the induced distribution of V_b results in the following accuracy guarantees.

Theorem 2.7. Given $\delta \in (0,1)$, $\epsilon \in (0,1/2)$, and $b \in L^2_{\mu}(I)$, assume that

$$K \ge \frac{3\log\left(\frac{4(N+1)}{\delta}\right)}{\epsilon^2}(N+1). \tag{2.15}$$

Then with probability at least $1 - \delta$, the solution \tilde{v} computed from Algorithm 1 with sampling measure $\nu = \nu_{\mu,V_b}$ has a unique solution, and we have the relative error bound,

$$\|\widetilde{v} - b\|^2 \le (1 + 2\epsilon) \|v^* - b\|^2,$$
 (2.16)

which in the finite I case is equivalent to

$$\|A\widetilde{x} - b\|_{2}^{2} \le (1 + 2\epsilon)\|Ax^{*} - b\|_{2}^{2}$$

For the proof, see Appendix B.

2.3.2 Sampling with $\nu = \nu_{\mu,V}$

The more tractable strategy is to sample with $\nu = \nu_{\mu,V}$, which does not require any *a priori* knowledge about the data *b*. We present several known results in this section that are compiled from [Woo14; Mah11; CM17b]. In order to be self-contained, we collect the proofs of these results in appendices C, D, and E. The simplest result, stated below, is an instance-wise bound on the error committed by $\tilde{\nu}$.

Theorem 2.8. Given $\epsilon, \delta \in (0, 1)$ and $b \in L^2_{\mu}(I)$, assume that

$$K \ge \frac{N}{\epsilon} \max\left\{\frac{2}{\delta(1-\epsilon)^2}, \frac{3\log\left(\frac{4N}{\delta}\right)}{\epsilon}\right\}$$
(2.17)

Then, with probability at least $1 - \delta$, the solution \tilde{v} computed from Algorithm 1 with sampling measure $\nu = \nu_{\mu,V}$ has a unique solution, and with this same probability we have the relative error bound,

$$\|\tilde{v} - b\|^{2} \le (1 + \epsilon) \|v^{*} - b\|^{2}, \qquad (2.18)$$

which specializes to $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2^2$ when I is finite.

For the proof, see Appendix C. The result above specifies a minimal sampling requirement to guarantee a relative residual of the τ -sketched least squares problem with specified probability. Perhaps the most restrictive portion of the sampling requirement (2.17) is the dependence $K \gtrsim 1/\delta$, where δ is the failure probability. This dependence arises out of a specific use of the Markov inequality. An alternative strategy that circumvents the Markov inequality involves expectations (with respect to the randomness in τ) on a high-probability event. Precisely, for any $\epsilon > 0$ and $\delta \in (0, 1)$, consider sampling

$$K \ge 2N\left(\frac{1}{\epsilon} + 3\log\left(\frac{2N}{\delta}\right)\right) \implies \mathbb{E}\left[\|\widetilde{v} - b\|^2 \mid C\right] \le (1+\epsilon) \|v^* - b\|^2, \tag{2.19}$$

with
$$\Pr(C) \ge 1 - \delta.$$
 (2.20)

That is, with high probability one achieves on average ϵ -relative proximity to the optimal estimator residual. Note that the sample complexity now scales only like $\log(1/\delta)$, which is much more appealing than the $1/\delta$ scaling required to achieve (2.18). The proof of (2.19) is in Appendix D.

Finally, one may remove the conditional event C in the expectation, but in this case one must control the behavior of the estimator off the event C. One strategy to accomplish this involves estimates for a *truncated* estimator defined in terms of a constant T exceeding the maximum value of v over the support of μ ,

$$v_T(\boldsymbol{y}) = \begin{cases} v(\boldsymbol{y}), & \text{if } |v(\boldsymbol{y})| \le T\\ T \operatorname{sign}(v(\boldsymbol{y})), & \text{if } |v(\boldsymbol{y})| > T, \end{cases} \qquad T \ge \sup_{\boldsymbol{y} \in I} |b(\boldsymbol{y})|.$$
(2.21)

Then the following result holds for any $\epsilon > 0$ and $\delta \in (0, 1)$:

$$K \ge 2N\left(\frac{1}{\epsilon} + 3\log\left(\frac{2N}{\delta}\right)\right) \implies \mathbb{E}\|\tilde{v}_T - b\|^2 \le (1+\epsilon)\|v^* - b\|^2 + 4\delta T^2$$
(2.22)

See appendix E for the proof. Note again that in this case the truncation probability δ can be taken very small with relatively little impact since $K \gtrsim \log(1/\delta)$.

2.4 Algorithmic considerations

The algorithms in Section 2.3.2 require sampling from the induced measure $\nu = \nu_{\mu,V}$. Although an explicit formula for the density (relative to μ) is given by the leverage score formula (2.11), sampling from this measure using this formula is nontrivial since it at least requires identification of a V-orthonormal basis, which can be prohibitively expensive even when $I = \operatorname{supp} \mu$ is finite when the dimension D is large. This raises the question of how one can efficiently sampling from $\nu_{\mu,V}$.

One strategy is to use an approximate sampler, $\nu \approx \nu_{\mu,V}$, but with the condition that ν satisfies the following equivalence relation compared to $\nu_{\mu,V}$,

$$\inf_{\boldsymbol{y}\in I} \frac{\mathrm{d}\nu}{\mathrm{d}\nu_{\mu,V}}(\boldsymbol{y}) = \gamma \in (0,1].$$

For example, this results in an introduction of a factor of γ in steps (A.6a) and (A.11) in our proofs. Under this condition, one can generate τ by sampling from $\nu \approx \nu_{\mu,V}$, and pay only a price of an additional multiplicative $1/\gamma$ factor in the sampling condition (2.17) to achieve the same guarantees. In many cases ν , can be designed or generated in computable ways that yield tractable γ values [Dri+12; AM15; CLV17; Rud+18], and in the function approximation context asymptotics can be used to identify a tractable ν [NJZ17].

An alternative procedure is to devise strategies to *exactly* sample from $\nu_{\mu,V}$, in which case one must make additional assumptions to avoid exponential complexity in D. The remainder of this paper is devoted to one such strategy in the tensor product setting corresponding to when V is defined by a *monotone lower* index set. In the finite-I setting, this corresponds to A being a monotone lower column subset of a Kronecker product matrix, and our corresponding sampling algorithm scales linearly in D.

3 Tensor-product structure

Our main focus in this article is a specialization of the setup in the previous sections, where we assume that μ is a product measure over *D*-dimensional space, and *V* is a certain *subspace* of a tensor-product space of functions. We describe and discuss this specialized setup in this section.

We again consider *D*-dimensional approximation for a fixed $D \in \mathbb{N}$ as in Section 2, but with *I* and μ formed from a *D*-fold tensor product. Precisely, we assume,

$$I := \bigotimes_{d=1}^{D} I_d \subset \mathbb{R}^D, \qquad \qquad \mu \coloneqq \bigotimes_{d=1}^{D} \mu_d, \qquad (3.1)$$

where $I_d \subset \mathbb{R}$, $d \in [D]$ is a collection of sets in \mathbb{R} , each of which may be finite, countably infinite, or uncountably infinite. Associated to each set I_d , $d \in [D]$, we also assume the probability measure $\mu_d : \mathcal{B}(I_d) \to [0, 1]$, where $\mathcal{B}(I_d)$ is the σ -algebra of Borel sets on I_d . With these particular constructions for I and μ , the space $L^2_{\mu}(I)$ is as defined in Section 2.

The subspace V will also be constructed through a type of tensorization. For each $d \in [D]$, consider a given set of $N_d \in \mathbb{N}$ functions $a_j^{(d)}$, $j \in [N_d]$, which are elements of $L^2_{\mu_d}(I_d)$,

$$V_d \coloneqq \text{span}\left\{a_j^{(d)}, \ j \in [N_d]\right\}, \qquad a_j^{(d)} \in L^2_{\mu_d}(I_d) \ \forall j \in [N_d], \ d \in [D].$$
(3.2)

The tensorial space of functions is $\bigotimes_{d=1}^{D} V_d \subset L^2_{\mu}(I)$, but we can consider a subspace of this through a multi-index set. Our multi-indices will lie on the strictly positive integer lattice in D dimensions, \mathbb{N}^D . Our first multi-index will be N, formed by concatenating the dimensions of the spaces V_d ,

$$\boldsymbol{N} \coloneqq (N_1, \dots, N_D) \in \mathbb{N}^D.$$
(3.3)

In what follows we use the following notation to denote the orthant in \mathbb{N}^D that is bounded by N along with its size (N),

$$[\boldsymbol{N}] \coloneqq \bigotimes_{d=1}^{D} [N_d], \qquad (\boldsymbol{N}) \coloneqq |[\boldsymbol{N}]| = \prod_{d=1}^{D} N_d.$$

We will now assume that some size-N subset \mathcal{J} of the bounded orthant [N] is given,

$$\mathcal{J} \subseteq [\mathbf{N}] \subset \mathbb{N}^D,$$
 $N = |\mathcal{J}|.$ (3.4a)

I.e., \mathcal{J} contains multi-indices whose maximum entry in location d is at most N_d . Finally, we will require an ordering of the multi-indices,

$$\mathcal{J} = \{\alpha_n\}_{n \in [N]} \,. \tag{3.4b}$$

The subspace of functions V we consider from here onward are spanned by products of functions from the V_d spaces,

$$V \coloneqq \operatorname{span}\left\{a_{\alpha} \mid \alpha \in \mathcal{J}\right\} = \operatorname{span}\left\{a_{n} \mid n \in [N]\right\}, \qquad a_{\alpha}(\boldsymbol{y}) \coloneqq \prod_{d=1}^{D} a_{\alpha^{(d)}}^{(d)}(y_{d}), \qquad (3.5)$$

where $\alpha^{(d)}$ is the *d*th index in the multi-index α . Note that we have introduced a linear ordering of the functions $a_j \in L^2_{\mu}(I)$ for $j \in N$ corresponding to the linear order (3.4b) of multi-indices in \mathcal{J} .

3.1 The multi-index set \mathcal{J}

In this section we introduce and discuss some common choices for \mathcal{J} . To describe our choices, we will use *weighted* ℓ^p functions on multi-indices α . Let $\boldsymbol{w} = (w_1, \ldots, w_D) \in (0, 1]^D$ denote a weight vector

satisfying $\max_d w_d = 1$. Then the weighted ℓ^p function is given by,

$$\|\alpha\|_{\boldsymbol{w},p} \coloneqq \left[\sum_{d=1}^{D} \left(\frac{\alpha^{(d)}}{w_d}\right)^p\right]^{1/p}, \qquad \qquad 0$$

$$\|\alpha\|_{\boldsymbol{w},\infty} \coloneqq \max_{d \in [D]} \frac{\alpha^{(d)}}{w_d},\tag{3.6b}$$

$$\|\alpha\|_{\boldsymbol{w},0} \coloneqq \sum_{d \in [D]} \mathbb{1}_{\alpha^{(d)} \neq 0},\tag{3.6c}$$

where $\mathbb{1}_S$ is the indicator function for the condition S. In our context, $w_j < w_k$ indicates that dimension k is more "important" in the sense that values of α satisfying $\|\alpha\|_{\boldsymbol{w},p} \leq G$ for a fixed G permits larger values of $\alpha^{(k)}$ than $\alpha^{(j)}$. Let $\mathbf{1} \coloneqq (1, 1, \ldots, 1) \in \mathbb{N}^D$ be a multi-index with all entries 1. When $\boldsymbol{w} = \mathbf{1}$, the functions above reduce to the standard (unweighted) ℓ^p functions, and we utilize abbreviated notation for this case,

$$\|\alpha\|_p \coloneqq \|\alpha\|_{1,p}, \qquad \qquad 0 \le p \le \infty.$$

Note that when p = 0, the value of \boldsymbol{w} plays no role. Very common index sets \mathcal{J} correspond to the **1**-centered ball of "order"/radius $G \geq 0$ in the \boldsymbol{w} -weighted ℓ^p metric:

$$\mathcal{J}_{\boldsymbol{w},p}(G) \coloneqq \left\{ \alpha \in \mathbb{N}^D \mid \|\alpha - \mathbf{1}\|_{\boldsymbol{w},p} \le G \right\}, \qquad \mathcal{J}_p(G) \coloneqq \mathcal{J}_{\mathbf{1},p}(G).$$
(3.7)

One typically chooses $G \ge 0$ as an integer. We plot examples of these sets in Figure 1. The definitions



Figure 1: Examples of index sets $\mathcal{J}_{\boldsymbol{w},p}$ in D = 2 dimensions defined by (3.7). Top row: unweighted sets $(\boldsymbol{w} = 1)$. Bottom row: weighted sets $(\boldsymbol{w} = (\frac{2}{3}, 1))$.

above are somewhat more transparent if one chooses the functions $a_j^{(d)}$ in (3.2) to be polynomials, e.g., the monomials $a_j^{(d)}(\boldsymbol{y}) = y_d^{j-1}$. Then $\mathcal{J}_1(G)$ (i.e., p = 1) corresponds to the space V of degree-G polynomials in D dimensions. The choice p = 2 is associated to approximation with the "Euclidean degree" G [Tre17]. The choice $p = \infty$ corresponds to a tensorial space V of polynomials with maximum degree G in any dimension. When p = 0 and G < D, then $\mathcal{J}_0(G)$ corresponds to a set of multi-indices α where at most G entres of α are larger than 1, and so in the polynomial context this means that functions in V are non-constant in at most G dimensions.

For $1 \le p \le \infty$, then $\|\cdot\|_p$ is a norm, so that $\mathcal{J}_p(G)$ in this case is a convex set.² Smaller values of p, say $p \ll 1$, diminish the presence of basis functions corresponding to multi-indices α where all

²More precisely, the polytope whose vertices are given by $\mathcal{J}_p(G)$ is equal to $\operatorname{conv}(\mathcal{J}_p(G))$.

components α_d are larger than 1. This is commonly interpreted as diminishing the basis functions with D-dimensional interactions. A more aggressive pruning of such high-dimensional interactions is furnished by the hyperbolic cross space,

$$\mathcal{J}_{\boldsymbol{w},_{\mathrm{HC}}}(G) \coloneqq \left\{ \alpha \in \mathbb{N}^D \mid \|\log \alpha\|_{\boldsymbol{w},1} \le \log(G+1) \right\}$$

where $\alpha \mapsto \log \alpha$ is componentwise. Hyperbolic cross spaces have ties to function spaces of certain smoothness in high dimensions [DG16; DTU18], and are natural spaces for sparse/compressive approximations [ABW22].

In all the above cases, taking $\boldsymbol{w} \neq \mathbf{1}$ yields anisotropic multi-index sets: $w_j < w_k$ implies that $\mathcal{J}_{\boldsymbol{w},p}(G)$ allows larger multi-indices in dimension k compared to dimension j. Utilizing such spaces is typical in high-dimensional approximation of parametric PDEs where often one has a priori information about the relative importance of each dimension.

There is one common property that is shared by all the index sets we have discussed.

Definition 3.1. We say that the set of multi-indices \mathcal{J} is a monotone lower set if the lexicographic multi-index version of \mathcal{J} is a monotone lower set, i.e., if \mathcal{J} satisfies,

$$\alpha \in \mathcal{J} \implies \beta \in \mathcal{J} \ \forall \ \mathbf{1} \le \beta \le \alpha, \tag{3.8}$$

where \leq is interpreted componentwise.

All the multi-index sets we have introduced before are monotone lower.

Proposition 3.2. For any $p \ge 0$, $G \ge 0$, and weight vector \boldsymbol{w} , then the multi-index sets $\mathcal{J}_{\boldsymbol{w},p}(G)$ and $\mathcal{J}_{\boldsymbol{w},_{\mathrm{HC}}}(G)$ are all monotone lower, and we have

$$\mathcal{J}_{\boldsymbol{w},p}(G) \subseteq [\boldsymbol{N}], \qquad \qquad \mathcal{J}_{\boldsymbol{w},_{\mathrm{HC}}}(G) \subseteq [\boldsymbol{N}], \qquad (3.9)$$

where the components of N are defined by identifying the smallest bounding box for \mathcal{J} :

With
$$\mathcal{J} = \mathcal{J}_{\boldsymbol{w},p}(G)$$
 or $\mathcal{J} = \mathcal{J}_{\boldsymbol{w},\text{HC}}(G)$, then $N_d = \lfloor Gw_d \rfloor = \max_{\alpha \in \mathcal{J}} \alpha_d$. (3.10)

We omit the proof; the main idea is to show that for any $\alpha \in \mathcal{J}$, then $\alpha - e_d$ for any $d \in [D]$ also lies in \mathcal{J} , where e_d is the canonical unit vector in direction d. Monotone lower sets are precisely the types of column subsets of a Kronecker product least-squares problem that we will consider. In particular, we will show that even though least squares problems on a Kronecker product column subset do *not* have Kronecker structure, if the column subset is monotone lower, then we can still compute accurate, approximate solutions with formal $\mathcal{O}(D)$ complexity.³

Finally, we remark that more generally we only require that our multi-index sets \mathcal{J} are monotone lower after a dimensionwise permutation. For example, with D = 2, the index set,

$$\mathcal{J} = \left\{ \begin{array}{ccc} (1,1), & (1,2), & (1,3), & (1,4) \\ (2,1), & & (2,4) \\ (3,1), & (3,2), & (3,3), & (3,4) \end{array} \right\},\,$$

is clearly not monotone lower, but if we define a 2-dimensional permutation π via dimensionwise permutations π_1 and π_2 ,

$$\begin{aligned} \pi((\alpha_1, \alpha_2)) &\coloneqq (\pi_1(\alpha_1), \pi_2(\alpha_2)), \\ \pi_1 &: [3] \to [3], \\ \pi_2 &: [4] \to [4], \end{aligned} \qquad \begin{aligned} \pi_1(\{1, 2, 3\}) &= \{1, 3, 2\}, \\ \pi_2(\{1, 2, 3, 4\}) &= \{1, 4, 3, 2\}, \end{aligned}$$

then $\pi(\mathcal{J})$ is a monotone lower set, and all our theory and algorithms apply by simply exercising this permutation map. We visualize two more sets in Figure 2 that are not monotone lower, but become monotone lower under appropriate dimensionwise permutations.

³Of course, the complexity will depend on N, which could grow exponentially in D.



Figure 2: Two index sets \mathcal{J} and $\widetilde{\mathcal{J}}$ (first and third plots) that are not monotone lower, but appropriate dimensionwise permutation maps π and $\widetilde{\pi}$ make them monotone lower (second and fourth plots). Because of this, our algorithms apply to least squares approximations on sets \mathcal{J} and $\widetilde{\mathcal{J}}$ by initially applying the permutations, applying our algorithms, and finally by exercising the inverse permutations.

3.2 Example: Polynomial approximation on tensorial domains

This section pairs with Example 2.2. For each $d \in [D]$, let I_d be an interval (possibly infinite) on the real line \mathbb{R} , and let μ_d be some probability measure on I_d . For example, if I_d is bounded, μ_d can be chosen as the uniform measure on I_d . We choose monomials as our univariate basis functions,

$$a_j^{(d)}(\boldsymbol{y}) = y_d^{j-1}$$

in which case the least squares problem (2.2) corresponds to a μ -weighted best- L^2 approximation with polynomials from the multi-index set \mathcal{J} :

$$\boldsymbol{x}^* = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^{|\mathcal{J}|}} \int_I \left(f(\boldsymbol{y}) - \sum_{n=1}^N x_n \boldsymbol{y}^{\alpha_n} \right)^2 \mathrm{d}\mu(\boldsymbol{y}),$$

where α_n is an enumeration of the multi-indices as introduced in (3.4b), and $\boldsymbol{y}^{\alpha_n}$ is the standard monomial multi-index notation,

$$\boldsymbol{y}^{lpha_n} = \prod_{d=1}^D y_d^{lpha_n^{(d)}}.$$

3.3 Example: Matrix-vector least squares problems

This section pairs with Example 2.1. Our formalism in the previous sections specializes to familiar vector and matrix realizations when each domain is finite. Choose each μ_d in (3.1) as a discrete measure over $M_d \ge 1$ points, with support

$$I_d = \operatorname{supp} \mu_d = \left\{ y_1^{(d)}, \dots y_{M_d}^{(d)} \right\}, \qquad \qquad \mu_d = \sum_{m=1}^{M_d} w_m^{(d)} \delta_{y_m^{(d)}}, \qquad (3.11)$$

for some non-negative weights $w_m^{(d)}$ where δ_y is the Dirac mass centered at y. Then the algebraic problem (2.2) is equivalent to solving the following standard least squares problem,

$$\boldsymbol{x}^* = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^N} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2, \qquad (3.12)$$

<u>م</u> ۲

where

$$(\boldsymbol{A})_{m,n} = \sqrt{w_m} a_n(\boldsymbol{y}_m), \qquad (\boldsymbol{b})_m = \sqrt{w_m} b(\boldsymbol{y}_m), \qquad (m,n) \in [(\boldsymbol{M})] \times [N].$$
(3.13)

where $w_i, i \in [(M)]$ are a lexicographic reordering of the weights $\prod_{d=1}^{D} w_{\beta^{(d)}}^{(d)}$ over all multi-indices $\beta \in [M] = [(M_1, \ldots, M_D)]$. Thus, our formalism in a very simple setting is standard linear algebraic least-squares.

In particular, this setup demonstrates the Kronecker product structure of our problem. For each $d \in [D]$, define the following $M_d \times N_d$ matrix:

$$\left(\mathbf{A}^{(d)}\right)_{m,n} = \sqrt{w_m^{(d)}} a_n^{(d)} \left(y_m^{(d)}\right), \qquad (m,n) \in [M_d] \times [N_d] \qquad (3.14)$$

The Kronecker product of these "univariate" matrices is an $(M) \times (N)$ matrix,

$$\boldsymbol{A}_{\rm kr} = \boldsymbol{A}^{(1)} \otimes \cdots \otimes \boldsymbol{A}^{(D)}. \tag{3.15}$$

The matrix \boldsymbol{A} in (3.13) is a column subset of \boldsymbol{A}_{kr} . In particular, write $(\boldsymbol{A}_{kr})_{:,\alpha}$ for some $\alpha \in [\boldsymbol{N}]$ to mean the column of \boldsymbol{A}_{kr} whose linear index corresponds to the linear index of α in a lexicographic ordering of $[\boldsymbol{N}]$. Then,

$$\boldsymbol{A} = \left(\boldsymbol{A}_{\mathrm{kr}}\right)_{:,,\mathcal{T}}.\tag{3.16}$$

Hence, the least squares problem (3.12) is a *column subset* of the corresponding full Kronecker product problem,

$$rgmin_{oldsymbol{x}\in\mathbb{R}^{(oldsymbol{N})}} \|oldsymbol{A}_{ ext{kr}}oldsymbol{x}-oldsymbol{b}\|^2$$
 .

Our main goal is to describe an efficient procedure for sampling from the induced distribution $\nu_{\mu,V}$ so that we can exercise Theorem 2.8 to compute solutions to (3.11) efficiently. Equivalently, we seek to *efficiently* sample the rows of \boldsymbol{A} according to its leverage scores, exploiting the fact that \boldsymbol{A} is a column subset of \boldsymbol{A}_{kr} . It is fairly straightforward to establish that the leverage scores of \boldsymbol{A}_{kr} are simply products of the leverage scores of $\boldsymbol{A}_{a,r}$ are simply products of the leverage scores of $\boldsymbol{A}_{a,r}$ and hence can be efficiently sampled (see Section 4.3.1). However, our proposed algorithm in Section 4.3.3 can also efficiently sample rows according to the leverage scores of \boldsymbol{A} when \mathcal{J} is monotone lower. This monotone lower subset property is in particular what distinguishes our setup (and approach) from similar least squares problems involving Kronecker products [FF94; FFH97; Ses17; MMV19; FH20].

4 Sampling algorithms for $\nu_{\mu,V}$

We have seen that in order efficiently solve least squares problems using the procedures outlined in Section 2.3, we require the ability to draw samples from the induced distribution ν defined in Section 2.2. This section discusses two algorithms for (exactly) sampling from this distribution, which amounts to computing and sampling according to leverage scores of \mathbf{A} when μ is finitely supported.

The goal of this section is to discuss algorithms that sample from the induced/leverage score distribution ν in (2.9). Our discussion will leverage a sequence of *univariate* orthonormal basis functions, along with some univariate induced distributions. In particular, we will identify an algorithm that can quickly and exactly sample from the *D*-dimensional measure ν leveraging only 1-dimensional sampling methods. In all that follows, we assume that \mathcal{J} is a finite set of multi-indices.

4.1 Direct sampling of ν

The most direct way to sample from the measure ν is to explicitly compute its density with respect to μ as given in (2.11), and then to use some method to sample from this *D*-dimensional measure. More explicitly, we have,

$$d\nu(\boldsymbol{y}) = \frac{1}{N} \sum_{j=1}^{N} |u_j(\boldsymbol{y})|^2 d\mu(\boldsymbol{y}).$$
(4.1)

Therefore, a direct way to sample from ν is:

- 1. Given the original basis functions a_n , compute the orthonormal basis $\{u_n\}_{j=n}^N$.
- 2. Sample from the D-dimensional density in (4.1).

However, the above approach is computationally expensive or even infeasible in general. First, computing the u_n requires integration over *D*-dimensional space according to the measure μ . Second, sampling from general multidimensional measures, such as (4.1), is itself an expensive undertaking, regardless of if one uses exact samplers (such as rejection sampling) or approximate samplers (such as Markov Chain Monte Carlo).

In the finitely-supported measure case, we can more precisely quantify the required cost. The steps in this case are to

- 1. Compute some element $U \in col(A)$. Using, for example, the QR decomposition, requires $\mathcal{O}(MN^2)$ complexity and $\mathcal{O}(MN)$ storage.
- 2. Compute and sample according to the (normalized) leverage scores in (2.13). Computing the scores requires $\mathcal{O}(MN)$ effort and $\mathcal{O}(M)$ storage. The actual sampling from the size-M discrete distribution requires a relatively negligible $\mathcal{O}(M)$ initialization cost with a subsequent $\mathcal{O}(1)$ cost per sample [Vos91].

We see that there is a rather explicit dependence on both M and N for both steps. Recall that when $D \gg 1$, both M and N can be so large that even forming and storing the full matrix A can be computationally infeasible. Thus, even in the finite-supported case, this direct approach is not appealing.

The purpose of the next sections is to identify algorithms for sampling that entirely circumvent dependence on M and N. The essential ingredient to accomplish this is the ability to sample from certain one-dimensional measures.

4.2 Sampling from univariate measures

Before discussing how one can more efficiently sample from the *D*-dimensional measure ν , we introduce some needed concepts for one-dimensional sampling. Recall from (3.2) that to each dimension $d \in [D]$ we have a collection of basis functions $(a_j^{(d)}), j \in [N_d]$. We require identification of a(ny) $L^2_{\mu_d}(I_d)$ -set of orthonormal basis functions $q_k^{(d)}$,

$$a_{j}^{(d)} = \sum_{k \in [N_d]} R_{k,j}^{(d)} q_k^{(d)}, \qquad \qquad \int_{I_d} q_k^{(d)}(y^{(d)}) q_j^{(d)}(y^{(d)}) d\mu_d(y^{(d)}) = \delta_{j,k}, \tag{4.2}$$

where $\mathbf{R}^{(d)} \in \mathbb{R}^{N_d \times N_d}$ is an invertible matrix. A common strategy to accomplish this is via a Gram-Schmidt-like triangular factorization, i.e.,

$$k > j \implies (R^{(d)})_{k,j} = 0 \implies \mathbf{R}^{(d)}$$
 is upper triangular.

Associated to the basis functions $q_k^{(d)}$ we construct univariate measures,

$$d\nu_k^{(d)}(y^{(d)}) = \left| q_k^{(d)}(y^{(d)}) \right|^2 d\mu_d(y^{(d)}), \qquad k \in [N_d], d \in [D], \qquad (4.3)$$

which are all probability measures over I_d since the $q_k^{(d)}$ functions have unit $L^2_{\mu_d}$ norms. We require the ability to sample from $\nu_k^{(d)}$ for each (k, d).

Assumption 4.1. For any k and d, drawing a sample from $\nu_k^{(d)}$ defined in (4.3) is computationally feasible and cheap, e.g., $\mathcal{O}(1)$ complexity.

Since these are all univariate distributions, one concrete way to sample is via inverse transform sampling, i.e., to compute,

$$Y_k^{(d)} = F_{\nu_k^{(d)}}^{-1}(U), \ U \sim \mathcal{U}[0,1] \implies Y_k^{(d)} \sim \nu_k^{(d)}, \tag{4.4}$$

where $F_{\nu_k^{(d)}}(\cdot)$ is the cumulative distibution function of $\nu_k^{(d)}$ and $F_{\nu_k^{(d)}}^{-1}$ is its function inverse. For example, for polynomial subspaces there are computational strategies to efficiently compute this for fairly general μ_d [Nar18].

Again, we can write all the above in somewhat more transparent form by specializing to our finite setup of Section 3.3. In that setup, assume for simplicity that each matrix $A^{(d)}$ has full column rank, so that each $A^{(d)}$ has a full-rank QR decomposition, i.e.,

$$\boldsymbol{A}^{(d)} = \boldsymbol{Q}^{(d)} \boldsymbol{R}^{(d)}, \qquad \qquad \boldsymbol{Q}^{(d)} \in \mathbb{R}^{M_d \times N_d}, \quad \boldsymbol{R}^{(d)} \in \mathbb{R}^{N_d \times N_d}, \tag{4.5}$$

with each $\mathbf{R}^{(d)}$ upper triangular and invertible. We assume availability of $\mathbf{Q}^{(d)}$, which is a one-time ("offline") computation for each $d \in [D]$ requiring a cumulative $O(DM_{\max}N_{\max}^2)$ complexity with $O(DM_{\max}N_{\max})$ storage, where $M_{\max} = \max_{d \in [D]} M_d$ and $N_{\max} = \max_{d \in [D]} N_d$. In particular, we will assume the ability to access the columns of $\mathbf{Q}^{(d)}$ for each d:

$$oldsymbol{Q}^{(d)} = \left[oldsymbol{q}_1^{(d)}, \ \ldots, \ oldsymbol{q}_{N_d}^{(d)}
ight].$$

In this finite setup, we have $I_d = \left\{ y_1^{(d)}, \ldots, y_{M_d}^{(d)} \right\}$, the mass weights for μ_d are $w_m^{(d)}$, and so $\nu_k^{(d)}$ is explicitly given as the discrete measure,

$$\nu_k^{(d)} = \sum_{m \in [M_d]} w_m^{(d)} \left| \left(q_k^{(d)} \right)_m \right|^2 \delta_{y_m^{(d)}}$$

suggesting the following definition of the (normalized) (k, d) one-dimensional leverage scores:

$$\ell_{k,m}^{(d)} = w_m^{(d)} \left| \left(q_k^{(d)} \right)_m \right|^2, \qquad m \in [M_d].$$
(4.6)

The quantities $\ell_{k,m}^{(d)}$ are leverage scores associated to the vector $\boldsymbol{q}_k^{(d)}$. Sampling from $\nu_k^{(d)}$ is equivalent to sampling from the discrete set $\{y_1^{(d)}, \ldots, y_{M_d}^{(d)}\}$ according to the (k, d) leverage scores defined above. An algorithm and cost summarization is given in Algorithm 2.

Algorithm 2: Sampling from $\nu_k^{(d)}$ in (4.3) for all $k \in [N_d]$ when μ is finitely supported.

1 Initialization;

Input: $d \in [D], N_d \in \mathbb{N}$

2 Compute $\mathbf{A}^{(d)}$ in (3.14). $[\mathcal{O}(M_d N_d) \cos t + \text{storage}];$

- **3** Compute $\boldsymbol{Q}^{(d)}$ in (4.5). $[\mathcal{O}(M_d N_d^2) \operatorname{cost}, \mathcal{O}(M_d N_d) \operatorname{storage}];$
- 4 Compute the normalized (k, d) leverage scores $\ell_{k,m}^{(d)}$ in (4.6) for all $k \in [N_d]$. $[\mathcal{O}(M_d N_d)$ cost and storage];
- 5 Initialization for sampling from $\{y_1^{(d)}, \ldots, y_{M_d}^{(d)}\}$ according to the (k, d) leverage scores. $[\mathcal{O}(M_d) \text{ cost per sample}];$
- 6 Sampling;
- Input: $k \in [N_d]$
- 7 Sample Y from $\{y_1^{(d)}, \ldots, y_{M_d}^{(d)}\}$ according to the (k, d) leverage scores. $[\mathcal{O}(1) \text{ cost per sample}];$ Output: Y

For sampling from such discrete measures, computationally effective, $\mathcal{O}(1)$ -per sample approaches exist with a one-time $\mathcal{O}(M_d)$ initialization cost [Vos91]. Thus, in the finitely-supported measure case, Assumption 4.1 is always satisfied with reasonable M_d - and N_d -dependent initialization costs.

4.3 Sampling algorithms

Having completed the requisite setup, we can now describe several sampling algorithms under certain structural assumptions on A. Section 4.3.1 discusses sampling when A is an actual Kronecker product matrix. Section 4.3.2 discusses sampling when the columns of A are orthogonal. Finally, Section 4.3.3 discusses a more interesting setting when the index set \mathcal{J} defining the columns of A is monotone lower.

4.3.1 Tensor product spaces

In this section we make the assumption that $\mathcal{J} = \mathcal{J}_{\boldsymbol{w},\infty}(K) = \times_{d \in [D]}[N_d]$. In practice one is not frequently interested in this case due to the large size of \mathcal{J} when D is large. Nevertheless, the ideas from this situation are ingredients for our strategy in the general monotone lower case.

First we observe that the dimension- $d(\mu_d, V_d)$ -induced distribution, $\nu^{(d)}$, is a uniform mixture of the measures $\nu_k^{(d)}$,

$$\nu^{(d)} = \frac{1}{N_d} \sum_{k \in [N_d]} \nu_k^{(d)},\tag{4.7}$$

which is a direct consequence of (4.3) and the computation (2.11). Therefore, $\nu^{(d)}$ is a uniform mixture of measures $\nu_k^{(d)}$, and under Assumption 4.1 is therefore easily sampled from.

Unsurprisingly, the induced distribution under the tensor-product space assumption of this section is a product measure.

Proposition 4.2. If $\mathcal{J} = \times_{d \in [D]}[N_d]$, then the (μ, V) -induced distribution ν in (2.9) is given by,

$$\nu = \bigotimes_{d \in [D]} \nu^{(d)},$$

where $\nu^{(d)}$ for each d is the (μ_d, V_d) univariate induced distribution in (4.7).

Proof. In this tensor-product setup, then with the univariate orthonormal functions $q_k^{(d)}$ introduced in the previous section, it is straightforward to show that the collection of functions,

$$q_{\alpha}(\boldsymbol{y}) \coloneqq \prod_{d \in [D]} q_{\alpha^{(d)}}^{(d)}(\boldsymbol{y}^{(d)}), \qquad \alpha \in \mathcal{J} = [\boldsymbol{N}],$$
(4.8)

is an orthonormal basis for \mathcal{J} . Thus, according to (2.11),

$$d\nu(\boldsymbol{y}) = \frac{1}{\prod_{d\in[D]} N_d} \sum_{\alpha\in[\boldsymbol{N}]} |q_\alpha(\boldsymbol{y})|^2 d\mu(\boldsymbol{y})$$

$$\stackrel{(4.8)}{=} \frac{1}{\prod_{d\in[D]} N_d} \sum_{\alpha\in[\boldsymbol{N}]} \prod_{d\in[D]} \left| q_{\alpha^{(d)}}^{(d)}(y^{(d)}) \right|^2 d\mu(\boldsymbol{y})$$

$$= \frac{1}{\prod_{d\in[D]} N_d} \sum_{\alpha^{(d)}=1}^{N_1} \cdots \sum_{\alpha^{(D)}=1}^{N_D} \prod_{d\in[D]} \left| q_{\alpha^{(d)}}^{(d)}(y^{(d)}) \right|^2 d\mu_1(y^{(1)}) \cdots d\mu_D(y^{(D)})$$

$$= \frac{1}{\prod_{d\in[D]} N_d} \prod_{d\in[D]} \left(\sum_{\alpha^{(d)}=1}^{N_d} \left| q_{\alpha^{(d)}}^{(d)}(y^{(d)}) \right|^2 d\mu_d(y^{(d)}) \right)$$

$$\stackrel{(4.7)}{=} \prod_{d\in[D]} d\nu^{(d)}.$$

The result above shows that at least in some specialized cases, the somewhat opaque *D*-dimensional measure ν is actually a tensor-product measure, and hence easily sampled from since one need only sample $\nu^{(d)}$.

When μ is finitely supported, the result above specializes to the statement that leverage scores for a Kronecker product matrix are products of the leverage scores from the individual matrices forming the product:

$$\mathcal{J} = \times_{d=1}^{D} [N_d] \text{ and } \mathbf{U}^{(d)} \in \operatorname{col}(\mathbf{A}^{(d)}) \Longrightarrow \mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(D)} \in \operatorname{col}(\mathbf{A})$$
$$\implies \ell_{\alpha}(\mathbf{A}) = \ell_{\alpha} \left(\bigotimes_{d=1}^{D} \mathbf{A}^{(d)}\right) = \prod_{d=1}^{D} \ell_{\alpha_d}(\mathbf{A}^{(d)}), \quad (4.9)$$

where $\ell_{\alpha}(\mathbf{A})$ is the leverage score associated to row of \mathbf{A} formed from the row α_d of $\mathbf{A}^{(d)}$ for $d \in [D]$. In this finitely supported μ case, Algorithm 3 describes how to sample from ν when \mathcal{J} is a hypercube on the lattice, and summarizes the corresponding complexity.

Algorithm 3: Algorithm from Section 4.3.1 when μ is finitely supported and \mathcal{J} is a product

set	t.		
1]	1 Initialization;		
]	Input: One-dimensional sizes (N_1, \ldots, N_D)		
2	For each $d \in [D]$, perform Initialization step in Algorithm 2. $[\mathcal{O}(\sum_{d=1}^{D} M_d N_d^2) \text{ cost and}$		
	$\mathcal{O}(\sum_{d=1}^{D} M_d)$ storage];		
3 Sampling;			
4	for $d = 1, \ldots, D$ do		
5	Select k uniformly at random from $[N_d]$. $[\mathcal{O}(1) \cos t + \text{storage}];$		
6	Generate sample $Y^{(d)}$ from Sampling step in Algorithm 2 with input (k, d) . $[\mathcal{O}(1) \cos t +$		
	storage];		
7	end		
8	Return $\boldsymbol{Y} = (Y^{(1)}, \dots, Y^{(D)});$		

Note that while Algorithm 3 samples from $\nu^{(d)}$ by using the mixture property (4.7), one can equivalently replace steps 5 and 6 in Algorithm 3 by a single step that samples from the (μ_d, V_d) one-dimensional leverage scores associated with $\mathbf{A}^{(d)}$. This simpler strategy utilizes the Kronecker product leverage score property (4.9).

4.3.2 Orthogonal basis functions

Another relatively simple case when exactly sampling from ν is fairly straightforward occurs when the one-dimensional functions $a_j^{(d)}$ for $j \in [N_d]$ are $L^2_{\mu_d}(I_d)$ -orthogonal. In this case, the one-dimensional orthonormal functions $q_j^{(d)}$ in (4.2) are simply scaled versions of $a_j^{(d)}$. The following result is then straightforward to establish.

Proposition 4.3. Assume that for every d we have the property,

$$\left\langle a_{j}^{(d)}, a_{k}^{(d)} \right\rangle_{\mu_{d}} = 0, \qquad j, k \in [N_{d}], \ j \neq k.$$
 (4.10)

Then,

$$q_j^{(d)} = \frac{a_j^{(d)}}{\|a_j^{(d)}\|_{\mu_d}},\tag{4.11}$$

and given any $\mathcal{J} = \{\alpha_1, \ldots, \alpha_N\}$, an orthonormal basis $u_n, n \in [N]$ for V is given by,

$$u_j(\boldsymbol{y}) = \prod_{d=1}^{D} q_{\alpha_j^{(d)}}^{(d)}(y^{(d)}), \qquad \qquad \mathcal{J} = \{\alpha_1, \dots, \alpha_N\}.$$
(4.12)

The normalization property (4.11) is immediate from (4.10), and (4.12) follows by the product property (3.5) of the a_{α} functions that span V.

Thus, if Assumption 4.1 holds, then combining (4.12) with (2.11) shows that we can sample efficiently from ν by (i) uniformly at random choosing some $\alpha_n \in \mathcal{J}$, and (ii) sampling $Y^{(d)}$ according to $\nu_{\alpha_n^{(d)}}^{(d)}$ for each d, forming the multivariate sample $\mathbf{Y} = (Y^{(1)}, \ldots, Y^{(d)})$ that is distributed according to ν .

When μ is finitely supported, then the assumption (4.10) is equivalent to $\mathbf{A}^{(d)}$ having orthogonal columns, and therefore the columns of \mathbf{A} are orthogonal, hence the leverage scores are easily computed directly from \mathbf{A} by appropriately normalizing the columns from matrices $\mathbf{A}^{(d)}$, in particular by computing,

$$\boldsymbol{q}_{k}^{(d)} = \frac{\boldsymbol{a}_{k}^{(d)}}{\|\boldsymbol{a}_{k}^{(d)}\|}, \qquad \qquad k \in [N_{d}], \ d \in [D].$$
(4.13)

An algorithm that achieves this type of sampling is described in Algorithm 4.

Algorithm 4: Algorithm from Section 4.3.2 under the orthogonal columns assumption (4.10) when μ is finitely supported.

1 Initialization;

Input: Index set \mathcal{J}

- 2 Compute N_d for $d \in [D]$ using (3.10). $[\mathcal{O}(DN) \operatorname{cost}];$
- **s** For each $d \in [D]$ compute individual columns of $\mathbf{Q}^{(d)}$ in (4.5) via (4.13). $[\mathcal{O}(\sum_{d=1}^{D} M_d N_d) \text{ cost} + \text{storage}];$
- 4 Compute the one-dimensional (k, d) leverage scores $\ell_{k,m}^{(d)}$ in (4.6) for $m \in [M_d]$ $[\mathcal{O}(\sum_{d=1}^D M_d N_d) \text{ cost} + \text{ storage}];$
- 5 Sampling;
- 6 Choose α uniformly at random from \mathcal{J} . [$\mathcal{O}(1)$ cost];
- 7 For $d \in [D]$, sample $Y^{(d)}$ according to $\nu_{\alpha^{(d)}}^{(d)}$, accomplished using the Sampling step in Algorithm 2. $[\mathcal{O}(D) \text{ cost}]$; **Output:** $\mathbf{Y} = (Y^{(1)}, \dots, Y^{(D)})$

4.3.3 Monotone lower \mathcal{J}

The most general case we consider in this paper is when \mathcal{J} is a monotone lower set of indices as defined in Definition 3.1. The sets $\mathcal{J}_{w,p}(K)$ introduced in Section 3.1 are examples of such subsets, but are not comprehensive (e.g., the \mathcal{J}_{HC} sets and the set $\pi(\mathcal{J})$ in the second column of Figure 2 are other examples).

The main result we require to establish fast sampling is the following identification of an orthonormal basis for V.

Proposition 4.4. Assume $\mathcal{J} = \{\alpha_1, \ldots, \alpha_N\}$ is monotone lower, and define,

$$q_{j}(\boldsymbol{y}) \coloneqq \prod_{d \in [D]} q_{\alpha_{j}^{(d)}}^{(d)}(y^{(d)}), \qquad \qquad \alpha_{j} = (\alpha_{j}^{(1)}, \dots, \alpha_{j}^{(D)}).$$
(4.14)

Then $\{q_j\}_{j\in[N]}$ is an $L^2_{\mu}(I)$ -orthonormal basis for V.

Proof. The $N = \dim V$ functions q_n , $n \in [N]$ are $L^2_{\mu}(I)$ -orthonormal by construction, and therefore we need only show that the functions a_j that span V lie in span $\{q_n\}_{n \in [N]}$. We have:

$$a_{j}(\boldsymbol{y}) \stackrel{(3.5)}{=} \prod_{d \in [D]} a_{\alpha_{j}^{(d)}}(y^{(d)}) \stackrel{(4.2)}{=} \prod_{d \in [D]} \sum_{k \in [N_{d}]} R_{k,\alpha_{j}^{(d)}}^{(d)} q_{k}^{(d)}(y^{(d)}) = \prod_{d \in [D]} \sum_{\beta^{(d)} \in [N_{d}]} R_{\beta^{(d)},\alpha_{j}^{(d)}}^{(d)} q_{\beta^{(d)}}^{(d)}(y^{(d)}) = \sum_{\beta \leq \alpha_{j}} \left(\prod_{d \in [D]} R_{\beta^{(d)},\alpha_{j}^{(d)}}^{(d)} \right) \left(\prod_{d \in [D]} q_{\beta^{(d)}}^{(d)}(y^{(d)}) \right) \stackrel{(3.8)}{=} \sum_{\beta \in \mathcal{J}} c_{\alpha_{j},\beta} \left(\prod_{d \in [D]} q_{\beta^{(d)}}^{(d)}(y^{(d)}) \right) = \sum_{n=1}^{N} c_{\alpha_{j},\alpha_{n}} q_{n}(\boldsymbol{y}),$$

proving the result.

19

Therefore, if \mathcal{J} is a monotone lower set, then there is an efficiently computable orthonormal basis (4.14) for V that is formed from products of univariate functions. The mixture property (2.11) of ν with $u_n = q_n$ then yields the following formula for ν :

$$\mathcal{J} = \{\alpha_1, \dots, \alpha_N\} \text{ monotone lower } \implies \nu = \frac{1}{N} \sum_{n \in [N]} \prod_{d \in [D]} \nu_{\alpha_n^{(d)}}^{(d)}.$$

I.e., ν is a uniform mixture of product measures, revealing the following algorithm for quickly sampling from ν when \mathcal{J} is monotone lower: (1) Uniformly at random select $\alpha \in \mathcal{J}$, (2) for each $d \in [D]$, sample $Y^{(d)} \sim \nu_{\alpha^{(d)}}^{(d)}$, and return $\mathbf{Y} = (Y^{(1)}, \ldots, Y^{(D)})$. Under Assumption 4.1 that univariate sampling according to $\nu_k^{(d)}$ is efficient, this results in an exact fast sampling algorithm from ν .

When μ is finitely supported, the conclusion of Proposition 4.4 regarding the construction of q_j in (4.14) is equivalent to the result that the size-M vectors,

$$\boldsymbol{q}_j = \otimes_{d \in [D]} \boldsymbol{q}_{\alpha_i^{(d)}}^{(d)}, \qquad j \in [N],$$

are columns vectors for a matrix Q that is an element of col(A). In this finitely supported case, Algorithm 5 formalizes the algorithm steps and cost for sampling from ν when \mathcal{J} is monotone lower.

Algorithm 5: Algorithm from Section 4.3.3 when μ is finitely supported and \mathcal{J} is a monotone lower set.

- 1 Initialization;
- Input: Multi-index set \mathcal{J}
- **2** Compute N_d for $d \in [D]$ using (3.10). $[\mathcal{O}(DN) \text{ cost}];$
- **s** For each $d \in [D]$, perform Initialization step in Algorithm 2. $[\mathcal{O}(\sum_{d=1}^{D} M_d N_d^2) \text{ cost and} \mathcal{O}(\sum_{d=1}^{D} M_d N_d) \text{ storage}];$
- 4 Sampling;
- 5 Choose α uniformly at random from \mathcal{J} . [$\mathcal{O}(1)$ cost];
- For d∈ [D], sample Y^(d) according to ν^(d)_{α(d)}, accomplished using the Sampling step in Algorithm 2. [O(D) cost];
 Output: Y = (Y⁽¹⁾,...,Y^(D))

5 Experiment results

In this section we demonstrate that the exact leverage score sampling proposed in Algorithm 5 results in increased accuracy compared to sketches where we sample rows uniformly at random, or compared to an approximate leverage score sampling method. Although our algorithm and theory apply to the case when the *D*-dimensional domain has infinitely many points (under assumption 4.1), for simplicity we consider a finite domain. In particular, we consider the setup of Section 3.3: each I_d contains M_d points $\{y_m^{(d)}\}_{m\in[M_d]}$ and corresponding weights $\{w_m^{(d)}\}_{m\in[M_d]}$ which define the matrices $\mathbf{A}^{(d)}$ in (3.14). Given an index set \mathcal{J} , the least squares matrix \mathbf{A} is formed as the \mathcal{J} -column subset, given in (3.16), and hence has $M = \prod_{d=1}^{D} M_d$ rows. We assume that \mathcal{J} is monotone lower in the sense of Definition 3.1. We will describe how the data \mathbf{b} is defined in each example that follows, and we perform row-sketched least squares as described in Algorithm 1, which involves defining the sketching measure ν in (2.4). We compare results from the following 3 sampling strategies:

Uniform: Sampling uniformly at random from the M rows of A, i.e.,

$$\nu = \frac{1}{M} \sum_{\boldsymbol{m} \in [(M_1, \dots, M_d)]} \delta_{\left(y_{m_1}^{(1)}, y_{m_2}^{(2)}, \dots, y_{m_D}^{(D)}\right)}.$$

The sampling cost is linear in D, and $\mathcal{O}(1)$ in both M_d and N_d .

TP sampling: We sample according to the leverage scores of the tensor product (TP) $A_{kr} = \otimes_d A^{(d)}$, described in Algorithm 3. Note that this implicitly assumes that $\mathcal{J} \approx [N]$, so that the (difficult to compute) leverage scores of former are approximately those of the (easily computed) latter. The sampling cost is linear in D and M_d , and quadratic in N_d .

Leverage sampling: We efficiently sample according to the exact leverage scores using Algorithm 5. The sampling cost is linear in D and M_d , and quadratic in N_d .

For each method above, we compute the relative residual of the sketched least squares solution \tilde{x} , along with the optimal relative error from the full least-squares solution:

Relative error =
$$\frac{\|\boldsymbol{A}\widetilde{\boldsymbol{x}} - \boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2},$$

Optimal relative error =
$$\frac{\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2}$$

Since "Relative error" is random, we will report its empirical distribution using 100 trials. Our examples are taken from polynomial approximation, specifically from parametric uncertainty quantification using polynomial chaos expansions (PCE). We assume that the nodes and weights $\{y_m^{(d)}, w_m^{(d)}\}$ are identical for each dimension d, and are given as the M_d -point Gauss-Legendre quadrature rule on [-1, 1].

Here we test the empirical results of three different sampling methods involving three numerical examples. We construct the Legendre PCE grid with tensor structure for all examples. In the first one, we investigate the Duffing oscillator under free vibration. Next, the Ishigami function is considered. Lastly, we work on the prediction of the remaining useful life of a Lithium-ion battery. The first two examples are three-dimension problems with larger value of N, while the last example is a high-dimension problem (d = 7) with smaller N. Our empirical results show that our proposed algorithm works in both cases.⁴

5.1 Nonlinear Duffing oscillator

The first problem is to study the uncertainty of the displacement solution $u(\boldsymbol{y},t)$ for a nonlinear single-degree-of-freedom Duffing oscillator [MS15] under free vibration. The system is described by,

$$\ddot{u}(\boldsymbol{y},t) + 2\omega_1\omega_2\dot{u}(\boldsymbol{y},t) + \omega_1^2(u(\boldsymbol{y},t) + \omega_3u^3(\boldsymbol{y},t)) = 0, u(\boldsymbol{y},0) = 1, \qquad \dot{u}(\boldsymbol{y},0) = 0,$$
(5.1)

where the uncertain parameters $\{\omega_i\}_{i=1}^3$ are

$$\omega_1 = 2\pi \ (1 + 0.2y^{(1)}), \quad \omega_2 = 0.05 \ (1 + 0.05y^{(2)}), \quad \omega_3 = -0.5 \ (1 + 0.5y^{(3)}), \tag{5.2}$$

with $\{y^{(d)}\}_{d=1}^3$ being three different uncertain input parameters. We sample $M_d = 20$ Gauss-Legendre nodes values of $y^{(d)}$ between -1 and 1. The QoI we choose is u(y, 4). The dimension D of the problem is 3, and the degrees of polynomial K are considered as 9, 12 for total degree space, 15, 18 for hyperbolic cross space. The reason for different degrees of different polynomials is because total degree space has smaller number of PC basis compared to hyperbolic cross space. The sample size is four times of polynomial subspace size $(4\mathcal{J}_1(K)$ for total degree and $4\mathcal{J}_{HC}(K)$ for hyperbolic cross). Figure 3 shows the empirical cumulative distribution function (CDF) for the relative error of 100 independent trials for the total degree space and the hyperbolic cross space.

 $^{^{4}}$ The codes of our sampling method for Duffing oscillator and Ishigami function are available at https://github.com/CU-UQ/monotone-lower-set.



Figure 3: Distribution of relative error in estimating the displacement $u(\mathbf{y}, 4)$ with a K = 7 (top left) and K = 9 (top right) order total degree PCE with optimal relative errors 2.6×10^{-2} and 2.9×10^{-3} , and a K = 15 (bottom left) and K = 18 (bottom right) order hyperbolic cross PCE with optimal relative errors 6.9×10^{-2} and 3.2×10^{-2} .

The Figure 3 shows that leverage sampling has superior performance compared with uniform sampling. As we discussed in Section 4.3.3, leverage sampling algorithm has the same computation complexity with uniform sampling algorithm under the assumption of monotone lower set \mathcal{J} . Additionally, the leverage sampling leads to a better result than TP sampling. The reason is that TP sampling use the approximate leverage scores as sampling weights while leverage sampling uses exact leverage scores. The empirical result validates our theoretical proof that leverage sampling algorithm can give us a higher accuracy without increasing the complexity.

5.2 Ishigami function

The second example we consider is the Ishigami function. It is a 3-dimensional, nonlinear, and nonmonotonic function with a rich history in PC expansion studies [IH90; DDH18]. The formula is given as

$$f(\boldsymbol{y}) = \sin\left(\pi y^{(1)}\right) + a\sin^2(\pi y^{(2)}) + b(\pi y^{(3)})^4 \sin\left(\pi y^{(1)}\right),\tag{5.3}$$

where $y^{(d)}$ are different variables sampled as Gauss-Legendre nodes between -1 and 1, $M_d = 20$ for all d. In this example we fix the parameters a = 7 and b = 0.1. For the total degree space, we choose K = 7, 9 with $\mathcal{J}_1 = 120, 220$ respectively. For the hyperbolic cross space, K is set to be 15 and 18, with respective subspace basis size \mathcal{J}_{HC} being 110 and 134. Same with the nonlinear Duffing example, we let sample size be 4 times the subspace basis size.



Figure 4: Distribution of the relative error in estimating the Ishigami function $f(\boldsymbol{y})$ with a K = 7 (top left) and K = 9 (top right) order total degree PCE with optimal relative errors 7.0×10^{-3} and 9.5×10^{-4} , and K = 15 (bottom left) and K = 18 (bottom right) order hyperbolic cross PCE with optimal relative errors 9.0×10^{-2} and 7.7×10^{-2} .

From Figure 4, we can find that both TP sampling and leverage sampling have better performance than uniform sampling. The observation we have from Section 5.1 still holds for the Ishigami function.

5.3 Prediction of remaining useful life of batteries

The last example is a high-dimensional problem focusing on model-based estimation of the remaining useful life (RUL) of a Lithium-ion battery (LIB) [SG13; SDG14; San15]. RUL of LIB means the amount of time a battery takes to reach a defined health threshold.

We assume the system model is given by the following equations,

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{f}(t, \boldsymbol{z}(t), \boldsymbol{\theta}(t), \boldsymbol{\nu}(t), \boldsymbol{v}_p(t)),$$
(5.4)

$$\boldsymbol{w}(t) = \boldsymbol{h}(t, \boldsymbol{z}(t), \boldsymbol{\theta}(t), \boldsymbol{\nu}(t), \boldsymbol{v}_m(t)),$$
(5.5)

where $\boldsymbol{z}(t)$ is the state vector, \boldsymbol{f} is the state equation, $\boldsymbol{\theta}(t)$ the model parameter vector, $\boldsymbol{\nu}(t)$ the input vector, $\boldsymbol{v}_p(t)$ the process noise vector, $\boldsymbol{w}(t)$ the output vector, \boldsymbol{h} the output equation, and $\boldsymbol{v}_m(t)$ the measurement noise vector.

In this example, we let the battery be discharged at a constant current represented as a beta random variable with $\alpha = 21.2$ and $\beta = 31.8$. Also, there are three state variables determining the final RUL. We let their state estimations and process noise terms be other uncertain parameters. As a result, the total dimension of this model D is 7. We construct linear transformations mapping the input parameters from intervals between -1 and 1 to their corresponding ranges, and sample the linear map inputs as Gauss-Legendre points between -1 and 1. The degree of polynomials K is fixed to be 3. More details about this model as well as its governing equations are discussed in [SG13; SDG14; San15]. Our numerical results are presented in Figure 5.



Figure 5: Distribution of relative error in estimating the RUL with $M_d = 4$ (top left) and $M_d = 5$ (top right) under total degree space with optimal relative errors 6.6×10^{-4} and 9.3×10^{-4} , and $M_d = 4$ (bottom left) and $M_d = 5$ (bottom right) under hyperbolic cross space with optimal relative errors 2.2×10^{-3} and 1.1×10^{-3} .

It is noticeable that the performance of uniform sampling is similar with (even outperforms) TP sampling in the top left part of Figure 5. It is because when $M_d = K + 1$, Q becomes a square matrix, which makes TP sampling degenerates into uniform sampling. To avoid this situation, the leverage sampling based on our purposed algorithm (Algorithm 5) is clearly a better option. With the exclusion of it, the result of RUL problem in general matches with our previous observations from Section 5.1 and Section 5.2.

6 Conclusion

The notion of leverage scores for row sketches in least squares problems allows one to accurately compute approximate solutions in an efficient, randomized fashion. This same idea appears as the induced measure in the function approximation setting. We have presented a unified view of these ideas, and surveyed various accuracy guarantees from both communities when using leverage score-like sampling.

Leverage scores are relatively expensive to compute for large matrices in general, but can be accomplished much more quickly when matrices have special structure. In this paper we have exploited non-trivial Kronecker product-like structure, corresponding to matrices that are a "monotone lower" column subset of a Kronecker product matrix, to devise algorithms for efficiently row-sampling from the full matrix. In this case the cost of sampling is substantially reduced, and one can *exactly* sample according to the leverage scores with tractable computational effort.

Acknowledgments

This work was supported by the AFOSR awards FA9550-20-1-0138 and FA9550-20-1-0188 with Dr. Fariba Fahroo as the program manager. The views expressed in the article do not necessarily represent the views of the AFOSR or the U.S. Government.

References

[ABW22] B. Adcock, S. Brugiapaglia, and C. G. Webster. Sparse Polynomial Approximation of High-Dimensional Functions. SIAM, 2022. ISBN: 978-1-61197-688-5.

- [Ahl+20] T. D. Ahle, M. Kapralov, J. B. Knudsen, R. Pagh, A. Velingker, D. P. Woodruff, and A. Zandieh. "Oblivious Sketching of High-Degree Polynomial Kernels". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2020, pp. 141–160.
- [AC09] N. Ailon and B. Chazelle. "The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors". In: *SIAM Journal on Computing* 39.1 (2009), pp. 302–322.
- [AM15] A. Alaoui and M. W. Mahoney. "Fast Randomized Kernel Ridge Regression with Statistical Guarantees". In: Advances in Neural Information Processing Systems. Vol. 28. Curran Associates, Inc., 2015.
- [ANW14] H. Avron, H. L. Nguyen, and D. P. Woodruff. "Subspace Embeddings for the Polynomial Kernel". In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. Cambridge, MA, USA: MIT Press, 2014, pp. 2258–2266.
- [BKW21] S. Bamberger, F. Krahmer, and R. Ward. "Johnson-Lindenstrauss Embeddings with Kronecker Structure". In: arXiv preprint arXiv:2106.13349 (2021). arXiv: 2106.13349.
- [BBK18] C. Battaglino, G. Ballard, and T. G. Kolda. "A Practical Randomized CP Tensor Decomposition". In: SIAM Journal on Matrix Analysis and Applications 39.2 (2018), pp. 876–901.
- [BBN11] R. Berman, S. Boucksom, and D. Nyström. "Fekete points and convergence towards equilibrium measures on complex manifolds". In: Acta Mathematica 207.1 (2011), pp. 1–27. ISSN: 0001-5962. DOI: 10.1007/s11511-011-0067-x.
- [Ber09a] R. J. Berman. "Bergman kernels and equilibrium measures for line bundles over projective manifolds". In: American Journal of Mathematics 131.5 (2009), pp. 1485–1524. ISSN: 1080-6377. DOI: 10.1353/ajm.0.0077.
- [Ber09b] R. J. Berman. "Bergman kernels for weighted polynomials and weighted equilibrium measures of \mathbb{C}^n ". In: Indiana University Mathematics Journal 58.4 (2009), pp. 1921–1946. ISSN: 0022-2518.
- [BBB15] D. J. Biagioni, D. Beylkin, and G. Beylkin. "Randomized Interpolative Decomposition of Separated Representations". In: *Journal of Computational Physics* 281.C (Jan. 2015), pp. 116–134. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2014.10.009.
- [CLV17] D. Calandriello, A. Lazaric, and M. Valko. "Distributed Adaptive Sampling for Kernel Matrix Approximation". In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. ISSN: 2640-3498. PMLR, 2017, pp. 1421–1429.
- [Che+20] K. Chen, Q. Li, K. Newton, and S. J. Wright. "Structured Random Sketching for PDE Inverse Problems". In: SIAM Journal on Matrix Analysis and Applications 41.4 (2020), pp. 1742–1770. DOI: 10.1137/20M1310497. eprint: https://doi.org/10.1137/20M1310497.
- [Che+16] D. Cheng, R. Peng, Y. Liu, and I. Perros. "SPALS: Fast Alternating Least Squares via Implicit Leverage Scores Sampling". In: Advances In Neural Information Processing Systems. 2016, pp. 721– 729.
- [CW17] K. L. Clarkson and D. P. Woodruff. "Low-Rank Approximation and Regression in Input Sparsity Time". In: Journal of the ACM 63.6 (Feb. 2017), 54:1–54:45. ISSN: 0004-5411. DOI: 10.1145/3019134.
- [CM17a] A. Cohen and G. Migliorati. "Optimal weighted least-squares methods". In: SMAI J. Comput. Math. 3 (2017), pp. 181–203. ISSN: 2426-8399. DOI: 10.5802/smai-jcm.24.
- [CM17b] A. Cohen and G. Migliorati. "Optimal weighted least-squares methods". In: SMAI Journal of Computational Mathematics 3 (2017). arxiv:1608.00512 [math.NA], pp. 181–203. ISSN: 2426-8399. DOI: 10.5802/smai-jcm.24.
- [Dia+19] H. Diao, R. Jayaram, Z. Song, W. Sun, and D. P. Woodruff. "Optimal Sketching for Kronecker Product Regression and Low Rank Approximation". In: arXiv preprint arXiv:1909.13384 (2019). arXiv: 1909.13384.
- [Dia+18] H. Diao, Z. Song, W. Sun, and D. Woodruff. "Sketching for Kronecker Product Regression and P-Splines". en. In: Proceedings of the 21st International Conference on Artificial Intelligence and Statistics. 2018, pp. 1299–1308.
- [DDH18] P. Diaz, A. Doostan, and J. Hampton. "Sparse polynomial chaos expansions via compressed sensing and D-optimal design". In: Computer Methods in Applied Mechanics and Engineering 336 (July 2018), pp. 640–666. ISSN: 00457825. DOI: 10.1016/j.cma.2018.03.020.
- [DMN17] T.-C. Dinh, X. Ma, and V.-A. Nguyên. "On the asymptotic behavior of Bergman kernels for positive line bundles". In: *Pacific Journal of Mathematics* 289.1 (2017), pp. 71–89. ISSN: 0030-8730. DOI: 10.2140/pjm.2017.289.71.

- [Dri+12] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. "Fast Approximation of Matrix Coherence and Statistical Leverage". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 3475–3506.
- [DG16] D. Dũng and M. Griebel. "Hyperbolic cross approximation in infinite dimensions". In: Journal of Complexity 33 (2016), pp. 55–88. ISSN: 0885-064X. DOI: 10.1016/j.jco.2015.09.006.
- [DTU18] D. Dũng, V. Temlyakov, and T. Ullrich. Hyperbolic Cross Approximation. Springer, 2018. ISBN: 978-3-319-92240-9.
- [FGF21] M. Fahrbach, M. Ghadiri, and T. Fu. "Fast Low-Rank Tensor Decomposition by Ridge Leverage Score Sampling". In: arXiv preprint arXiv:2107.10654 (2021). arXiv: 2107.10654.
- [FH20] D. W. Fausett and H. Hashish. "Overview of QR Methods for Large Least Squares Problems Involving Kronecker Products". In: Overview of QR Methods for Large Least Squares Problems Involving Kronecker Products. De Gruyter, 2020, pp. 71–80. ISBN: 978-3-11-231409-8. DOI: 10.1515/ 9783112314098-009.
- [FF94] D. W. Fausett and C. T. Fulton. "Large Least Squares Problems Involving Kronecker Products". In: SIAM Journal on Matrix Analysis and Applications 15.1 (1994), pp. 219–227. ISSN: 0895-4798. DOI: 10.1137/S0895479891222106.
- [FFH97] D. W. Fausett, C. T. Fulton, and H. Hashish. "Improved parallel QR method for large least squares problems involving Kronecker products". en. In: *Journal of Computational and Applied Mathematics* 78.1 (1997), pp. 63–78. ISSN: 0377-0427. DOI: 10.1016/S0377-0427(96)00109-4.
- [HD15] J. Hampton and A. Doostan. "Coherence motivated sampling and convergence analysis of least squares polynomial Chaos regression". In: Computer Methods in Applied Mechanics and Engineering 290 (2015), pp. 73–97. ISSN: 0045-7825. DOI: 10.1016/j.cma.2015.02.006.
- [IH90] T Ishigami and T Homma. "An importance quantification technique in uncertainty analysis for computer models". eng. In: [1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis. IEEE Comput. Soc. Press, 1990, pp. 398–403. ISBN: 9780818621079.
- [Iwe+21] M. A. Iwen, D. Needell, E. Rebrova, and A. Zare. "Lower Memory Oblivious (Tensor) Subspace Embeddings with Fewer Random Bits: Modewise Methods for Least Squares". In: SIAM Journal on Matrix Analysis and Applications 42.1 (2021), pp. 376–416. DOI: 10.1137/19M1308116. eprint: https://doi.org/10.1137/19M1308116.
- [JKW20] R. Jin, T. G. Kolda, and R. Ward. "Faster Johnson-Lindenstrauss Transforms via Kronecker Products". In: Information and Inference: A Journal of the IMA (Oct. 2020). ISSN: 2049-8772. DOI: 10.1093/imaiai/iaaa028. eprint: https://academic.oup.com/imaiai/advance-articlepdf/doi/10.1093/imaiai/iaaa028/34904656/iaaa028.pdf.
- [LK20] B. W. Larsen and T. G. Kolda. "Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition". In: arXiv preprint arXiv:2006.16438 (2020). arXiv: 2006.16438.
- [MS22] L. Ma and E. Solomonik. "Cost-Efficient Gaussian Tensor Network Embeddings for Tensor-structured Inputs". In: *arXiv preprint arXiv:2205.13163* (2022). arXiv: 2205.13163.
- [Mah11] M. W. Mahoney. "Randomized Algorithms for Matrices and Data". In: Foundations and Trends in Machine Learning 3.2 (2011), pp. 123–224.
- [MS15] C. V. Mai and B. Sudret. "Polynomial Chaos Expansions For Damped Oscillators". In: 12th International Conference on Applications of Statistics and Probability in Civil Engineering. Vancouver, Canada, July 2015.
- [Mal22] O. A. Malik. "More Efficient Sampling for Tensor Decomposition With Worst-Case Guarantees". In: Proceedings of the 39th International Conference on Machine Learning. Vol. 162. Proceedings of Machine Learning Research. PMLR, July 2022, pp. 14887–14917.
- [MB20] O. A. Malik and S. Becker. "Guarantees for the Kronecker Fast Johnson-Lindenstrauss Transform Using a Coherence and Sampling Argument". en. In: *Linear Algebra and its Applications* 602 (Oct. 2020), pp. 120–137. ISSN: 0024-3795. DOI: 10.1016/j.laa.2020.05.004.
- [MB21] O. A. Malik and S. Becker. "A Sampling-Based Method for Tensor Ring Decomposition". In: Proceedings of the 38th International Conference on Machine Learning. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 7400–7411.
- [MMV19] A. Marco, J.-J. Martínez, and R. Viaña. "Least squares problems involving generalized Kronecker products and application to bivariate polynomial regression | SpringerLink". In: Numerical Algorithms (2019), pp. 21–39.

- [MT20] P.-G. Martinsson and J. Tropp. "Randomized Numerical Linear Algebra: Foundations & Algorithms". In: arXiv preprint arXiv:2002.01387 (2020). arXiv: 2002.01387.
- [Mit20] B. S. Mityagin. "The Zero Set of a Real Analytic Function". In: *Mathematical Notes* 107.3-4 (Mar. 2020), pp. 529–530. DOI: 10.1134/s0001434620030189.
- [Nar18] A. Narayan. "Computation of induced orthogonal polynomial distributions". In: Electronic Transactions on Numerical Analysis 50 (2018). arXiv:1704.08465 [math], pp. 71–97. DOI: 10.1553/etna_ vol50s71.
- [NJZ17] A. Narayan, J. Jakeman, and T. Zhou. "A Christoffel function weighted least squares algorithm for collocation approximations". In: *Mathematics of Computation* 86.306 (2017). arXiv: 1412.4305 [math.NA], pp. 1913–1947. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/mcom/3192.
- [NN13] J. Nelson and H. L. Nguyên. "OSNAP: Faster Numerical Linear Algebra Algorithms via Sparser Subspace Embeddings". In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. 2013, pp. 117–126. DOI: 10.1109/F0CS.2013.21.
- [Nev86] P. Nevai. "Géza Freud, orthogonal polynomials and Christoffel functions. A case study". In: Journal of Approximation Theory 48.1 (1986), pp. 3–167. ISSN: 0021-9045. DOI: 10.1016/0021-9045(86)90016-X.
- [Pag13] R. Pagh. "Compressed Matrix Multiplication". In: ACM Transactions on Computation Theory 5.3 (Aug. 2013), 9:1–9:17. ISSN: 1942-3454. DOI: 10.1145/2493252.2493254.
- [PP13] N. Pham and R. Pagh. "Fast and Scalable Polynomial Kernels via Explicit Feature Maps". In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '13. New York, NY, USA: ACM, 2013, pp. 239–247. ISBN: 978-1-4503-2174-7. DOI: 10.1145/2487575.2487591.
- [RR20] B. Rakhshan and G. Rabusseau. "Tensorized Random Projections". In: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, Aug. 2020, pp. 3306–3316.
- [RR21] B. T. Rakhshan and G. Rabusseau. "Rademacher Random Projections with Tensor Networks". In: NeurIPS Workshop on Quantum Tensor Networks in Machine Learning. 2021.
- [Rud+18] A. Rudi, D. Calandriello, L. Carratino, and L. Rosasco. "On Fast Leverage Score Sampling and Optimal Learning". In: Advances in Neural Information Processing Systems. Vol. 31. Curran Associates, Inc., 2018.
- [San15] S. Sankararaman. "Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction". In: *Mechanical Systems and Signal Processing* 52-53 (2015), pp. 228– 247. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2014.05.029.
- [SDG14] S. Sankararaman, M. Daigle, and K. Goebel. "Uncertainty Quantification in Remaining Useful Life Prediction Using First-Order Reliability Methods". In: *Reliability, IEEE Transactions on* 63 (June 2014), pp. 603–619. DOI: 10.1109/TR.2014.2313801.
- [SG13] S. Sankararaman and K. Goebel. "Uncertainty Quantification in Remaining Useful Life of Aerospace Components using State Space Models and Inverse FORM". In: 2013.
- [Sar06] T. Sarlos. "Improved Approximation Algorithms for Large Matrices via Random Projections". In: 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06). ISSN: 0272-5428. 2006, pp. 143–152. DOI: 10.1109/FOCS.2006.37.
- [Ses17] P. Seshadri. Kronecker Product Least Squares. arXiv:1705.08731 [math]. 2017. DOI: 10.48550/arXiv. 1705.08731.
- [SA22] P. F. Shustin and H. Avron. "Semi-Infinite Linear Regression and Its Applications". In: SIAM Journal on Matrix Analysis and Applications 43.1 (2022), pp. 479–511. ISSN: 0895-4798. DOI: 10.1137/ 21M1411950.
- [Sim08] B. Simon. "The Christoffel-Darboux Kernel". In: Perspectives in Partial Differential Equations, Harmonic Analysis and Applications. Ed. by D. Mitrea and M. Mitrea. Vol. 79. Proceedings of Symposia in Pure Mathematics. arXiv:0806.1528 [math]. American Mathematical Society, 2008.
- [Son+21] Z. Song, D. Woodruff, Z. Yu, and L. Zhang. "Fast Sketching of Polynomial Kernels of Polynomial Degree". In: Proceedings of the 38th International Conference on Machine Learning. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 9812–9823.
- [Sun+18] Y. Sun, Y. Guo, J. A. Tropp, and M. Udell. "Tensor Random Projection for Low Memory Dimension Reduction". In: NeurIPS Workshop on Relational Representation Learning. 2018.

- [TT15] A. Townsend and L. N. Trefethen. "Continuous analogues of matrix factorizations". In: Proc. R. Soc. A 471.2173 (2015), p. 20140585. ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.2014.0585.
- [Tre17] L. Trefethen. "Multivariate polynomial approximation in the hypercube". In: Proceedings of the American Mathematical Society 145.11 (2017), pp. 4837–4844. ISSN: 0002-9939, 1088-6826. DOI: 10.1090/ proc/13623.
- [Vos91] M. Vose. "A linear algorithm for generating random numbers with a given distribution". In: IEEE Transactions on Software Engineering 17.9 (1991). Conference Name: IEEE Transactions on Software Engineering, pp. 972–975. ISSN: 1939-3520. DOI: 10.1109/32.92917.
- [WZ20] D. Woodruff and A. Zandieh. "Near Input Sparsity Time Kernel Embeddings via Adaptive Sampling". In: Proceedings of the 37th International Conference on Machine Learning. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 10324–10333.
- [WZ22] D. Woodruff and A. Zandieh. "Leverage Score Sampling for Tensor Product Matrices in Input Sparsity Time". In: Proceedings of the 39th International Conference on Machine Learning. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 23933–23964.
- [Woo14] D. P. Woodruff. "Sketching as a Tool for Numerical Linear Algebra". In: Foundations and Trends in Theoretical Computer Science 10.1-2 (2014), pp. 1–157.
- [Xu95] Y. Xu. "Christoffel Functions and Fourier Series for Multivariate Orthogonal Polynomials". In: Journal of Approximation Theory 82.2 (1995), pp. 205–239. ISSN: 0021-9045. DOI: 10.1006/jath.1995.
 1075.

A "Structural" conditions

We present two main results in this section that are well-known conditions for row-sketched least squares solutions. These conditions are often referred to as "structural conditions" for sketched least squares procedures that enable convergence statements.

A.1 The ϵ -embedding condition

This section seeks to prove the following subspace embedding result via randomization. The first result we are aware of in this direction is [Sar06], although our presentation below more closely follows [Woo14; CM17a].

Lemma A.1. Fix $\delta, \epsilon \in (0, 1)$. With τ constructed as in (2.4), suppose that

$$K \ge \frac{3\log\left(\frac{4N}{\delta}\right)}{\epsilon^2}N.\tag{A.1}$$

Then with probability at least $1 - \frac{\delta}{2}$, we have

$$(1-\epsilon)\|v\|^{2} \le \|v\|_{\tau}^{2} \le (1+\epsilon)\|v\|^{2}, \qquad \forall v \in V.$$
(A.2)

This result ensures that the μ -norm $\|\cdot\|$ and the (randomly generated) τ -norm $\|\cdot\|_{\tau}$ are equivalent with respect to the constant ϵ . Due to the unbiasedness condition (2.5), this asymptotic equivalence is expected, and the Lemma makes this equivalence statement quantitative in the finite-sample regime using a concentration inequality.

Proof. The main tool is a matrix Chernoff bound, but first we require some notation. Define the $N \times N$ Gram matrices involving μ and τ inner products,

$$(\boldsymbol{G}_{\mu})_{n,k} = \langle u_n, u_k \rangle_{\mu} = \delta_{n,k}, \qquad (\boldsymbol{G}_{\tau})_{n,k} = \langle u_n, u_k \rangle_{\tau}, \qquad (A.3)$$

where we recall from (2.10) that u_n is an L^2_{μ} -orthonormal basis for V so that $G_{\mu} = I$. With these Gram matrices, define the probabilistic event,

$$C = \{ \|\boldsymbol{G}_{\tau} - \boldsymbol{G}_{\mu}\| \leq \epsilon \} = \{ \|\boldsymbol{G}_{\tau} - \boldsymbol{I}\| \leq \epsilon \},\$$

where $\|\cdot\|$ on vectors is the standard induced ℓ^2 (spectral) norm. By using the fact that,

$$v = \sum_{n \in [N]} x_n u_n \quad \Longrightarrow \quad \|v\|_\mu^2 = oldsymbol{x}^T oldsymbol{G}_\mu oldsymbol{x}, \ \|v\|_ au^2 = oldsymbol{x}^T oldsymbol{G}_ au oldsymbol{x},$$

then we have that the event C implies our desired embedding (A.2). Thus, to complete the proof we need only prove,

$$\Pr(C) \ge 1 - \frac{\delta}{2}.\tag{A.4}$$

To analyze this probability, note from (2.4) that G_{τ} is equal to,

$$\boldsymbol{G}_{\tau} = \frac{1}{K} \sum_{k \in [K]} \boldsymbol{g}_{k} \boldsymbol{g}_{k}^{T}, \qquad (A.5)$$

where $\boldsymbol{g}_k, k \in [K]$ are iid vectors in \mathbb{R}^N given by,

$$oldsymbol{g}_k = \sqrt{rac{\mathrm{d} \mu}{\mathrm{d}
u}(oldsymbol{z}_k)}oldsymbol{u}(oldsymbol{z}_k), \qquad oldsymbol{u}(oldsymbol{z}) = \left(u_1(oldsymbol{z}), \ldots, u_N(oldsymbol{z})
ight)^T, \qquad oldsymbol{z}_k \sim
u$$

From (2.11), we have that $d\mu/d\nu(\boldsymbol{z}) = N/\|\boldsymbol{u}(\boldsymbol{z})\|_2^2$, and so,

$$\mathbb{E} \|\boldsymbol{g}_k \boldsymbol{g}_k^T\|^2 = \mathbb{E} \|\boldsymbol{g}_k\|_2^2 = \int_I \|\boldsymbol{u}(\boldsymbol{z})\|_2^2 \frac{\mathrm{d}\mu}{\mathrm{d}\nu_f}(\boldsymbol{z}) \mathrm{d}\nu_f(\boldsymbol{z}) = \int_I N \mathrm{d}\nu_f(\boldsymbol{z}) = N,$$
$$\mathbb{E} (\boldsymbol{g}_k \boldsymbol{g}_k^T) = \int_I \boldsymbol{u}(\boldsymbol{z}) \boldsymbol{u}(\boldsymbol{z})^T \frac{\mathrm{d}\mu}{\mathrm{d}\nu_f}(\boldsymbol{z}) \mathrm{d}\nu_f(\boldsymbol{z}) = \int_I \boldsymbol{u}(\boldsymbol{z}) \boldsymbol{u}(\boldsymbol{z})^T \mathrm{d}\mu(\boldsymbol{z}) = \boldsymbol{I}$$

Armed with these expectations, we can exercise the matrix Chernoff bound on the iid sum (A.5), which states:

$$\Pr\left(\|\boldsymbol{G}_{\tau} - \boldsymbol{I}\| > \epsilon\right) = \Pr\left(\|K\boldsymbol{G}_{\tau} - K\boldsymbol{I}\| > K\epsilon\right) \le 2N \left[\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right]^{K/N}.$$
 (A.6a)

We have,

$$N\left[\frac{e^{\epsilon}}{(1+\epsilon)^{(1+\epsilon)}}\right]^{K/N} \le \frac{\delta}{4} \quad \Longleftrightarrow \quad K \ge \frac{N\log\left(\frac{4N}{\delta}\right)}{(1+\epsilon)\log(1+\epsilon)-\epsilon},\tag{A.6b}$$

and

$$(1+\epsilon)\log(1+\epsilon) - \epsilon \ge \log\left(\frac{4}{e}\right)\epsilon^2 \ge \frac{1}{3}\epsilon^2, \qquad \epsilon \in (0,1)$$
(A.6c)

Combining the three results (A.6) shows that (2.15) implies (A.4).

A.2 τ -Sketches of b_{\perp}

Lemma A.2. Define $b_{\perp} \coloneqq f - v^*$, where v^* is the true least squares solution in (2.1). Given c > 0, $\delta \in (0, 2)$, let τ be randomly constructed by sampling $z_k \sim \nu$ as described above with

$$K \ge \frac{2N}{c\delta},\tag{A.7}$$

With $(u_n)_{n\in[N]}$ an L^2_μ -orthonormal basis for V, then,

$$\sum_{n \in [N]} \mathbb{E} \left| \left\langle b_{\perp}, u_n \right\rangle_{\tau} \right|^2 = \frac{N}{K} \| b_{\perp} \|^2, \tag{A.8}$$

$$\Pr\left[\sum_{n\in[N]} |\langle b_{\perp}, u_n \rangle_{\tau}|^2 > c \|b_{\perp}\|_2^2\right] \le \frac{\delta}{2}.$$
(A.9)

Proof. The main tool is Markov's inequality applied to the non-negative random variable $\sum_{n \in [N]} |\langle b_{\perp}, u_n \rangle_{\tau}|^2$, stating that for any c > 0, then

$$\Pr\left[\sum_{n\in[N]} |\langle b_{\perp}, u_n \rangle_{\tau}|^2 > c \|b_{\perp}\|_2^2\right] \le \frac{1}{c\|b_{\perp}\|^2} \mathbb{E}\sum_{n\in[N]} |\langle b_{\perp}, u_n \rangle_{\tau}|^2.$$
(A.10)

To compute the right-hand side, note that

$$\left| \langle b_{\perp}, u_n \rangle_{\tau} \right|^2 = \frac{1}{K^2} \left| \sum_{k \in [K]} A_{n,k} \right|^2, \qquad \qquad A_{n,k} \coloneqq \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(\boldsymbol{z}_k) u_n(\boldsymbol{z}_k) b_{\perp}(\boldsymbol{z}_k).$$

Since \boldsymbol{z}_k are iid, then for each n we have,

$$\mathbb{E}A_{n,k} = 0 \text{ and } \{A_{n,k}\}_{k \in [K]} \text{ iid} \Longrightarrow \mathbb{E} \left| \sum_{k \in [K]} A_{n,k} \right|^2 = \sum_{k \in [K]} \operatorname{var}A_{n,k} = K \operatorname{var}A_{n,1}$$

The variance of $A_{n,1}$ is given by,

$$\operatorname{var} A_{n,1} = \int |u_n(\boldsymbol{y})|^2 |b_{\perp}(\boldsymbol{y})|^2 \left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(\boldsymbol{y})\right)^2 \mathrm{d}\nu(\boldsymbol{y}) = \int |u_n(\boldsymbol{y})|^2 |b_{\perp}(\boldsymbol{y})|^2 \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(\boldsymbol{y}) \mathrm{d}\mu(\boldsymbol{y}).$$
(A.11)

Therefore, we have,

$$\mathbb{E}\sum_{n\in[N]} |\langle b_{\perp}, u_n \rangle_{\tau}|^2 = \frac{1}{K} \sum_{n\in[N]} \int |u_n(\boldsymbol{y})|^2 |b_{\perp}(\boldsymbol{y})|^2 \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(\boldsymbol{y}) \mathrm{d}\mu(\boldsymbol{y})$$
$$= \frac{1}{K} \int \left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(\boldsymbol{y}) \sum_{n\in[N]} |u_n(\boldsymbol{y})|^2 \right) |b_{\perp}(\boldsymbol{y})|^2 \mathrm{d}\mu(\boldsymbol{y})$$
$$\stackrel{(2.11)}{=} \frac{N}{K} \int |b_{\perp}(\boldsymbol{y})|^2 \mathrm{d}\mu(\boldsymbol{y}) = \frac{N}{K} \|b_{\perp}\|^2 \overset{(A.7)}{\leq} \frac{1}{2} c\delta \|b_{\perp}\|^2,$$
n (A.10) proves (A.8).

and using this in (A.10) proves (A.8).

The quantity (A.8) is a bound on how b_{\perp} , which is L^2_{μ} -orthogonal to V, projects onto V according to the measure τ . Since $\tau \neq \nu$, this projection is generally not zero; in other contexts this projection is called aliasing error.

Proof of Theorem 2.7 В

We assume that $b \notin V$ so that dim $V_b = N + 1$. The same result holds if $b \in V$ by replacing N + 1by N and (ν_{μ,V_b}, V_b) by $(\mu_{\mu,V}, V)$ in what follows.

First we note that the first-order optimality conditions for the optimization problems (2.1) and (2.6)defining v^* and \tilde{v} , respectively, imply that for $v \in V$ we have the orthogonality relations,

$$\langle v^* - f, v \rangle_{\mu} = 0,$$
 $\langle \tilde{v} - f, v \rangle_{\tau} = 0.$ (B.1)

The main idea for the proof is to show and leverage an ϵ -embedding property for the sketching measure τ relative to μ . This sketching property is the condition:

$$(1-\epsilon)\|v\|^{2} \le \|v\|_{\tau}^{2} \le (1+\epsilon)\|v\|^{2}, \qquad \forall v \in V_{f}.$$
(B.2)

This property holds with some probability since τ is randomly generated. Assuming (B.2) is true, then we have

$$\begin{split} \|\widetilde{v} - f\|^{2} &\stackrel{(*)}{=} \|v^{*} - f\|^{2} + \|\widetilde{v} - v^{*}\|^{2} \\ &\stackrel{(B.2)}{\leq} \|v^{*} - f\|^{2} + \frac{1}{1 - \epsilon} \|\widetilde{v} - v^{*}\|_{\tau}^{2} \\ &\stackrel{(*)}{=} \|v^{*} - f\|^{2} + \frac{1}{1 - \epsilon} \left(\|v^{*} - f\|_{\tau}^{2} - \|\widetilde{v} - f\|_{\tau}^{2} \right) \\ &\stackrel{(B.2)}{\leq} \|v^{*} - f\|^{2} + \frac{1 + \epsilon}{1 - \epsilon} \|v^{*} - f\|^{2} - \|\widetilde{v} - f\|^{2}, \end{split}$$

where the equalities marked (*) utilize (B.1) and the Pythagorean theorem. Rearranging the inequality above and using $\epsilon < 1/2$ yields the desired result (2.16). Thus, the proof will be completed by showing (B.2) is an event that occurs with probability at least $1-\delta$; Lemma A.1 applied to the dimension-(N+1)space V_b shows that (2.15) ensures this property.

Proof of Theorem 2.8 \mathbf{C}

We first prove a more general result, which holds for any $\delta, \epsilon \in (0, 1)$ and c > 0, namely that if

$$K \ge N \max\left\{\frac{3\log\left(\frac{4N}{\delta}\right)}{\epsilon^2}, \frac{2}{c\delta}\right\},\tag{C.1}$$

then

$$\|\widetilde{v} - b\|^2 \le \left(1 + \frac{c}{(1-\epsilon)^2}\right) \|v^* - b\|^2 \quad \text{w/ prob.} \ge 1 - \delta.$$
(C.2)

Note that Theorem 2.8 is simply (C.1) and (C.2) with the replacement $c = \epsilon (1 - \epsilon)^2$. Therefore, we focus on proving (C.2). By the Pythagorean theorem, we have,

$$\|\widetilde{v} - b\|^{2} = \|v^{*} - b\|^{2} + \|v^{*} - \widetilde{v}\|^{2},$$

so that the result is proven if we can show,

$$(2.17) \Longrightarrow \|v^* - \widetilde{v}\|^2 \le \frac{c}{(1-\epsilon)^2} \|v^* - b\|^2 \text{ w/ prob. at least } 1 - \delta.$$

We express $\tilde{v}, v^* \in V$ through their coordinates \tilde{w}, w^* , respectively, in the L^2_{μ} -orthonormal basis u_n :

$$\widetilde{v} = \sum_{n \in [N]} \widetilde{w}_n u_n, \quad v^* = \sum_{n \in [N]} w_n^* u_n, \quad \Rightarrow \|\widetilde{v} - v^*\| = \|\widetilde{w} - w^*\|_2$$

Define the $K \times N$ matrix \boldsymbol{U} as,

$$(\boldsymbol{U})_{k,n} = \sqrt{v_k} u_n(\boldsymbol{z}_k),$$

so that by (A.3), $\boldsymbol{G}_{\tau} = \boldsymbol{U}^T \boldsymbol{U}$. Then we have,

(2.17)
$$\xrightarrow{\text{Lemma A.1}} \|\boldsymbol{G}_{\tau} - \boldsymbol{I}\| \le \epsilon \text{ w/ prob. at least } 1 - \delta/2 \Longrightarrow \|\boldsymbol{G}_{\tau}^{-1}\| \le \frac{1}{1 - \epsilon} \text{ wp } 1 - \delta/2,$$

and in particular G_{τ} is invertible with this same probability. Under this event, both \tilde{w} and w^* are given by unique solutions to finite least squares problems:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{arg\,min}} \|\boldsymbol{U}\boldsymbol{w} - \boldsymbol{v}^*\|, \qquad \qquad \widetilde{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{arg\,min}} \|\boldsymbol{U}\boldsymbol{w} - \tilde{\boldsymbol{b}}\|, \qquad (C.3)$$

where $(\boldsymbol{v}^*)_k = \sqrt{v_k} v^*(\boldsymbol{z}_k)$ and $\tilde{\boldsymbol{b}} = \sqrt{v_k} b(\boldsymbol{z}_k)$. Under the event that \boldsymbol{G}_{τ} is invertible, then the normal equations imply,

$$\|\widetilde{\boldsymbol{w}} - \boldsymbol{w}^*\|_2 \le \|\boldsymbol{G}_{\tau}^{-1}\|_2 \|\boldsymbol{U}^T(\widetilde{\boldsymbol{b}} - \boldsymbol{v}^*)\|_2 \le \frac{1}{1 - \epsilon} \|\boldsymbol{U}^T(\widetilde{\boldsymbol{b}} - \boldsymbol{v}^*)\|_2 \text{ w/ prob. at least } 1 - \delta/2.$$
(C.4)

In addition, note that,

(C.1)
$$\xrightarrow{\text{Lemma A.2}} \left\| \boldsymbol{U}^T \left(\boldsymbol{b} - \boldsymbol{v}^* \right) \right\|_2^2 \le c \|b_{\perp}\|^2 = c \|v^* - b\|^2$$

which proves (C.2).

D Proof of (2.19)

The proof is similar to the proof of Theorem 2.8 in Section C, so we omit some details. Given $\epsilon>0,$ choose

$$\widetilde{\epsilon} = \frac{\sqrt{3(1-\delta)\epsilon \log(2N/\delta)}}{1+\sqrt{3(1-\delta)\epsilon \log(2N/\delta)}} \in (0,1),$$
(D.1)

and assume the sampling requirement

$$K \ge \frac{3\log\left(\frac{2N}{\delta}\right)}{\tilde{\epsilon}^2}N,\tag{D.2}$$

then Lemma A.1 ensures that

$$\Pr(C) \ge 1 - \delta, \qquad \qquad C = \left\{ \|\boldsymbol{G}_{\tau}^{-1}\|_{2} \le \frac{1}{1 - \tilde{\epsilon}} \right\}. \tag{D.3}$$

By the Pythagorean Theorem, we have

$$\mathbb{E}[\|\tilde{v} - b\|^2 \mid C] = \|v^* - b\|^2 + \mathbb{E}[\|v^* - \tilde{v}\|^2 \mid C]$$
(D.4)

Since for any measurable event C and non-negative random variable X, we have,

$$\mathbb{E}[X|C] \le \frac{\mathbb{E}X}{\Pr(C)},\tag{D.5}$$

then the second term on the right-hand side of (D.4) can be bounded as,

$$\mathbb{E}[\|\boldsymbol{v}^* - \tilde{\boldsymbol{v}}\|^2 \mid C] \stackrel{(C.3)}{=} \mathbb{E}[\|\boldsymbol{w}^* - \tilde{\boldsymbol{w}}\|^2 \mid C] \stackrel{(C.4)}{\leq} \frac{1}{(1 - \tilde{\epsilon})^2} \mathbb{E}[\|\boldsymbol{U}^*(\boldsymbol{b} - \boldsymbol{v}^*)\|^2 \mid C]$$

$$\stackrel{(D.5)}{\leq} \frac{1}{\Pr(C)(1 - \tilde{\epsilon})^2} \mathbb{E}[\|\boldsymbol{U}^*(\boldsymbol{b} - \boldsymbol{v}^*)\|^2]$$

$$\stackrel{(A.8),(D.3)}{\leq} \frac{N}{K(1 - \delta)(1 - \tilde{\epsilon})^2} \|\boldsymbol{v}^* - \boldsymbol{b}\|^2.$$

Combining these facts along with (D.2) yields,

$$\mathbb{E}\|\widetilde{v}-b\|^2 \le \left(1 + \left(\frac{\widetilde{\epsilon}}{1-\widetilde{\epsilon}}\right)^2 \frac{1}{3(1-\delta)\log(2N/\delta)}\right) \|v^*-b\|^2 \stackrel{(\mathrm{D}.1)}{=} (1+\epsilon) \|v^*-b\|^2,$$

which corresponds to the sampling requirement

$$K \geq \frac{3\log\left(\frac{2N}{\delta}\right)}{\tilde{\epsilon}^2} N \stackrel{(\mathrm{D.1})}{=} \frac{N}{\epsilon} \left(1 + \sqrt{3(1-\delta)\epsilon\log\left(\frac{2N}{\delta}\right)}\right)^2.$$

The proof is finished by noting that the sampling requirement in (2.19) is stronger than the above since $(a+b)^2 \leq 2a^2 + 2b^2$ for all real a, b, and with the removal of the $1-\delta$ term, and so implies the above.

E Proof of (2.22)

The proof adds one ingredient to the proof of (2.19) in appendix D. Note in particular that our sampling requirement is identical to that of (2.19). Recall from (2.21) that $T \ge 0$ is any bound on the pointwise value of b, and v_T is the pointwise T-truncation of the least squares solution v. Let the event C be as in (D.3), and write,

$$\mathbb{E}\|\widetilde{v}_T - b\|^2 = \Pr(C)\mathbb{E}[\|\widetilde{v}_T - b\|^2 \mid C] + \Pr(C^c)\mathbb{E}[\|\widetilde{v}_T - b\| \mid C^c].$$

Since $|b(\boldsymbol{y})| \leq T$, then $|\tilde{v}_T(\boldsymbol{y}) - b(\boldsymbol{y})| \leq 2T$, for any $\boldsymbol{y} \in I$, and we also have,

$$|\widetilde{v}_T(\boldsymbol{y}) - b(\boldsymbol{y})| \le |\widetilde{v}(\boldsymbol{y}) - b(\boldsymbol{y})|, \tag{E.1}$$

with probability 1. Thus by the law of total expectation,

$$\mathbb{E} \|\widetilde{v}_T - b\|^2 = \Pr(C)\mathbb{E}[\|\widetilde{v}_T - b\|^2 \mid C] + \Pr(C^c)\mathbb{E}[\|\widetilde{v}_T - b\|^2 \mid C^c]$$

$$\stackrel{(D.3)}{\leq} \Pr(C)\mathbb{E}[\|\widetilde{v}_T - b\|^2 \mid C] + 4\delta T^2$$

$$\stackrel{(E.1)}{\leq} \Pr(C)\mathbb{E}[\|\widetilde{v} - b\|^2 \mid C] + 4\delta T^2$$

$$\stackrel{(E.1)}{\leq} \mathbb{E}[\|\widetilde{v} - b\|^2 \mid C] + 4\delta T^2$$

$$\stackrel{(2.19),(D.3)}{\leq} (1 + \epsilon)\|v^* - b\|^2 + 4\delta T^2,$$

completing the proof.