The GaussianSketch for Almost Relative Error Kernel Distance

Jeff M. Phillips
School of Computing, University of Utah
jeffp@cs.utah.edu

Wai Ming Tai
School of Computing, University of Utah
wmtai@cs.utah.edu

Abstract

We introduce two versions of a new sketch for approximately embedding the Gaussian kernel into Euclidean inner product space. These work by truncating infinite expansions of the Gaussian kernel, and carefully invoking the RecursiveTensorSketch [Ahle et al. SODA 2020]. After providing concentration and approximation properties of these sketches, we use them to approximate the kernel distance between points sets. These sketches yield almost $(1+\varepsilon)$-relative error, but with a small additive α term. In the first variants the dependence on $1/\alpha$ is poly-logarithmic, but has higher degree of polynomial dependence on the original dimension d. In the second variant, the dependence on $1/\alpha$ is still poly-logarithmic, but the dependence on d is linear.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Kernel Distance, Kernel Density Estimation, Sketching

Category RANDOM

Acknowledgements We thank Rasmus Pagh for early conversations on this topic which helped reignite and motivate this line of thought. Part of the work was completed while the first author was visiting the Simons Institute for Theory of Computing.

1 Introduction

Kernel methods are a pillar of machine learning and general data analysis. These approaches consider classic problems such as PCA, linear regression, linear classification, k-means clustering which at their heart fit a linear subspace to a complex data set. Each of the methods can be solved by only inspecting the data via a dot product $\langle x, p\rangle$. Kernel methods, and specifically the "kernel trick," simply replaces these Euclidean dot products with a non-linear inner product operation. The two most common inner products are the polynomial kernel $K_z(x,p) = (\langle x,p\rangle + 1)^z$ and the Gaussian kernel $K(x,p) = \exp(-\|x-p\|^2)$.

The "magic" of the kernel method works mainly because of the existence of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ associated with any positive definite (p.d.) kernel [43] K. It is a function space, so for any data point $x \in \mathbb{R}^d$, there is a mapping $\phi : \mathbb{R}^d \to \mathcal{H}_K$ so $\phi(x) = K(x,\cdot)$. Since $\phi(x)$ is a function with domain $\mathbb{R}^d$, and each "coordinate" of $\phi(x)$ is associated with another point $p \in \mathbb{R}^d$, there are an infinite number of "coordinates," and $\mathcal{H}_K$ can be infinite dimensional. However, since $\langle \phi(x), \phi(p)\rangle_{\mathcal{H}_K} = K(x,p)$, this embedding does not ever need to be computed, we can simply evaluate $K(x,p)$. And life was good.

However, at the dawn of the age of big data, it became necessary to try to explicitly, but approximately, compute this map ϕ. Kernel methods typically start by computing and then analyzing the $n \times n$ gram matrix K_X where $(K_X)_{i,j} = K(x_i, x_j)$ for a data sets X of size n. As n became huge, this became untenable. In a hallmark paper, Rahimi and Recht [37] devised random Fourier features (RFFs) for p.d. kernels (with max value 1, e.g., Gaussians) that compute a random map $\tilde{\phi} : \mathbb{R}^d \to \mathbb{R}^{\tilde{D}}$ so $\langle \tilde{\phi}(x), \tilde{\phi}(p)\rangle$ is an unbiased estimate

of $K(x,p)$, and with probability at least $1-\delta$ has error $|K(x,p) - \langle\tilde{\phi}(x), \tilde{\phi}(p)\rangle| \leq \varepsilon$. For just one pair of points they require $\tilde{D} = O((1/\varepsilon^2)\log(1/\delta))$, or for all comparisons among n points $\tilde{D}_n = ((1/\varepsilon^2)\log(n/\delta))$, or for any points in a region Λ of volume $\mathsf{vol}(\Lambda) \leq V$, then $\tilde{D}_V = ((1/\varepsilon^2)\log(V/\delta))$.

However, relative-error-preserving RKHS embeddings for p.d. kernels are impossible without some restriction on the size n or domain Λ of the data. Consider n data points each far from each other so any pair $x, p \in \mathbb{R}^d$ satisfies $K(x,p) < 1/n$. In any relative-error-approximate embedding $\hat{\phi} : \mathbb{R}^d \to \mathbb{R}^{\hat{D}}$, each point must be virtually orthogonal to all other points, and hence $\Omega(n)$ dimensions are required [28].

Instead, to obtain (almost) relative-error results in big data sets, researchers have relied on other approaches such as sampling [45], exploiting structure of p.d. Gram matrices [34], devising modified RFFs for regularized kernel regression [9], or building data structures for kernel density estimate queries [12].

The kernel distance and data set embeddings. To address these difficulties, we first turn our attention from the inner product $\langle\phi(x), \phi(p)\rangle_{\mathcal{H}_K} = K(x,p)$ in the RKHS to the natural distance it implies. Before stating this distance, we generalize the inner product to point sets $P \subset \mathbb{R}^d$ (which extends naturally to probability distributions μ_P with domain $\mathbb{R}^d$). We treat P as a discrete probability distribution with uniform $1/|P|$ weight on each point. This can be represented in $\mathcal{H}_K$ as $\Phi(P) = \frac{1}{|P|}\sum_{x\in P}\phi(x)$, known as the kernel mean [33]. Indeed, for any query point $p \in \mathbb{R}^d$, the inner product $\langle\Phi(P), \phi(p)\rangle_{\mathcal{H}_K} = \frac{1}{|P|}\sum_{x\in P} K(x,p)$ is precisely the kernel density estimate at p. For two point sets $P, Q \subset \mathbb{R}^d$ we define $\kappa(P,Q) = \frac{1}{|P|}\frac{1}{|Q|}\sum_{x\in P}\sum_{y\in Q} K(p,q) = \langle\Phi(P), \Phi(Q)\rangle_{\mathcal{H}_K}$.

Now the kernel distance [36, 26] (alternatively known as the current distance [23] or maximum mean discrepancy [24, 39]) is defined

$$\mathtt{D}_K(P,Q) = \|\Phi(P) - \Phi(Q)\|_{\mathcal{H}_K} = \sqrt{\kappa(P,P) + \kappa(Q,Q) - 2\kappa(P,Q)}.$$

Under a slightly restricted class of kernels (a subset of p.d. kernels), called *characteristic* kernels [42], this distance is a metric. These include the Gaussian kernels which we focus on hereafter. This distance looks and largely acts like Euclidean distance; indeed, restricted to any finite-dimensional subspace, it is equivalent to Euclidean distance.

In data analysis and statistics, kernel mean is a compact way to represent a point set distribution. One can also use kernel distance to compare different point set as opposed to more expensive measure such as Wasserstein distance. In practice, there are various applications such as hypothesis test and geometric search (see section 4 for detail discussion) that use kernel distance as a core component. We suggest the reader refer to [38, 40] for more details on the statistical perspective of kernel distance. Therefore, making computation of the kernel distance scalable by a kernel distance embedding is of significant importance for those downstream applications. More generally, one can view oblivious kernel distance embedding as special case of oblivious subspace embedding for RKHS [32, 2], which gives a stronger guarantee than a subspace in the RKHS is preserved within relative error. However, many application of kernel distance do not require such a strong guarantee, which generally attain worse results (see below for more detail comparison).

So a natural question to ask is if this distance is preserved within relative error via some approximate lifting. Clearly RFFs guarantee additive ε-error. However, relate this problem to the Johnson-Lindenstrauss (JL) Lemma [25]: JL describe a family of random projections from a high-dimensional space to a D'-dimensional space which preserve $(1+\varepsilon)$-relative error

on Euclidean distance, again with $D' = O((1/\varepsilon^2)\log(n/\delta))$ for any $\binom{n}{2}$ pairs of distances, succeed with probability $1-\delta$, but only guarantees additive error on inner products.

Moreover, it is possible to apply the JL Lemma to create such an approximate embedding. First for any set of n points X, we can create $n \times n$ Gram matrix K_X (that is positive definite), and decompose it to $K_X = B_X B_X^T$. Then each row $(B_X)_i$ in B_X is the n-dimensional vector representation of the ith data point, and the Euclidean distance $\|(B_X)_i - (B_X)_j\|_2$ is the kernel distance between data points i and j [31, 8]. Then we can apply JL on these rows $\{(B_X)_i\}$ to obtain such an approximate embedding. However, this embedding is not oblivious to the data (necessary for many big data settings like streaming) and still requires $\Omega(n^2)$ time just to create the Gram matrix, not to mention the time for decomposition.

Another recent approach [14] analyzed RFFs for this task, and shows that these approximate embeddings do guarantee relative error on the kernel distance, but only between each pair of points $x, p \in \mathbb{R}^d$ (e.g., so $\frac{\|\hat{\phi}(x)-\hat{\phi}(p)\|}{\mathtt{D}_K(x,p)} \in (1 \pm \varepsilon)$), and as we describe next many downstream analysis tasks require the distance preserved between point *sets*. Alternatively, if we assume $\mathtt{D}_K^2(P,Q) > \alpha$, then standard RFFs can provide a relative error guarantee using $\tilde{D} = O(\frac{1}{\varepsilon^2\alpha^2}\log\frac{1}{\delta})$. However, such a large factor in α is undesirable, since typically $\alpha \ll \varepsilon$.

Our Results. We provide two sketches $G : \mathbb{R}^d \to \mathbb{R}^D$ for the Gaussian kernel, improving on work of Rahimi and Recht [37] and Avron *et al.* [9], which achieves almost relative error for kernel distance. Let $F(X) = \frac{1}{|X|}\sum_{x\in X} G(x)$ extend the sketch to point sets $X \subset \mathbb{R}^d$. Then we show that for two point sets $P, Q \subset \mathbb{R}^d$

$$\left|\mathtt{D}_K^2(P,Q) - \|F(P) - F(Q)\|^2\right| \leq \varepsilon \mathtt{D}_K^2(P,Q) + \alpha.$$

As we can always reduce the dimension $G : \mathbb{R}^d \to \mathbb{R}^D$ using JL to about $D = 1/\varepsilon^2$, we focus on reducing the runtime dependence, in particular the dependence on α.

In the first sketch (the GaussianSketch) to process a single point with $G(x)$ it takes $O\left(\frac{d^2}{\varepsilon^2}\log\frac{d}{\varepsilon} + ds\right)$ time, with $s = \Theta\left(\frac{\log(d\exp(dL^2)/\alpha)}{\log(\frac{1}{L^2}\log(d\exp(dL^2)/\alpha))}\right)$, where L describes the (L_∞) radius of the domain containing X. So the dependence on $1/\alpha$ is less than a single logarithmic term.

The second sketch (the GaussianSketchHD) is useful when the dimension d is potentially large (it turns out to be very similar to a recent sketch in [2], but our analysis is different). Then the runtime to compute $G(x)$ is $O\left(\frac{s^3}{\varepsilon^2}\log\frac{s}{\varepsilon} + s^2 d\right)$ where $s = \Theta\left(\frac{\log(4\exp(2R^2)/\alpha)}{\log(\frac{1}{R^2}\log(4\exp(2R^2)/\alpha))}\right)$, and R is the (L_2) domain radius. Now the dependence on $1/\alpha$ is still poly-logarithmic, but the dependence on dimension d is linear.

For example, we can set $\alpha = n^{-C_1}$, $R = C_2\sqrt{\log n}$ and $L = C_3\sqrt{\log n}$ for some absolute constant C_1, C_2, C_3. In low dimension, we have $s = \Theta(\frac{\log n}{\log d})$ and the running time is $O(\frac{d^2}{\varepsilon^2}\log\frac{d}{\varepsilon} + \frac{d\log n}{\log d})$. In high dimension, we have $s = \Theta(\log n)$ and the running time is $O\left(\frac{1}{\varepsilon^2}\log^3 n\log(\log n/\varepsilon) + d\log^2 n\right)$.

Implications. Several concrete applications work directly on this kernel distance between point sets. First, the kernel two-sample test [24, 33] is a non-parametric way to perform hypothesis tests between two empirical distributions; simply, the null hypothesis of them being drawn from the same distribution is rejected if the kernel distance is sufficiently large. While the sketched kernel two-sample test has proven effective under additive error [48], when the significance threshold is $\Theta(1/n)$, the RFF-based solutions require time $O(n^2)$, no better than brute force; but setting ε constant and $\alpha = 1/n$, our sketches provide near-linear or almost-linear time runtimes. Second, devising a Locality Sensitive Hash (LSH)

between point sets (or geometrically-aware LSH for probability distributions) has lacked a great general solution. Despite progress in special cases (e.g., for polygons [13], curves [18]), more general distances between geometric distributions, like Earth-Mover distance require $\Omega(\log s)$ distortion on a domain with at least s discrete points [7]. In general, an LSH requires relative error to properly provide $(1+\varepsilon)$-approximate nearest neighbor results. In Section 4 we specify how our new almost relative-error embeddings for the kernel distance provide efficient solutions for these applications.

Furthermore, this embedding can be composed with a Johnson-Lindenstrauss-type embedding [25, 3, 4, 1, 46] to create an overall oblivious embedding of dimension roughly $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$, that is with no dependence on $1/\alpha$ or d (or n or domain radius L or R in the *for each* setting), and roughly the same guarantees.

1.1 Comparison to Other Recent Work on Large Data and Kernels

Recent related works on kernel approximation do not provide our guarantees; we survey here work that addresses similar problems, and often require similar sets of error parameters.

Approximated KDEs. Charikar and Siminelakis [12] describe a data structure of size $n\hat{D}$ and query time $\hat{D}$, which answers $\kappa(P,t)$ queries within $(1+\varepsilon)$-relative error as long as $\kappa(P,t)>\alpha$; it requires $\hat{D}=O(\frac{1}{\varepsilon^2}\frac{1}{\sqrt{\alpha}}\log\frac{1}{\delta}e^{O(\log^{2/3} n\log\log n)})$. However, this cannot argue much about how large $\mathtt{D}_K(P,Q)$ has to be for this to achieve relative error on the kernel distance since it could be $\mathtt{D}_K(P,Q)$ is small but $\kappa(P,t)$ and $\kappa(P,P)$ are both large. Moreover, its guarantees only work for a single point set P with point queries t, not for two or more points sets P,Q, as we argue many downstream data analysis tasks require.

Approximated kernel regression. Avron *et al.* [9] modify the RFF embeddings using different sampling probability related to the statistical leverage in the kernel space. This approximates a λ-regularized kernel regression problem, creating a $\tilde{D}$-dimensional embedding; that is for an $n\times n$ gram matrix K_X, and a regularization parameter λ it creates a $n\times\tilde{D}$ matrix Z so $(1-\varepsilon)(K_X+\lambda I_n)\preceq ZZ^*+\lambda I_n\preceq(1+\varepsilon)(K_X+\lambda I_n)$, using $\tilde{D}=O(\frac{1}{\varepsilon^2}(L^d\log^{d/2}(n/\lambda)+\log^{2d}(n/\lambda))\log(s_\lambda(K)/\delta))$. Following our forthcoming methods for analysis, one can modify this result to $(1+\varepsilon)$-approximate the kernel distance, with an additive α term, with an embedding of dimension $D=O\left(\frac{1}{\varepsilon^2}(L^d\log^{d/2}\frac{n}{\alpha}+\log^{2d}\frac{n}{\alpha})\log\frac{n}{\delta}\right)$.

Also, Ahle *et al.* [2] recently showed that one can create such $\tilde{D}$-dimensional embedding where $\tilde{D}=O(\frac{1}{\varepsilon^2}(R^2+\log\frac{n}{\varepsilon\lambda})^5 s_\lambda(K_X))$ in $O(\frac{1}{\varepsilon^2}(R^2+\log\frac{n}{\varepsilon\lambda})^6 s_\lambda(K_X))$ time for each data point. Again, in our setting, one can interpret this result as $(1+\varepsilon)$-approximate the kernel distance, with an additive α term, in $O(\frac{1}{\varepsilon^2}(R^2+\log\frac{n}{\varepsilon\alpha})^6 s_\alpha(K_X))$ time.

Compared to our bounds (adapted to our problem using our techniques), these depend on n and s_λ (ours do not), the low-d one is exponential in d (ours is polynomial), and the other powers are larger.

Approximate Kernel PCA. Suppose we are given a data set $X=\{x_1,\ldots,x_n\}\subset\mathbb{R}^d$, and want to find a low rank (rank k) approximation of $X_\phi=\{\phi(x_1),\phi(x_2),\ldots,\phi(x_n)\}\in\mathcal{H}_K$. In particular, this can be described concretely in the context of the Gram matrix K_X and its decomposition $B_X B_X^T$. Given any $n\times m$ matrix M, let $[M]_k$ be its best rank-k approximation. A natural question is to find a rank-k matrix $\tilde{K}_X$ so

$$\left\|K_X-\tilde{K}_X\right\|_F^2\leq(1+\varepsilon)\left\|K_X-[K_X]_k\right\|_F^2.$$

While most previous work [19, 30, 22, 41, 44] has focused on providing absolute (or additive) error bounds. For instance, they showed roughly $\|K_X - \tilde{K}_X\|_F^2 \leq \|K_X - [K_X]_k\|_F^2 + \varepsilon n$ using e.g., Nyström sampling and RFFs. More recently, Musco and Woodruff [35] for p.d. Gram matrices K_X show how to efficiently find $\tilde{K}_X$ with relative error. This only requires $O(nk^{\omega-1} \cdot \mathsf{poly}(\log n/\varepsilon))$ inspections of entries of K_X, where $\omega < 2.373$ is the matrix multiplication exponent. This is not data oblivious, and uses properties of the p.d. matrix, so it does not provide an embedding sketch.

A closely related problem is approximate kernel PCA problem which is to find a $n \times k$ orthonormal matrix V so that

$$\|B_X - VV^T B_X\|_F^2 \leq (1+\varepsilon)\|B_X - [B_X]_k\|_F^2.$$

The RKHS basis V, provides a compact and non-linear set of attributes to describe a complex data set X, and has many uses in analyzing complex data which lacks strong linear correlations. Musco and Woodruff [34] provide an algorithm with runtime $O(\mathsf{nnz}(X)) + \tilde{O}(n^{\omega+1.5}(\frac{k}{\sigma_{k+1}\varepsilon^2})^{\omega-1.5})$; which has polynomial dependence on $1/\sigma_{k+1}$. They leave open whether this can be removed or reduced while maintaining only roughly $\mathsf{nnz}(X)$ dependence on X. The matrix V returned by their algorithm can be used to approximate the matrix K_X by writing $B_X P B_X^T$ where P is the projection onto the row span of $VV^T B_X$.

Our techniques can be combined with the a sketch for the polynomial kernel [10] to explicitly solve for V so

$$\|B_X - VV^T B_X\|_F^2 \leq (1+\varepsilon)\|B_X - [B_X]_k\|_F^2 + \alpha.$$

with similar dimensions required for approximating the kernel distance; the s parameter increases roughly by $\log n/\log\log n$. This is detailed in Appendix A. If the data size n has a known bound, then this provides an oblivious sketch for this almost relative error kernel PCA problem. Moreover, replacing the σ_{k+1} with $\varepsilon\alpha$, it almost answers the kernel PCA $\mathsf{nnz}(X)$ question of Musco and Woodruff [34] – however our algorithm does not depend on the number-of-non-zeros of X through our sketches, so we leave as an open question if our sketches $G(x)$, particular the GaussianSketchHD or similar, can be generated in time $O(\mathsf{nnz}(x)\mathsf{polylog}(1/\alpha) + n\mathsf{poly}(k, 1/\varepsilon, \log(1/\alpha))$.

2 The GaussianSketch and its Properties

In this section we describe our new sketches for approximate mapping from $\mathbb{R}^d$ to an RKHS associated with a Gaussian kernel. They are based on the RecursiveTensorSketch of Ahle *et al.* [2], so we first review its properties.

The RecursiveTensorSketch. We first introduce RecursiveTensorSketch hash family [2]. Given positive integers n, m and k, RecursiveTensorSketch$_{n,m,k}$ is the family of hash functions $T : \mathbb{R}^{n^k} \to \mathbb{R}^m$ as constructed in [2]. This hash family will be used to construct our main sketch and has the following guarantee [2]: suppose $u, v \in \mathbb{R}^{n^k}$ and picking $m = O(\frac{k}{\varepsilon^2})$, then the expectation $\mathsf{E}(\langle T(u), T(v)\rangle) = \langle u, v\rangle$ and the variance $\mathsf{Var}(\langle T(u), T(v)\rangle) \leq \frac{\varepsilon^2}{10}\|u\|^2\|v\|^2$. Moreover, the running time of computing $T(x)$ for any $x \in \mathbb{R}^{n^k}$ is $O(km\log m + kn)$.

The GaussianSketch. Now, we can define the hash family of the first sketch for the Gaussian kernel GaussianSketch. Given a vector $x \in \mathbb{R}^d$ and a positive integer s, we first define

d vectors $y_x^{(1)} \ldots, y_x^{(d)} \in \mathbb{R}^s$ such that ith coordinate of $y_x^{(j)}$ is $\exp(-x_j^2)\sqrt{\frac{2^{i-1}}{(i-1)!}}x_j^{i-1}$. Given an integer m, define GaussianSketch$_{m,s}$ to be the family of hash functions that if G is in it, then $G(x) = T(y_x^{(1)} \otimes \cdots \otimes y_x^{(d)})$ where T is randomly chosen from RecursiveTensorSketch$_{s,m,d}$.

Here, $x \otimes y$ is Kronecker product. Namely, given $x \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$, $x \otimes y$ is a pq dimensional vector indexed by two integers i, j where $i = 1, \ldots, p$ and $j = 1, \ldots, q$ such that $(x \otimes y)_{i,j} = x_i \cdot y_j$. For notational convenience, we extend Kronecker product when p and q are infinity. Namely, given $\{x_i\}_{i=1}^{\infty}$ and $\{y_j\}_{j=1}^{\infty}$ are infinite sequences, $x \otimes y$ is also an infinite sequence indexed by two positive integers i, j such that $(x \otimes y)_{i,j} = x_i \cdot y_j$. Also, denote $x^{\otimes k} = x \otimes x^{\otimes k-1}$ and $x^{\otimes 0} = 1$.

The rationale for the GaussianSketch comes from the following infinite expansion of the Gaussian kernel. Define $\bar{y}_x^{(j)}$ (for $j \in [d]$) as the infinite dimensional analog of $y_x^{(j)}$ with its ith coordinate as $\exp(-x_j^2)\sqrt{\frac{2^{i-1}}{(i-1)!}}x_j^{i-1}$.

▶ **Lemma 1.** *For $x, p \in \mathbb{R}^d$*

$$\begin{aligned}
&\exp(-\|x-p\|^2) \\
&= \sum_{j_1=0}^{\infty} \cdots \sum_{j_d=0}^{\infty} \left(\exp(-\|x\|^2)\left(\prod_{i=1}^{d}\sqrt{\frac{2^{j_i}}{j_i!}}x_i^{j_i}\right)\right)\left(\exp(-\|p\|^2)\left(\prod_{i=1}^{d}\sqrt{\frac{2^{j_i}}{j_i!}}p_i^{j_i}\right)\right) \\
&= \left\langle \bar{y}_x^{(1)} \otimes \cdots \otimes \bar{y}_x^{(d)}, \bar{y}_p^{(1)} \otimes \cdots \otimes \bar{y}_p^{(d)} \right\rangle.
\end{aligned}$$

Proof.

$$\begin{aligned}
&\exp(-\|x-p\|^2) \\
&= \exp(-\|x\|^2)\exp(-\|p\|^2)\exp(2\langle x, p\rangle) \\
&= \exp(-\|x\|^2)\exp(-\|p\|^2)\prod_{i=1}^{d}\exp(2x_ip_i) \\
&= \exp(-\|x\|^2)\exp(-\|p\|^2)\prod_{i=1}^{d}\left(\sum_{j=0}^{\infty}\frac{1}{j!}(2x_ip_i)^j\right) \quad \text{by Taylor expansion of } \exp(\cdot) \\
&= \exp(-\|x\|^2)\exp(-\|p\|^2)\sum_{j_1=0}^{\infty}\cdots\sum_{j_d=0}^{\infty}\left(\prod_{i=1}^{d}\frac{1}{j_i!}(2x_ip_i)^{j_i}\right) \\
&= \sum_{j_1=0}^{\infty} \cdots \sum_{j_d=0}^{\infty} \left(\exp(-\|x\|^2)\left(\prod_{i=1}^{d}\sqrt{\frac{2^{j_i}}{j_i!}}x_i^{j_i}\right)\right)\left(\exp(-\|p\|^2)\left(\prod_{i=1}^{d}\sqrt{\frac{2^{j_i}}{j_i!}}p_i^{j_i}\right)\right) \\
&= \left\langle \bar{y}_x^{(1)} \otimes \cdots \otimes \bar{y}_x^{(d)}, \bar{y}_p^{(1)} \otimes \cdots \otimes \bar{y}_p^{(d)} \right\rangle.
\end{aligned}$$

◀

Note that the Gaussian sketch takes as input one element of these inner products, but trimmed so that each $\bar{y}_x^{(j)}$ is trimmed to $y_x^{(j)}$ (without the $\bar{}$ marker) that only has s terms each.

The GaussianSketchHD. We can also define another hash family of sketches for the Gaussian kernel GaussianSketchHD, which works better for high dimension d, but will have worse dependence on other error and domain parameters. For $j = 1, \ldots, s$, it will use T_j as randomly chosen from RecursiveTensorSketch$_{d,m_j,j-1}$. Given a vector $x \in \mathbb{R}^d$, a positive integer s, and s positive integers $m_1, \ldots, m_s$, define GaussianSketchHD$_{m_1,\ldots,m_s,s}$

to be the family of hash functions that if G is in it, then $G(x) \in \mathbb{R}^m$ with $(m_{j-1}+1)$th coordinate to m_jth coordinate be $\sqrt{\frac{2^{j-1}}{(j-1)!}}\exp(-\|x\|^2)T_j(x^{\otimes j-1}) = T_j(z_x^{(j)}) \in \mathbb{R}^{m_j}$ where $z_x^{(j)} = \sqrt{\frac{2^{j-1}}{(j-1)!}}\exp(-\|x\|^2)x^{\otimes j-1} \in \mathbb{R}^{d^{j-1}}$ and $m = \sum_{j=1}^{s} m_j$. Denote z_x the $\frac{d^s-1}{d-1}$ dimensional vector where the first coordinate is $z_x^{(1)}$, the next d coordinates are $z_x^{(2)}$, the next d^2 coordinates are $z_x^{(3)}$, and so on. The GaussianSketchHD uses the following, a different infinite expansion of the Gaussian kernel (also explored by Cotter *et al.* [17]).

▶ **Lemma 2.** *For $x, p \in \mathbb{R}^d$,*

$$\exp(-\|x-p\|^2) = \sum_{i=0}^{\infty}\left\langle \exp(-\|x\|^2)\sqrt{\frac{2^i}{i!}}x^{\otimes i}, \exp(-\|p\|^2)\sqrt{\frac{2^i}{i!}}p^{\otimes i}\right\rangle = \sum_{i=0}^{\infty}\left\langle z_x^{(i)}, z_p^{(i)}\right\rangle$$

Proof.

$$\begin{aligned}
&\exp(-\|x-p\|^2)\\
&= \exp(-\|x\|^2)\exp(-\|p\|^2)\exp(2\langle x,p\rangle)\\
&= \exp(-\|x\|^2)\exp(-\|p\|^2)\sum_{i=0}^{\infty}\frac{1}{j!}(2\langle x,p\rangle)^j && \text{by Taylor expansion of } \exp(\cdot)\\
&= \exp(-\|x\|^2)\exp(-\|p\|^2)\sum_{i=0}^{\infty}\frac{2^j}{j!}\langle x^{\otimes j}, p^{\otimes j}\rangle\\
&= \sum_{j=0}^{\infty}\left\langle \exp(-\|x\|^2)\sqrt{\frac{2^j}{j!}}x^{\otimes j}, \exp(-\|p\|^2)\sqrt{\frac{2^i}{j!}}p^{\otimes j}\right\rangle
\end{aligned}$$

◀

2.1 Concentration Bounds for GaussianSketch and GaussianSketchHD

The sketches will inherit the concentration properties of the RecursiveTensorSketch. Similar observations were recently observed by Ahle *et al.* [2]. Consider a weighted set of elements $X \subset \mathbb{R}^d$ with weights α_x for $x \in X$, and we use the general concentration bounds for these under the GaussianSketch.

▶ **Lemma 3** ([2]). *Let G be a randomly chosen hash function in* GaussianSketch$_{m,s}$ *with $m = O\left(\frac{d}{\varepsilon^2}\right)$. Let $v = \sum_{x\in X}\alpha_x y_x^{(1)} \otimes \cdots \otimes y_x^{(d)}$, then $\mathsf{E}\left[\left\|\sum_{x\in X}\alpha_x G(x)\right\|^2\right] = \|v\|^2$ and $\mathsf{Var}\left[\left\|\sum_{x\in X}\alpha_x G(x)\right\|^2\right] \leq \frac{\varepsilon^2}{10}\|v\|^4$ and hence with probability at least $9/10$ we have $\left|\left\|\sum_{x\in X}\alpha_x G(x)\right\|^2 - \|v\|^2\right| \leq \varepsilon\|v\|^2$.*

If G is randomly chosen from GaussianSketchHD$_{m_1,\ldots,m_s,s}$, then $G(x) = Sz_x$, where S is a $m \times \frac{d^s-1}{d-1}$ random matrix (recall $m = \sum_{j=1}^{s} m_j$) so, for the $(m_{i-1}+1)$th row to the m_ith row, and the $(\frac{d^{i-1}-1}{d-1}+1)$th column to the $\frac{d^i-1}{d-1}$th column forms a matrix S_i where $T_i(z_x^{(i)}) = S_i z_x^{(i)}$, and the rest of entries are zero.

▶ **Lemma 4** ([2]). *Suppose A, B has $\frac{d^s-1}{d-1}$ columns. Denote A_i and B_i be ith row of A and B respectively. By taking $m_i = O\left(\frac{i}{\varepsilon^2}\right)$, we have $\mathbf{Pr}\left[\left\|AB^T - AS^TSB^T\right\|_F^2 \leq \varepsilon^2\|A\|_F^2\|B\|_F^2\right] \geq 1-\delta$.*

2.2 Truncation Bounds for GaussianSketch and GaussianSketchHD

These sketches are effective when it is useful to analyze the effect of sketching a large data set X of size n, and we desire to show the cumulative measured across all pairs of elements. For each sketch we expand these infinite sums, and determine the truncation parameter s so the sum of terms past s have a bounded effect.

In our analysis, we will use the following inequality which follows by standard calculus analysis, for any $\eta > 0$,

$$\sum_{j=s}^{\infty} \frac{\eta^j}{j!} \leq \frac{\left(\sup_{y\in[-\eta,\eta]} \exp(y)\right) \eta^s}{s!} \leq \frac{\exp(\eta)\eta^s}{s!} \tag{1}$$

The following expression also arises in our analysis.

▶ **Lemma 5.** *For $\xi, a, b > 0$, setting $s = \Theta\left(\frac{\log \frac{\xi\cdot a}{\alpha}}{\log\left(\frac{1}{b}\log\frac{\xi\cdot a}{\alpha}\right)}\right)$ then the we have $\xi \cdot a \left(\frac{b}{s}\right)^s \leq \alpha$.*

Proof. By setting $\frac{s}{b} = C\frac{\gamma}{\log\gamma}$ for some large constant C where $\gamma = \frac{1}{b}\log\frac{\xi a}{\alpha}$, we have

$$\frac{s}{b}\log\frac{s}{b} = C\frac{\gamma}{\log\gamma}\log\left(C\frac{\gamma}{\log\gamma}\right) = \gamma\cdot C\left(1 + \frac{\log C}{\log\gamma} - \frac{\log\log\gamma}{\log\gamma}\right) \geq \gamma = \frac{1}{b}\log\frac{\xi a}{\alpha}.$$

Now, if we rearrange the inequality then $\xi \cdot a \left(\frac{b}{s}\right)^s \leq \alpha$. ◀

Consider a point set $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\} \subset \mathbb{R}^d$, denote K_X as the $n \times n$ matrix with $(K_X)_{i,j} = \exp(-\left\|x^{(i)} - x^{(j)}\right\|^2)$. First truncate K_X using Lemma 1 to obtain the $n\times n$ matrix $K^{\mathsf{GS}}_{X,s}$ with

$$\begin{aligned}
&(K^{\mathsf{GS}}_{X,s})_{i,j} \\
&= \sum_{j_1=0}^{s-1}\cdots\sum_{j_d=0}^{s-1}\left(\exp(-\left\|x^{(i)}\right\|^2)\left(\prod_{a=1}^{d}\sqrt{\frac{2^{j_a}}{j_a!}}(x_a^{(i)})^{j_a}\right)\right) \\
&\qquad\qquad \cdot\left(\exp(-\left\|x^{(j)}\right\|^2)\left(\prod_{a=1}^{d}\sqrt{\frac{2^{j_a}}{j_a!}}(p_a^{(j)})^{j_a}\right)\right)
\end{aligned}$$

▶ **Lemma 6.** *Suppose $X \subset \mathbb{R}^d$ so for all $x^{(i)} \in X$ has $\left\|x^{(i)}\right\|_\infty \leq L$ for some $L > 0$. Given a vector $w \in \mathbb{R}^n$ with $\left(\sum_{i=1}^n |w_i|\right)^2 \leq \xi$, we have*

$$w^T(K_X - K^{\mathsf{GS}}_{X,s})w \leq \left(\sum_{i=1}^n |w_i|\right)^2 d\exp(2dL^2)\left(\frac{2eL^2}{s}\right)^s \leq \alpha,$$

where the last $\leq \alpha$ inequality follows from setting $s = s_{L,d,\alpha} = \Theta\left(\frac{\log\frac{\xi\cdot d\exp(2dL^2)}{\alpha}}{\log\left(\frac{1}{2eL^2}\log\frac{\xi\cdot d\exp(2dL^2)}{\alpha}\right)}\right)$.

Proof. From Lemma 1, we have

$$\begin{aligned}
&(K_X - K^{\mathsf{GS}}_{X,s})_{i,j} \\
&= \sum_{\substack{j_1,\ldots,j_d \\ \text{one of } j_b \geq s}}\left(\exp(-\left\|x^{(i)}\right\|^2)\left(\prod_{a=1}^{d}\sqrt{\frac{2^{j_a}}{j_a!}}(x_a^{(i)})^{j_a}\right)\right) \\
&\qquad\qquad \cdot\left(\exp(-\left\|x^{(j)}\right\|^2)\left(\prod_{a=1}^{d}\sqrt{\frac{2^{j_a}}{j_a!}}(x_a^{(j)})^{j_a}\right)\right)
\end{aligned}$$

Then we can analyze these in aggregate with respect to a test vector z. The first line uses the fact that a matrix A (for instance with $A = K_X - K_{X,s}^{\mathsf{GS}}$) written as $\sum_j (\sum_{x_i \in X} \psi_j(x_i))(\sum_{x_i' \in X} \psi_j(x_i'))$ can be simplified $w^T A w = \sum_j (\sum_{x_i \in X} w_i \psi_j(x_i))^2$.

$$
\begin{aligned}
& w^T (K_X - K_{X,s}^{\mathsf{GS}}) w \\
&= \sum_{\substack{j_1,\ldots,j_d \\ \text{one of } j_b \geq s}} \left(\sum_{i=1}^n w_i \exp(-\left\|x^{(i)}\right\|^2) \left(\prod_{a=1}^d \sqrt{\frac{2^{j_a}}{j_a!}} (x_a^{(i)})^{j_a} \right) \right)^2 \\
&\leq \sum_{b=1}^d \sum_{\substack{j_1,\ldots,j_d \\ j_b \geq s}} \left(\sum_{i=1}^n w_i \exp(-\left\|x^{(i)}\right\|^2) \left(\prod_{a=1}^d \sqrt{\frac{2^{j_a}}{j_a!}} (x_a^{(i)})^{j_a} \right) \right)^2 && \text{by union bound} \\
&\leq \sum_{b=1}^d \sum_{\substack{j_1,\ldots,j_d \\ j_b \geq s}} \left(\sum_{i=1}^n |w_i| \left(\prod_{a=1}^d \sqrt{\frac{2^{j_a}}{j_a!}} L^{j_a} \right) \right)^2 && \text{assuming } \left\|x^{(i)}\right\|_\infty \leq L \\
&\leq \left(\sum_{i=1}^n |w_i| \right)^2 \left(\sum_{b=1}^d \sum_{\substack{j_1,\ldots,j_d \\ j_b \geq s}} \left(\prod_{a=1}^d \frac{(2L^2)^{j_a}}{j_a!} \right) \right)
\end{aligned}
$$

The term $\sum_{b=1}^d \sum_{\substack{j_1,\ldots,j_d \\ j_b \geq s}} \left(\prod_{a=1}^d \frac{(2L^2)^{j_a}}{j_a!} \right)$ can be expressed as the follows.

$$
\begin{aligned}
& \sum_{b=1}^d \sum_{\substack{j_1,\ldots,j_d \\ j_b \geq s}} \left(\prod_{a=1}^d \frac{(2L^2)^{j_a}}{j_a!} \right) \\
&= \sum_{b=1}^d \left(\sum_{j_1=0}^\infty \frac{(2L^2)^{j_1}}{j_1!} \right) \cdots \left(\sum_{j_b=s}^\infty \frac{(2L^2)^{j_b}}{j_b!} \right) \cdots \left(\sum_{j_d=0}^\infty \frac{(2L^2)^{j_d}}{j_d!} \right) \\
&= \sum_{b=1}^d \left(\prod_{\substack{a=1 \\ a \neq b}}^d \exp(2L^2) \right) \left(\sum_{j_b=s}^\infty \frac{(2L^2)^{j_b}}{j_b!} \right) \\
&\leq \sum_{b=1}^d \left(\exp((d-1)2L^2) \right) \frac{\exp(2L^2)(2L^2)^s}{s!} && \text{by (1)} \\
&= \frac{d \exp(2dL^2)(2L^2)^s}{s!} \\
&\leq d \exp(2dL^2) \left(\frac{2eL^2}{s} \right)^s && \text{by the fact } s! \geq \left(\frac{s}{e} \right)^s
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
w^T (K_X - K_{X,s}^{\mathsf{GS}}) w &\leq \left(\sum_{i=1}^n |w_i| \right)^2 \left(\sum_{b=1}^d \sum_{\substack{j_1,\ldots,j_d \\ j_b \geq s}} \left(\prod_{a=1}^d \frac{(2L^2)^{j_a}}{j_a!} \right) \right) \\
&\leq \left(\sum_{i=1}^n |w_i| \right)^2 d \exp(2dL^2) \left(\frac{2eL^2}{s} \right)^s \\
&\leq \alpha
\end{aligned}
$$

where the last inequality follows Lemma 5 using $\xi = (\sum_{i=1}^{n} |w_i|)^2$, $a = d\exp(2dL^2)$ and $b = 2eL^2$. ◀

Now truncate K_X based on Lemma 2 to obtain $K^{\mathsf{HD}}_{X,s}$ with

$$(K^{\mathsf{HD}}_{X,s})_{i,j} = \sum_{a=0}^{s-1} \left\langle \exp(-\left\|x^{(i)}\right\|^2)\sqrt{\frac{2^a}{a!}}(x^{(i)})^{\otimes a}, \exp(-\left\|x^{(j)}\right\|^2)\sqrt{\frac{2^a}{a!}}(x^{(j)})^{\otimes a} \right\rangle$$

▶ **Lemma 7.** *Define $\Lambda^d_R = \{x \in \mathbb{R}^d \mid \|x\|_2 \leq R\}$. For a point set $X \subset \Lambda^d_R$, and a vector $w \in \mathbb{R}^n$ with $(\sum_{i=1}^{n} |w_i|)^2 \leq \xi$, we have*

$$w^T(K_X - K^{\mathsf{HD}}_{X,s})w \leq \left(\sum_{i=1}^{n} |w_i|\right)^2 \exp(2R^2)\left(\frac{2eR^2}{s}\right)^s \leq \alpha$$

where the last $\leq \alpha$ inequality follows from setting $s = s_{R,\alpha} = \Theta\left(\frac{\log \frac{\xi \cdot \exp(2R^2)}{\alpha}}{\log\left(\frac{1}{2eR^2}\log \frac{\xi \cdot \exp(2R^2)}{\alpha}\right)}\right)$.

Proof. From Lemma 2, we have

$$(K_X - K^{\mathsf{HD}}_{X,s})_{i,j} = \sum_{a=s}^{\infty} \left\langle \exp(-\left\|p^{(i)}\right\|^2)\sqrt{\frac{2^a}{a!}}(p^{(i)})^{\otimes a}, \exp(-\left\|p^{(j)}\right\|^2)\sqrt{\frac{2^a}{a!}}(p^{(j)})^{\otimes a} \right\rangle$$

Then we can analyze these in aggregate with respect to a test vector z. The first line uses the fact that a matrix A (for instance with $A = K_X - K^{\mathsf{HD}}_{X,s}$) written as $\sum_j(\sum_{x_i \in X} \psi_j(x_i))(\sum_{x'_i \in X} \psi_j(x'_i))$ can be simplified $w^T A w = \sum_j(\sum_{x_i \in X} w_i \psi_j(x_i))^2$.

$$\begin{aligned}
&w^T(K_X - K^{\mathsf{HD}}_{X,s})w \\
&= \sum_{a=s}^{\infty} \left\| \sum_{i=1}^{n} w_i \exp(-\left\|x^{(i)}\right\|^2)\sqrt{\frac{2^a}{a!}}(x^{(i)})^{\otimes a} \right\|^2 \\
&\leq \sum_{a=s}^{\infty} \left(\sum_{i=1}^{n} |w_i| \left\| \exp(-\left\|x^{(i)}\right\|^2)\sqrt{\frac{2^a}{a!}}(x^{(i)})^{\otimes a} \right\| \right)^2 \\
&\leq \sum_{a=s}^{\infty} \left(\sum_{i=1}^{n} |w_i| \sqrt{\frac{2^a}{a!}} R^a \right)^2 && \text{assuming } \left\|x^{(i)}\right\| \leq R \\
&= \left(\sum_{i=1}^{n} |w_i|\right)^2 \left(\sum_{a=s}^{\infty} \frac{(2R^2)^a}{a!}\right) \\
&\leq \left(\sum_{i=1}^{n} |w_i|\right)^2 \frac{\exp(2R^2)(2R^2)^s}{s!} && \text{by (1)} \\
&\leq \left(\sum_{i=1}^{n} |w_i|\right)^2 \exp(2R^2)\left(\frac{2eR^2}{s}\right)^s && \text{by the fact } s! \geq \left(\frac{s}{e}\right)^s \\
&\leq \alpha
\end{aligned}$$

where the last inequality follows Lemma 5 using $\xi = (\sum_{i=1}^{n} |w_i|)^2$, $a = \exp(2R^2)$ and $b = 2eR^2$. ◀

3 Application to the Gaussian Kernel Distance

Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be Gaussian kernel. Namely, for any $x, y \in \mathbb{R}^d$, $K(x,y) = \exp(-\|x-y\|^2)$. Given two point sets $P, Q \subset \mathbb{R}^d$, one can define a similarity function $\kappa(P,Q) = \frac{1}{|P|}\frac{1}{|Q|}\sum_{x\in P}\sum_{y\in Q} K(x,y)$ and a squared kernel distance

$$\mathtt{D}_K^2(P,Q) = \kappa(P,P) - 2\kappa(P,Q) + \kappa(Q,Q).$$

We make the important observation that the above formulation is equivalent to the following form which will be much simpler to fit within our framework:

$$\mathtt{D}_K^2(P,Q) = \sum_{x\in P\cup Q}\sum_{y\in P\cup Q} \beta_x\beta_y \exp(-\|x-y\|^2)$$

where β_x is $\frac{1}{|P|}$ if $x \in P$ and $-\frac{1}{|Q|}$ if $x \in Q$.

We now express $\mathtt{D}_K^2(P,Q)$ as the infinite sum using Lemma 1.

$$\begin{aligned}
&\mathtt{D}_K^2(P,Q)\\
&= \sum_{x\in P\cup Q}\sum_{y\in P\cup Q} \beta_x\beta_y \exp(-\|x-y\|^2)\\
&= \sum_{x\in P\cup Q}\sum_{y\in P\cup Q} \beta_x\beta_y \sum_{j_1=0}^{\infty}\cdots\sum_{j_d=0}^{\infty}\left(\exp(-\|x\|^2)\left(\prod_{i=1}^{d}\sqrt{\frac{2^{j_i}}{j_i!}}x_i^{j_i}\right)\right)\\
&\qquad\qquad\cdot\left(\exp(-\|y\|^2)\left(\prod_{i=1}^{d}\sqrt{\frac{2^{j_i}}{j_i!}}y_i^{j_i}\right)\right)\\
&= \sum_{j_1=0}^{\infty}\cdots\sum_{j_d=0}^{\infty}\left(\sum_{x\in P\cup Q}\beta_x\exp(-\|x\|^2)\left(\prod_{i=1}^{d}\sqrt{\frac{2^{j_i}}{j_i!}}x_i^{j_i}\right)\right)^2\\
&= \left\|\sum_{x\in P\cup Q}\beta_x\bar{y}_x^{(1)}\otimes\cdots\otimes\bar{y}_x^{(d)}\right\|^2,
\end{aligned}$$

where each $\bar{y}_x^{(j)}$ is an infinite dimension vector with ith coordinate $\exp(-x_j^2)\sqrt{\frac{2^{i-1}}{(i-1)!}}x_j^{i-1}$.

▶ **Theorem 8.** *For any $\varepsilon, R, \alpha > 0$, let G be randomly chosen from* GaussianSketch$_{m,s}$ *with $m = O\left(\frac{d}{\varepsilon^2}\right)$ and $s = \Theta\left(\frac{\log\frac{4d\exp(2dL^2)}{\alpha}}{\log\left(\frac{1}{2eL^2}\log\frac{4d\exp(2dL^2)}{\alpha}\right)}\right)$. Let $\Omega_L^d = \{x \in \mathbb{R}^d \mid \|x\|_\infty \le L\}$. Define a mapping function F from any $X \subset \Omega_L^d$ so $F(X) = \sum_{x\in X} G(x)$, which is a vector in $\mathbb{R}^m$. Then for any $P, Q \subset \Omega_L^d$ with probability at least $9/10$*

$$\left|\|F(P)-F(Q)\|^2 - \mathtt{D}_K^2(P,Q)\right| \le \varepsilon\mathtt{D}_K^2(P,Q) + \alpha.$$

The mapping $G : \mathbb{R}^d \to \mathbb{R}^m$ can be computed in $O\left(\frac{d^2}{\varepsilon^2}\log\frac{d}{\varepsilon} + ds\right)$ time.

Proof. To analyze the GaussianSketch, we need to account for error from two sources: from the RecursiveTensorSketch (using Lemma 3) and parameter m, and from the truncation of the Taylor expansion at s (using Lemma 6). In this case we analyze the following infinite expansion

$$\mathtt{D}_K^2(P,Q) = \left\|\sum_{x\in P\cup Q}\beta_x\bar{y}_x^{(1)}\otimes\cdots\otimes\bar{y}_x^{(d)}\right\|^2,$$

where each $\bar{y}_x^{(j)}$ is an infinite dimension vector with ith coordinate $\exp(-x_j^2)\sqrt{\frac{2^{i-1}}{(i-1)!}}x_j^{i-1}$.

Let $v = \sum_{x\in P\cup Q} \beta_x \bar{y}_x^{(1)} \otimes \cdots \otimes \bar{y}_x^{(d)}$. Then by Lemma 3 by setting $m = O(d/\varepsilon^2)$ we have with probability at least 9/10 that

$$\left| \left\| \sum_{x\in P\cup Q} \beta_x G(x) \right\|^2 - \|v\|^2 \right| \leq \varepsilon \|v\|^2.$$

Next note that $(\sum_{x\in P\cup Q} |\beta_x|)^2 \leq 4 = \xi$. So by Lemma 6 the truncation by only s terms can be accounted for as

$$\mathtt{D}_K^2(P,Q) - \|v\|^2 = \beta^T \left(K_{P\cup Q} - K^{\mathsf{GS}}_{P\cup Q,s}\right)\beta \leq 4d\exp(2dL^2)\left(\frac{2eL^2}{s}\right)^2 \leq \alpha,$$

where $K_{P\cup Q}$ and $K^{\mathsf{GS}}_{P\cup Q,s}$ are defined as in Lemma 6 with $X = P\cup Q$.

Combining these together we have

$$(1-\varepsilon)\left(\mathtt{D}_K^2(P,Q) - \alpha\right) \leq (1-\varepsilon)\|v\|^2 \leq |F(P) - F(Q)| \leq (1+\varepsilon)\|v\|^2 \leq (1+\varepsilon)\mathtt{D}_K^2(P,Q).$$

and hence as desired

$$\left|\|F(P) - F(Q)\|^2 - \mathtt{D}_K^2(P,Q)\right| \leq \varepsilon \mathtt{D}_K^2(P,Q) + \alpha.$$

Recall that the running time of G for mapping a point is

$$O(dm\log m + ds) = O\left(\frac{d^2}{\varepsilon^2}\log\frac{d}{\varepsilon} + ds\right). \qquad \blacktriangleleft$$

Using the Gaussian Sketch HD for high dimensions. We first express $\exp(-\|x-y\|^2)$ as another infinite sum using Lemma 2. Starting with $\mathtt{D}_K^2(P,Q) = \sum_{x\in P\cup Q}\sum_{y\in P\cup Q}\beta_x\beta_y \exp\left(-\|x-y\|^2\right)$ where β_x is $\frac{1}{|P|}$ if $x\in P$ and $-\frac{1}{|Q|}$ if $x\in Q$, we have

$$\begin{aligned}\mathtt{D}_K^2(P,Q) &= \sum_{x\in P\cup Q}\sum_{y\in P\cup Q}\beta_x\beta_y \left\langle \exp(-\|x\|^2)\sqrt{\frac{2^i}{i!}}x^{\otimes i}, \exp(-\|y\|^2)\sqrt{\frac{2^i}{i!}}y^{\otimes i}\right\rangle \\ &= \sum_{i=0}^{\infty}\left\| \sum_{x\in P\cup Q}\beta_x \exp(-\|x\|^2)\sqrt{\frac{2^i}{i!}}x^{\otimes i}\right\|^2.\end{aligned}$$

▶ **Theorem 9.** *For any $\varepsilon, R, \alpha > 0$, let G be randomly chosen from* $\textsc{GaussianSketchHD}_{m_1,\ldots,m_s,s}$ *with $m_i = O\left(\frac{i}{\varepsilon^2}\right)$ and $s = \Theta\left(\frac{\log\frac{4\exp(2R^2)}{\alpha}}{\log\left(\frac{1}{2eR^2}\log\frac{4\exp(2R^2)}{\alpha}\right)}\right)$. Let $\Lambda_R^d = \{x\in\mathbb{R}^d \mid \|x\|_2 \leq R\}$. Define a mapping function F from any $X\subset\Lambda_L^d$ so $F(X) = \sum_{x\in X} G(x)$, which is a vector in $\mathbb{R}^m$ where $m = \sum_{i=1}^s m_i$. Then for any $P,Q\subset\Lambda_R^d$ with probability at least* 9/10

$$\left|\|F(P) - F(Q)\|^2 - \mathtt{D}_K^2(P,Q)\right| \leq \varepsilon \mathtt{D}_K^2(P,Q) + \alpha.$$

The mapping $G:\mathbb{R}^d\to\mathbb{R}^m$ can be computed in $O(\frac{s^3}{\varepsilon^2}\log\frac{s}{\varepsilon} + s^2 d)$ time.

Proof. Suppose $G(x) \in \mathbb{R}^m$ with $(m_{i-1}+1)$th coordinate to m_ith coordinate be $\sqrt{\frac{2^{i-1}}{(i-1)!}} \exp(-\|x\|^2) T_i(x^{\otimes i-1})$. Here, T_i is randomly chosen from $\textsc{RecursiveTensorSketch}_{d,m_i,i-1}$ for $i = 1, \ldots, s$.

We first need to invoke Lemma 4 to inherit the appropriate concentration bounds from the RecursiveTensorSketch. We use $t \times \frac{d^s-1}{d-1}$ matrices A and B as just row vectors with $t = 1$, and let $A = B$. In particular, define this single row as $z = \sum_{x \in P \cup Q} \beta_x [z_x^{(1)}, z_x^{(2)}, \ldots, z_x^{(s)}]$, then the conclusion of Lemma 4 is that with probability at least $1-\delta$

$$\left| \|z\|^2 - \left\| \sum_{x \in P \cup Q} \beta_x G(x) \right\|^2 \right|^2 = \left| \|z\|^2 - z S^T S z^T \right|_F^2 \leq \varepsilon^2 \|z\|^4.$$

So by Lemma 7 the truncation by only s terms can be accounted for as

$$\mathtt{D}_K^2(P,Q) - \|z\|^2 = \beta^T (K_{P \cup Q} - K^{\mathsf{HD}}_{P \cup Q, s})) \beta \leq 4d \exp(2dL^2)(2eL^2/s)^2 \leq \alpha,$$

where $K_{P \cup Q}$ and $K^{\mathsf{HD}}_{P \cup Q, s}$ are defined as in Lemma 7 with $X = P \cup Q$.

Combining these together we have

$$(1-\varepsilon)(\mathtt{D}_K^2(P,Q) - \alpha) \leq (1-\varepsilon)\|z\|^2 \leq \|F(P) - F(Q)\|^2 \leq (1+\varepsilon)\|z\|^2 \leq (1+\varepsilon)\mathtt{D}_K^2(P,Q).$$

and hence as desired

$$\left| \|F(P) - F(Q)\|^2 - \mathtt{D}_K^2(P,Q) \right| \leq \varepsilon \mathtt{D}_K^2(P,Q) + \alpha.$$

Recall that the running time of G for mapping a point is $O(\sum_{i=1}^s i m_i \log m_i + id) = O(\sum_{i=1}^s \frac{i^2}{\varepsilon^2} \log \frac{i}{\varepsilon} + id) = O(\frac{s^3}{\varepsilon^2} \log \frac{s}{\varepsilon} + s^2 d)$. ◀

4 Extensions and Data Analysis Implications

There are many data analysis applications where useful sketched bounds almost immediately follow from this new embedding. Before we begin, we start by improving the dimensionality of the embedding with a simple post-processing. We can applying a Johnson-Lindenstrauss-type embedding [25, 3, 4, 1] to the m-dimensional space to obtain $O(1/\varepsilon^2)$-dimensional space that, with constant probability, preserves the distance of a pair of point sets. Furthermore, we can use median trick to boost the success probability to $1-\delta$ by running $O(\log \frac{1}{\delta})$ independent copies. For applications in kernel two-sample hypothesis testing and nearest neighbor searching, setting δ depends on the number of queries q we make, for instances the bounded number needed for k-means clustering [16], now applied to kernel k-means. These results are useful for reducing the *storage space* of data representations. Recall that the running time of JL embedding from m-dimensional space to ρ-dimensional space is $O(m \log \rho + \rho^2)$ [3, 4].

4.1 Kernel Two-Sample Test

The kernel two-sample test [24] is a "non-parametric" hypothesis test between two probability distributions represented by finite samples P and Q; let $n = |P \cup Q|$. Then this test simply calculates $\mathtt{D}_K(P,Q)$, and if the value is large enough it rejects the null hypothesis that P and Q represent the same distribution. Since its introduction a few year ago it has seen many applications and relations; see the recent 140 page survey [33]. Zhao and Deng [48]

proposed to speed this test up for large sets using RFFs which improves runtime and in some cases even statistical power. While several improvements are suggested [47] including using FastFood [29], these all only provide additive ε-error.

Consider $P \sim \mu_P$ and $Q \sim \mu_Q$. If $\mu_P = \mu_Q$, then empirical distributions P, Q may have $\mathtt{D}_K(P,Q) = \Theta(1/n)$. Hence distinguishing the case of $\mu_P = \mu_Q$ from them not being equal would either require additive error $\varepsilon = \Theta(1/n)$, or relative $(1+\varepsilon)$-error with a minimum $\Theta(1/n)$ additive error. RFFs would require $\Theta(1/\varepsilon^2) = \Theta(n^2)$ dimensions, so one may just as well compute $\mathtt{D}_K(P,Q)$ exactly in $O(n^2)$ time. In our approach, we can set ε to be a constant (say $\varepsilon = 0.2$) and α to be $\Theta(1/n)$. Assuming a constant region diameter, the total running time is $O\left(\frac{n \log n}{\log\log n}\right)$ in the low dimensional case (by Theorem 8) or $O\left(\frac{n \log^2 n(\log n + d)}{\log^2 \log n}\right)$ in the high dimensional case (by Theorem 9).

Another way to determine if $\mathtt{D}_K(P,Q)$ should estimate P and Q as distinct, is to run permutation tests. That is for some large number (e.g., $q = 1000$) of trials, select two sets P_j, Q_j iid from $P \cup Q$, of size $|P|$ and $|Q|$ respectively. For each generated pair we calculate (or estimate using Theorem 8 or Theorem 9) the value of $\mathtt{D}_K(P_j, Q_j)$, and then use the 95th-percentile of these values as a threshold. Note since each P_j, Q_j is drawn from the same domain as P, Q, then the guarantees on the accuracy of the featurized estimate carries over directly even under a large q number of permutations.

4.2 LSH for Point Sets, Geometric Distributions

The new results also allow us to immediately design LSH and nearest neighbor structures for the kernel distance by relying on standard Euclidean LSH [6]. Building a search engine for low-dimensional shapes [21] has long been a goal in computational geometry and geometric modeling. A difficulty arises in that many of the best-known shape distance measures require an alignment (e.g., Frechet [20, 5] or earth movers [11]) which creates many challenges in designing LSH-type procedures. Some methods have been designed, but with limitations, e.g., on point set size for earth mover distance [7] or number of segments in curves for discrete Frechet [18]. The kernel distance provides an alternative distance for shapes, low-dimensional distributions, or curves [26]; it can encode normals or tangents as well to encode direction information of curves [23]. That is, given two shapes composed of (or approximated by) point sets P_i, P_j, the distance between the shapes is simply $\mathtt{D}_K(P_i, P_j)$.

Given a family of point sets $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$ such that each $P_i \subset \mathbb{R}^d$ has size at most n, an ε-approximate nearest neighbor of a query point set Q is a point set $\hat{P} \in \mathcal{P}$ so that $\mathtt{D}_K(\hat{P}, Q) \leq (1+\varepsilon) \min_{P_j \in \mathcal{P}} \mathtt{D}_K(P_j, Q)$. Here, we assume that $\mathtt{D}_K(P_i, P_j) \geq \alpha'$ for any $i \neq j$. For $\varepsilon \leq 1/2$, we can embed each P_j to $F(P_j) \in \mathbb{R}^D$, and then invoke the key result from Andoni and Indyk [6] for a c'-approximate nearest neighbor, so the total error factor is $c'(1+\varepsilon)$. Overall, we can retrieve a c-approximate nearest neighbor (setting $c = c'(1+\varepsilon)$) to a query $Q \subset \mathbb{R}^d$ with $O(DN^{1/c^2 + o(1)})$ query time after using $O(DN^{1+1/c^2+o(1)})$ space and $O(DN^{1+1/c^2+o(1)} + N(\frac{n \log \frac{1}{\varepsilon\alpha'}}{\log\log \frac{1}{\varepsilon\alpha'}} + \frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}))$ preprocessing when d is small or $O(DN^{1+1/c^2+o(1)} + Nn(\frac{\log^2 \frac{1}{\varepsilon\alpha'}(\log \frac{1}{\varepsilon\alpha'} + d)}{\varepsilon^2 \log^2 \log \frac{1}{\varepsilon\alpha'}}))$ preprocessing when d is large, both assuming a data region with constant diameter.

References

1 Dmitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Comp. & Sys. Sci.*, 66:671–687, 2003.

2 Thomas D Ahle, Michael Kapralov, Jakob BT Knudsen, Rasmus Pagh, Ameya Velingker, David P Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *SODA*, 2020.

3 Nir Ailon and Edo Liberty. Fast dimension reduction using rademacher series on dual bch codes. *Discrete & Computational Geometry*, 42(615), 2009.

4 Nir Ailon and Edo Liberty. An almost optimal unrestricted fast johnson-lindenstrauss transform. In *SODA*, 2011.

5 Helmut Alt and Leonidas J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation: A survey. In *Handbook of Computational Geometry*. -, 1996.

6 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, 2006.

7 Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. Earth mover distance over high-dimensional spaces. In *SODA*, 2008.

8 N. Aronszajn. Theory of reproducing kernels. *Trans. AMS*, 68:337–404, 1950. URL: `http://www.jstor.org/stable/1990404`.

9 Haim Avron, Michael Kapralov, Cameron Musco, Chistopher Musco, Ameya Velingker, and Amir Zandier. Random fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *ICML*, 2017.

10 Haim Avron, Huy L. Nguyen, and David P. Woodruff. Subspace embeddings for the polynomial kernel. In *NIPS*, 2014.

11 Khanh Do Ba, Huy L. Nguyen, Huy N. Nguyen, and Ronnit Rubinfeld. Sublinear time algorithms for earth mover's distance. *Theory Comput Syst*, 48:428–442, 2011.

12 Moses Charikar and Paris Siminelakis. Hashing-based-estimators for kernel density in high dimensions. In *FOCS*, 2017.

13 Edgar Chaźez, Ana C. Chávez Cáliz, and Jorge L. López-López. Affine invariants of generalized polygons and matching under affine transformations. *Computational Geometry: Theory and Applications*, 58:60–69, 2017.

14 Di Chen and Jeff M. Phillips. Relative error embeddings for the gaussian kernel distance. In *Algorithmic Learning Theory*, 2017.

15 Kenneth L Clarkson and David P Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 205–214. ACM, 2009.

16 Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Mădălina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *STOC*, 2015.

17 Andrew Cotter, Joseph Keshet, and Nathan Srebro. Explicit approximations of the gaussian kernel. *arXiv preprint arXiv:1109.4603*, 2011.

18 Anne Driemel and Francesco Silvestri. Locality-sensitive hashing of curves. In *33rd International Symposium on Computational Geometry*, 2017.

19 Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

20 Thomas Eiter and Heikki Mannila. Computing discrete Frechet distance. Technical report, Christian Doppler Laboratory for Expert Systems, 1994.

21 Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3D models. *ACM Transactions on Graphics*, 22:83–105, 2003.

22 Mina Ghashami, Daniel Perry, and Jeff M. Phillips. Streaming kernel principal component analysis. In *AIStats*, 2016.

23 Joan Glaunès and Sarang Joshi. Template estimation form unlabeled point set data and surfaces for computational anatomy. In *Math. Found. Comp. Anatomy*, 2006.

24 Arthur Gretton, Marsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alex J. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

25 William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

26 Sarang Joshi, Raj Varma Kommaraju, Jeff M. Phillips, and Suresh Venkatasubramanian. Comparing distributions and shapes using the kurrent distance. In *Proceedings 27th Annual Symposium on Computational Geometry*, 2011. arXiv:1001.0591.

27 Ravi Kannan, Santosh Vempala, and David Woodruff. Principal component analysis and higher correlations for distributed data. In *Conference on Learning Theory*, pages 1040–1057, 2014.

28 Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In *FOCS*, 2017.

29 Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood — approximating kernel expansions in loglinear time. In *ICML*, 2013.

30 David Lopez-Paz, Suvrit Sra, Alex Smola, Zoubin Ghahramani, and Bernhard Schölkopf. Randomized nonlinear component analysis. *ICML*, 2014.

31 J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 209:441–458, 1909.

32 Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *arXiv preprint arXiv:1605.09522*, 2016.

33 Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10:1–141, 2017.

34 Cameron Musco and David Woodruff. Is input sparsity time possible for kernel low-rank approximation? In *NeurIPS*, 2017.

35 Cameron Musco and David P. Woodruff. Sublinear time low-rank approximation of positive semidefinite matrices. In *FOCS*, 2017.

36 Jeff M. Phillips and Suresh Venkatasubramanian. A gentle introduction to the kernel distance. Technical report, Arxiv:1103.1625, 2011.

37 Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

38 Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, pages 2263–2291, 2013.

39 Alex J. Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert space embedding for distributions. In *ICALT*, 2007.

40 Bharath Sriperumbudur et al. On the optimal estimation of probability measures in weak and strong topologies. *Bernoulli*, 22(3):1839–1893, 2016.

41 Bharath Sriperumbudur and Nicholas Sterge. Approximate kernel pca using random features: Computational vs. statistical trade-off. Technical report, arXiv: 1706.06296, 2018.

42 Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *JMLR*, pages 2389–2410, 2011.

43 Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R. G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010.

44 Enayat Ullah, Poorya Mianjy, Teodor V. Marinov, and Raman Arora. Streaming kernel pca with $\tilde{o}(\sqrt{n})$ random features. In *NeruIPS*, 2018.

45 Shusen Wang, Alex Gittens, and Michael W. Mahoney. Scalable kernel k-means clustering with nystrom approximation: Relative-error bounds. *JMLR*, [arXiv:1706.02803], (to appear).

46 David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10:1–157, 2014.

47 Wojciech Zaremba, Arthur Gretton, and Matthew Blaschko. B-tests: Low variance kernel two-sample tests. In *NIPS*, 2013.

48 Ji Zhao and Deyu Meng. Fastmmd: Ensemble of circular discrepancy for efficient two-sample test. *Neural Computation*, 27:1354–1372, 2015.

A Gaussian Kernel PCA

Let k be a positive integer and $\varepsilon > 0$. Avron *et al.* [10] provide the following algorithm. Suppose S and T are randomly chosen from $\text{RecursiveTensorSketch}_{s,m,d}$ and $\text{RecursiveTensorSketch}_{s,r,d}$ respectively where $m = \Theta(d(k^2 + \frac{k}{\varepsilon}))$ and $r = \Theta(\frac{dm^2}{\varepsilon^2})$. Given n vectors $v^{(1)}, \ldots, v^{(n)} \in \mathbb{R}^{s^d}$, compute $n \times m$ matrix M with ith row as $S(v^{(i)})$ and $n \times r$ matrix N that ith row as $T(v^{(i)})$. Let U be the orthonormal basis for column space of M and W be $m \times k$ matrix containing top k left singular vector of $U^T N$. Finally, return $V = UW$. This algorithm has the following guarantee.

▶ **Lemma 10** ([10] with straightforward modification). *Given a n-by-s^d matrix A, a positive integer k and $\varepsilon > 0$. The above algorithm that has rows of A as input returns a matrix V such that*

$$\left\|A - VV^T A\right\|_F^2 \leq (1+\varepsilon)\left\|A - [A]_k\right\|_F^2$$

where $[A]_k$ is the best rank-k approximation of A.

Now, we can directly modify the above algorithm into our context for rank-k Gaussian low-rank approximation. Given a point set $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and a positive integer s. Suppose G and H are randomly chosen from $\text{GaussianSketch}_{m,s}$ and $\text{GaussianSketch}_{r,s}$ respectively. Recall that $m = \Theta(d(k^2 + \frac{k}{\varepsilon}))$ and $r = \Theta(\frac{dm^2}{\varepsilon^2})$. Compute the $n \times m$ matrix M with ith row as $G(x_i)$ and $n \times r$ matrix N with ith row as $H(x_i)$. Let U be the orthonormal basis for column space of M and W be $m \times k$ matrix containing top k left singular vector of $U^T N$. Finally, return $V = UW$.

▶ **Theorem 11.** *Let $\varepsilon, L, \alpha > 0$ and $s = \Theta\left(\frac{\log \frac{4n^2 d \exp(2dL^2)}{\alpha}}{\log\left(\frac{1}{2eL^2}\log \frac{4n^2 d \exp(2dL^2)}{\alpha}\right)}\right)$. For $\Omega_L^d = \{x \in \mathbb{R}^d \mid \|x\|_\infty \leq L\}$ and $X \subset \Omega_L^d$, and let A_X be a pd matrix with elements $(A_X)_{i,j} = K(x_i, x_j) = \exp(-\|x_i - x_j\|^2)$ for $x_i, x_j \in X$ and factorization $A_X = B_X B_X^T$. Then with constant probability*

$$\left\|B_X - VV^T B_X\right\|_F^2 \leq (1+\varepsilon)\left\|B_X - [B_X]_k\right\|_F^2 + \alpha.$$

The runtime to compute V is $O\left(nds + n\frac{d^4(k^2+\frac{k}{\varepsilon})^3}{\varepsilon^2}\right)$.

Proof. Let $v_x^{(i)}$ be a vector in $\mathbb{R}^s$ with jth coordinate to be $\exp(-x_i^2)\sqrt{\frac{2^{j-1}}{(j-1)!}}x_i^{j-1}$ for any $x \in \mathbb{R}^d$.

By Lemma 10, taking A_s as an $n \times s^d$ matrix with ith row as $v_{x_i}^{(1)} \otimes \cdots \otimes v_{x_i}^{(d)}$. We have

$$\left\|A_s - VV^T A_s\right\|_F^2 \leq (1+\varepsilon)\left\|A_s - [A_s]_k\right\|_F^2$$

From Lemma 6, $v^T(B_X B_X^T - A_s A_s^T)v \leq \left(\sum_{i=1}^n |v_i|\right)^2 d\exp(2dL^2)\left(\frac{2eL^2}{s}\right)^s \leq \alpha/n$. To see this expression is at most α/n, first observe that columns of V are orthonormal, and

therefore, the norm of each row of $I - VV^T$ is at most 2. Hence, $\left(\sum_{i=1}^{n} |v_i|\right)^2 \leq 4n$. Then the choice of s and Lemma 5 with $\xi = 4n^2$, $a = d\exp(2dL^2)$ and $b = 2eL^2$ complete this derivation.

We now have

$$\begin{aligned}\left\|B_X - VV^T B_X\right\|_F^2 &= \mathsf{Tr}((I - VV^T)B_X B_X^T (I - VV^T)^T) \\ &\leq \left\|A_s - VV^T A_s\right\|_F^2 + \mathsf{Tr}((I - VV^T)(B_X B_X^T - A_s A_s^T)(I - VV^T)^T) \\ &\leq \left\|A_s - VV^T A_s\right\|_F^2 + \alpha\end{aligned}$$

On the other hand, by Lemma 1, $B_X B_X^T - A_s A_s^T$ is still positive definite. Therefore,

$$\begin{aligned}&\|A_s - [A_s]_k\|_F^2 \\ &= \left\|A_s - UU^T A_s\right\|_F^2 \quad \text{where } U \text{ is the matrix of top-}k \text{ left singular vectors of } A_s \\ &\leq \left\|A_s - U'U'^T A_s\right\|_F^2 \quad \text{where } U \text{ is the matrix of top-}k \text{ left singular vectors of } B_X \\ &= \|B_X - [B_X]_k\|_F^2 - \mathsf{Tr}((I - U'U'^T)(B_X B_X^T - A_s A_s^T)(I - U'U'^T)) \\ &\leq \|B_X - [B_X]_k\|_F^2 \quad \text{recall that } B_X B_X^T - A_s A_s^T \text{ is positive definite}\end{aligned}$$

We can plug in everything.

$$\begin{aligned}\left\|B_X - VV^T B_X\right\|_F^2 &\leq \left\|A_s - VV^T A_s\right\|_F^2 + \alpha \\ &\leq \|A_s - [A_s]_k\|_F^2 + \alpha \\ &\leq \|B_X - [B_X]_k\|_F^2 + \alpha.\end{aligned}$$

To see the running time, it takes $O(d(s + m\log m))$ to compute $G(\cdot)$ and $O(d(s + r\log r))$ time to compute $H(\cdot)$, and hence n times as much to compute matrices M and N. We can compute the basis U of M in $O(nm^2)$ time, and the projection $U^T N$ in $O(nrm)$ time. The basis W takes $O(rm^2)$ time, and the final low rank basis $V = UW$ takes $O(nmk)$ time. Thus the total runtime is $O(nd(s + m\log m + r\log r) + nm^2 + nrm + rm^2 + nmk) = O(nd(s + rm))$ using that $r > m^2 > k^4$ that $m > \log r$, and assuming $n > r$. Now using $m = O(d(k^2 + k/\varepsilon))$ and $r = O(dm^2/\varepsilon^2) = O(d^3(k^4 + k^2/\varepsilon^2)/\varepsilon^2)$ and we have a total time of $O\left(nds + n\frac{d^4(k^2+\frac{k}{\varepsilon})^3}{\varepsilon^2}\right)$. ◀

Gaussian Low Rank Approximation with Gaussian Sketch HD in High Dimensions. Now, we can also modify the above algorithm into our context for rank-k Gaussian low-rank approximation in another way. Given a point set $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and a positive integer s. Suppose G and H are randomly chosen from $\textsc{GaussianSketchHD}_{m_1,\ldots,m_s,s}$ and $\textsc{GaussianSketchHD}_{r_1,\ldots,r_s,s}$ respectively. Here, $m_i = \Theta(i(k^2 + \frac{k}{\varepsilon}))$ and $r_i = \Theta(\frac{im^2}{\varepsilon^2})$ where $m = \sum_{i=1}^{s} m_i$. Compute the $n \times m$ matrix M with ith row as $G(x_i)$ and $n \times r$ matrix N with ith row as $H(x_i)$. Let U be the orthonormal basis for column space of M and W be $m \times k$ matrix containing top k left singular vector of $U^T N$. Finally, return $V = UW$.

Note that a hash function in GaussianSketchHD is not directly applying a hash function in RecursiveTensorSketch. Therefore, Lemma 10 cannot be directly applied. However, we can still exploit the structure of it in order to prove the same lemma.

As Avron *et al.* [10] suggest, it is generally possible by combining Lemma 4 and arguments in [10, 15, 27]. We have the following lemma. Here, denote A_s is a $n \times \frac{d^s-1}{d-1}$ matrix that ith row as z_{x_i} for given point set $X = \{x_1, x_2 \ldots, x_n\} \subset \mathbb{R}^d$.

▶ **Lemma 12.** *Given a point set $X \subset \mathbb{R}^d$, a positive integer k and $\varepsilon > 0$. The above algorithm returns a matrix V such that*

$$\left\|A_s - VV^T A_s\right\|_F^2 \leq (1+\varepsilon)\left\|A_s - [A_s]_k\right\|_F^2$$

where $[A]_k$ is the best rank-k approximation of A.

Before getting into Lemma 12, the following lemma from [10] which is implied by Lemma 4 would be helpful.

▶ **Lemma 13** ([10] implied by Lemma 4 with straightforward modification)**.** *For any positive integer k', given any $\frac{d^s-1}{d-1} \times k'$ matrix B with orthonormal columns, we have $\left\|B^T S^T SB - I\right\|_2 \leq \varepsilon$. Here, S is randomly chosen from* GaussianSketchHD$_{n_1,\ldots,n_s,s}$ *where $n_i = \frac{ik'^2}{\varepsilon^2}$.*

Proof. (of Lemma 12)

In the proof of Theorem 3.1 from [15], the only properties of S used are

- Given any $\frac{d^s-1}{d-1} \times k$ matrix B with orthonormal columns, we have $\left\|B^T S^T SB - I\right\|_2 \leq \varepsilon_0$ for some constant $\varepsilon_0 > 0$
- For any two matrices A, B with $\frac{d^s-1}{d-1}$ columns, $\left\|AB^T - AS^T SB^T\right\|_F \leq \sqrt{\frac{\varepsilon}{k}}\left\|A\right\|_F\left\|B\right\|_F$

The first property can be shown by Lemma 13 since we pick $m_i = \Omega(ik^2)$ and the second property can be shown by Lemma 4 since we pick $m_i = \Omega(\frac{ik}{\varepsilon})$. Also, Theorem 3.1 of [15] implies Lemma 4.2 of [15] which means there is a matrix Z such that $\left\|UZ - A_s\right\|_F \leq (1+\varepsilon)\left\|A_s - [A_s]_k\right\|_F$ in our context. Combining Lemma 4.3 of [15], we have

$$\left\|U[U^T A_s] - A_s\right\|_F \leq (1+\varepsilon)\left\|A_s - [A_s]_k\right\|_F \tag{2}$$

Now, Lemma 13 implies Lemma 2.1 from [27] and further implies

$$\left\|WW^T U^T A_s - A_s\right\|_f \leq (1+\varepsilon)\left\|A_s - [A_s]_k\right\|_F \tag{3}$$

by setting k' in Lemma 13 be m and picking $r_i = \Theta(\frac{3^i m^2}{\varepsilon^2})$. Using equation (2) and (3) in the proof of Theorem 1.1 from [27], we have our conclusion $\left\|A_s - UWW^T U^T A_s\right\|_F^2 = \left\|A_s - VV^T A_s\right\|_F^2 \leq (1+\varepsilon)\left\|A_s - [A_s]_k\right\|_F^2$.

◀

▶ **Theorem 14.** *Let $\varepsilon, R, \alpha > 0$ and $s = \Theta\left(\frac{\log \frac{4n^2 \exp(2R^2)}{\alpha}}{\log\left(\frac{1}{2eR^2}\log \frac{4n^2\exp(2R^2)}{\alpha}\right)}\right)$. For $\Lambda_R^d = \{x \in \mathbb{R}^d \mid \|x\|_2 \leq R\}$ and $X \subset \Lambda_R^d$, and let A_X be a pd matrix with elements $(A_X)_{i,j} = K(x_i, x_j) = \exp(-\left\|x_i - x_j\right\|^2)$ for $x_i, x_j \in X$ and factorization $A_X = B_X B_X^T$. Then with constant probability*

$$\left\|B_X - VV^T B_X\right\|_F^2 \leq (1+\varepsilon)\left\|B_X - [B_X]_k\right\|_F^2 + \alpha.$$

The runtime to compute V is $O(nds^2 + n\frac{3^{4s}(k^2+\frac{k}{\varepsilon})^3}{\varepsilon^2})$.

Proof. By Lemma 12, we have

$$\left\|A_s - VV^T A_s\right\|_F^2 \leq (1+\varepsilon)\left\|A_s - [A_s]_k\right\|_F^2.$$

From Lemma 7, $v^T(B_X B_X^T - A_s A_s^T)v \leq \left(\sum_{i=1}^n |v_i|\right)^2 \exp(2R^2)\left(\frac{2eR^2}{s}\right)^s \leq \alpha/n$ with our setting of s as long as $\left(\sum_{i=1}^n |v_i|\right)^2 \leq 4n$. Indeed the columns of V are orthonormal, so the norm of each row of $I - VV^T$ is at most 2, and thus $\left(\sum_{i=1}^n |v_i|\right)^2 \leq 4n$.

We now have

$$\begin{aligned}\left\|B_X - VV^T B_X\right\|_F^2 &= \mathsf{Tr}((I-VV^T)B_X B_X^T (I-VV^T)^T) \\ &\le \left\|A_s - VV^T A_s\right\|_F^2 + \mathsf{Tr}((I-VV^T)(B_X B_X^T - A_s A_s^T)(I-VV^T)^T) \\ &\le \left\|A_s - VV^T A_s\right\|_F^2 + n\cdot(\alpha/n)\end{aligned}$$

Also by Lemma 2, $B_X B_X^T - A_s A_s^T$ is still positive definite. Therefore,

$$\begin{aligned}&\|A_s - [A_s]_k\|_F^2 \\ &= \left\|A_s - UU^T A_s\right\|_F^2 \qquad \text{where } U \text{ is the matrix of top-}k \text{ left singular vectors of } A_s \\ &\le \left\|A_s - U'U'^T A_s\right\|_F^2 \qquad \text{where } U' \text{ is the matrix of top-}k \text{ left singular vectors of } B_X \\ &= \|B_X - [B_X]_k\|_F^2 - \mathsf{Tr}((I-U'U'^T)(B_X B_X^T - A_s A_s^T)(I-U'U'^T)) \\ &\le \|B_X - [B_X]_k\|_F^2 \qquad \text{recall that } B_X B_X^T - A_s A_s^T \text{ is positive definite}\end{aligned}$$

We can plug in everything.

$$\begin{aligned}\left\|B_X - VV^T B_X\right\|_F^2 &\le \left\|A_s - VV^T A_s\right\|_F^2 + \alpha \\ &\le \|A_s - [A_s]_k\|_F^2 + \alpha \\ &\le \|B_X - [B_X]_k\|_F^2 + \alpha\end{aligned}$$

To see the running time, it takes $O(\sum_{i=1}^{s} i(d + m_i \log m_i))$ to compute $G(\cdot)$ and $O(\sum_{i=1}^{s} i(d + r_i \log r_i))$ time to compute $H(\cdot)$. Using that $r_i > m_i^2 > k^4$ and $m_i > 1/\varepsilon$ then it takes less time to compute $H(\cdot)$ than $G(\cdot)$, and this runtime is $O(ds^2 + s^2 r_s \log r_s) = O(ds^2 + s^2 r \log r)$ since the r_i values are exponentially increasing in i, and so $r_s = O(r)$ for $r = \sum_{i=1}^{s} r_i$. The time to compute M and N is n time longer.

We can compute the basis U of M in $O(nm^2)$ time, and the projection $U^T N$ in $O(nrm)$ time – this step is the post-sketch bottlneck. The basis W takes $O(rm^2)$ time, and the final low rank basis $V = UW$ takes $O(nmk)$ time. Thus the total runtime is $O(n(ds^2 + s^2 r \log r) + nm^2 + nrm + rm^2 + nmk) = O(n(ds^2 + rm))$ using that $r > m^2 > k^4$ that $m > s^2 \log r$, and assuming $n > r$. Now using $m = O(s^2(k^2 + k/\varepsilon))$ and $r = O(sm^2/\varepsilon^2) = O(s^3(k^4 + k^2/\varepsilon^2)/\varepsilon^2)$ and we have a total time of $O(nds^2 + ns^4(k^2 + \frac{k}{\varepsilon})^3/\varepsilon^2)$. ◀