58th Annual IEEE Symposium on Foundations of Computer Science

Hashing-Based-Estimators for Kernel Density in High Dimensions

Moses Charikar
Department of Computer Science
Stanford University
moses@cs.stanford.edu

Paris Siminelakis
Department of Electrical Engineering
Stanford University
psimin@stanford.edu

***Abstract*—Given a set of points $P \subset \mathbb{R}^d$ and a kernel k, the Kernel Density Estimate at a point $x \in \mathbb{R}^d$ is defined as $\mathrm{KDE}_P(x) = \frac{1}{|P|}\sum_{y\in P} k(x,y)$. We study the problem of designing a data structure that given a data set P and a kernel function, returns *approximations to the kernel density* of a query point in *sublinear time*. We introduce a class of unbiased estimators for kernel density implemented through locality-sensitive hashing, and give general theorems bounding the variance of such estimators. These estimators give rise to efficient data structures for estimating the kernel density in high dimensions for a variety of commonly used kernels. Our work is the first to provide data-structures with theoretical guarantees that improve upon simple random sampling in high dimensions.**

***Keywords*-Kernel Density, Locality Sensitive Hashing, Kernel-Matrix Vector Multiplication, Cell-probe model**

I. Introduction

A fundamental question in Statistics and Learning Theory is the following: *given a set of points $P \subset \mathbb{R}^d$ sampled from some unknown distribution $\mathcal{D}$ estimate the probability at an arbitrary point $x \in \mathbb{R}^d$*. This problem is known as *density estimation* and different ways to formalize it lead to very different statistical and computational tasks. In the past two decades the problem has attracted significant interest in theoretical computer science [10], [1]. Some of the most important problems that have been studied are learning discrete distributions [23], learning mixture models [36], and more recently the topic of robust estimation in high dimensions [13], [24]. In this paper, we focus on *Kernel Density Estimation* (KDE), one of the most widely developed methods in non-parametric estimation.

A. Kernel Density Estimation

In this approach, given a set of n points P, starting from the empirical distribution $\tilde{\mu}(x) = \frac{1}{n}\sum_{y\in P}\delta_y$, one obtains a smooth distribution by "convolving" it with a kernel function k, whose smoothness is typically controlled by a parameter $\sigma > 0$ called the *bandwidth*.

Definition 1 (Kernel Density)**.** *Given a kernel function $k_\sigma : \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$ and a dataset $P \subset \mathbb{R}^d$ we define the Kernel Density (KD) of P at a point $x \in \mathbb{R}^d$ as:*

$$\mathrm{KDE}_P(x) := \frac{1}{|P|}\sum_{y\in P} k_\sigma(x,y) \tag{1}$$

This is a natural way to extend the function smoothly from a discrete set of points to the whole space that is independent of any particular parametric assumption on the underlying distribution of the data. Selecting the kernel and bandwidth are intensively studied subjects in the literature of non-parametric estimation [12] for which there is still ongoing theoretical research [17]. The kernel function $k(x,y)$ is typically a function of only $x-y$ (shift invariant kernels) or just the euclidean distance $\|x-y\|$ (radial kernels). One of the most prominent functions is the **Gaussian kernel**

$$k_\sigma(x,y) = \exp\left(-\frac{\|x-y\|^2}{\sigma^2}\right) \tag{2}$$

The importance of KDE lies in that it gives a simple and general way of approximating the underlying probability distribution that can be subsequently used to perform more complex and computationally intensive tasks. Examples include *mode estimation* [7], *outlier detection* [33], *local regression* [15], *reproducing kernel Hilbert spaces* [32], *density based clustering* [31], and *topological data analysis* [22], [16]. Kernel Density Estimation is consequently an important primitive that is a building block in many applications.

In all of the above settings, at some point, the following problem is solved: *given $P \subset \mathbb{R}^d$, $z \in \mathbb{R}^n$, compute* $\mathrm{KDE}_P^z(x) := \sum_{i=1}^n k(x,y_i)z_i$. This can be computed exactly in linear time, but this is prohibitively slow for large data sets, especially since it is needed repeatedly in applications of interest.

The problem has been studied extensively in the *batch* setting, where given a set of n points, the goal is to compute, for each of the points, a sum of contributions due to all the points, i.e. n queries of the above form. Such computations are prevalent in the field of scientific computing and involve computing approximations to $y = Kz$ where K is an $n \times n$ kernel matrix. In low

0272-5428/17 $31.00 © 2017 IEEE
DOI 10.1109/FOCS.2017.99

dimensions, Fast Multipole Methods (combining hierarchical space partitions with Taylor approximations) were developed [25] to reduce the trivial $O(n^2)$ runtime to $O(n \log n)$ (and $O(n)$ in some cases). The fast multipole algorithm [19] has been enormously influential in numerical analysis and scientific computing; it was named as one of the top 10 algorithms of the 20th century by the editors of *Computing in Science and Engineering* [14]. For this work, Greengard and Rokhlin received the 2001 Steele Prize. The KDE problem thus lies in the core of both scientific computing as well as machine learning.

In this paper, we study the problem of *approximately* computing the KDE. For most of the paper we fix k_σ to be the Gaussian Kernel and define the following computational problem:

Definition 2 (KDE Problem)**.** *Given a dataset $P \subset \mathbb{R}^d$ of n points, and $\epsilon, \delta, \tau \in [0,1]$ construct a data structure that given a query $x \in \mathbb{R}^d$ with $\mathrm{KDE}_P(x) = \mu \in [\tau, 1]$ returns a number $\hat{\mu}$ such that $\mathbb{P}[|\hat{\mu} - \mu| \geq \epsilon \cdot \mu] \leq \delta$. We call this problem the (μ, ϵ, δ)-*KDE problem.

B. Our Contribution

The starting point of our work is the old and tested idea of *importance sampling*. Given non-negative weights $w_1, \ldots, w_n$ and a distribution Q over $[n]$ (inducing probabilities $q_1, \ldots, q_n$), an *unbiased estimator* for $\mu := \frac{1}{n}\sum_{i=1}^{n} w_i$ is given by sampling $I \in [n]$ according to Q and returning $Z = w_I/(q_i n)$. The minimum variance estimator is obtained by setting $q_i^* = w_i / \sum_j w_j$ for which the variance is zero. What precludes us from obtaining such probabilities is that in our setting of KDE, the weights $w_i = w_i(x) := k(x, x_i)$ depend on the query x, and thus the sampling distribution Q needs to be adaptive to the query. Furthermore, having an ideal distribution Q^* indirectly involves knowing the *normalizing constant* $\sum_i w_i = n\mu$, the very quantity we wish to estimate. Thus, the main challenge in turning the idea of importance sampling into an algorithm is to have an *efficient way* to define an *adaptive sampling distribution* $Q(x)$ that has *low variance*. We next present our methods at an abstract level and subsequently show their implications for Kernel Density Estimation.

1) Hashing-Based-Estimators (HBE): Our main contribution is to introduce a hashing-based framework to *succinctly* define an adaptive distribution $\mathcal{Q}(x)$ and to provide sharp tools *bounding the variance* of the resulting unbiased estimator. Our estimators are formed by:

- **Preprocessing:** given a hash function h sampled from a hash family $\mathcal{H}$, where the colision probability of x, y is $p(x, y)$, we evaluate h on $x_1, \ldots, x_n$ and form the corresponding hash table H.
- **Querying:** let $H(x) \subseteq P$ denote the cell where the query x falls into and let $y \in P$ be a random element of $H(x)$, we return $Z_h(x) = \frac{k(x,y)}{p(x,y)}|H(x)|$ (or 0 if $H(x)$ is empty).

We say that a HBE has *complexity* T if the evaluation time (resp. space usage) is bounded by T (resp. $T \cdot n$). This *two-level sampling* procedure induces probabilities that depend both on P and $\mathcal{H}$. Although $|H(x)|$ is *a priori* a random variable, it becomes known to us through the preprocessing step. This sidesteps the issue of computing the normalizing constant separately for each query and is at the core of our approach.

The challenge in fully implementing this scheme is to bound the second moment of our estimator. In this regard, picking a random element from $H(x)$ turns out to be crucial, as it is this step that allows us to get an analytical handle on the second moment of Z_h. We provide two general theorems that bound the variance of *Hashing-Based-Estimators*. The first theorem applies to any such unbiased HBE and shows that

Theorem 1 (Two-points suffice)**.** *Up to absolute constants the variance of a HBE is maximized by datasets where there are* only two values *for the weights and sampling probabilities.*

This characterization is extremely useful as given $p_i(x) := p(x, x_i)$ and $w_i(x)$, it reduces bounding the variance to a simple case analysis. The bound depends on the compatibility between the probability and weights, and is captured by the maximum element of an explicitly defined matrix. Going beyond the general case we identify a natural class of HBE that induce sampling probabilities that vary as a power of the weights.

Definition 3. *An HBE is (β, M)-*scale free *for parameters $\beta \in (0,1]$ and $M \geq 1$ if, for all $i \in [n]$ and x, $M^{-1} \cdot w_i^{\beta}(x) \leq p_i(x) \leq M \cdot w_i^{\beta}(x)$.*

The second theorem provides a refined analysis of the variance of scale-free HBE that is able to capture additional structure when one exists. In particular, the upper bound on the variance improves when most of the contribution to $\mu = \mu(x)$ comes from relatively large weights w_i. We state here the weakest bound that assumes nothing about the weights.

Theorem 2. *Let Z be an unbiased (β, M)-scale free HBE for $\beta \in [\frac{1}{2}, 1]$. Then $\mathbb{E}[Z^2] \leq \mu^2 \cdot (4M^3/\mu^{\beta})$.*

The main technical tools behind the analysis are two Hölder-type inequalities that we develop. The first one (Lemma 1) is a simple two-sided (matrix) extension of

Hölder's inequality that bounds a quadratic form over the intersection of two weighted ℓ_1-balls. A clever application of this inequality gives us the proof of Theorem 1. The second (Lemma 2) is a non-trivial Hölder-type inequality for monotone vectors that in combination with some easy consequences of Hölder's inequality gives us the refined analysis of *scale-free* estimators. The proof also shows the possibility of exploiting other structural assumptions of the data. This is important in the statistical setting (where the data set is sampled from a distribution) or for parameter tuning in practice.

Our theorem shows that under no structural assumptions, the optimal choice is $\beta^* = 1/2$ and results in an estimator with relative variance $\frac{\mathrm{Var}[Z]}{(\mathbb{E}[Z])^2} \leq \mathrm{V}(\mu) = 4M^3 \cdot \mu^{-1/2}$. Using the *Median-of-means* (MoM) technique, we can estimate μ within a multiplicative accuracy of $(1 \pm \varepsilon)$ using $O\left(M^3 \frac{1}{\epsilon^2} \frac{1}{\sqrt{\mu}} \log(1/\delta)\right)$ independent samples. Observe that this does not directly imply an algorithm to estimate μ using this many samples, as setting the number of samples (sufficient for accurate estimation) requires approximate knowledge of μ, the very quantity we are aiming to estimate.

We resolve this issue by proposing an adaptive procedure that uses $O(1)$ times additional samples to get a constant factor approximation to μ. We start with an overestimate of μ and iteratively decrease it until we get close enough to the truth where a consistency check is satisfied. The resulting algorithm, *Adaptive Mean Relaxation*, is applicable to settings where one has an unbiased estimator whose (upper bound on) variance is a non-decreasing function of the mean μ and the relative variance is a decreasing function of the mean.

2) Kernel Density Estimation through Locality Sensitive Hashing: We next turn to address the (μ, ϵ, δ)-KDE problem. To provide intuition about the problem we first analyze two simple randomized estimators, the (uniform) Random Sampling (RS) estimator and an estimator based on Random Fourier Features (RFF) [30]. We show that in the worst case, the first has variance bounded by μ whereas the RFF estimator has constant variance. Using the MoM framework and our adaptive procedure, one immediately gets an algorithm to solve the KDE problem using $O(\min\{\frac{1}{\epsilon^2}\frac{1}{\mu}\log(1/\delta), n\})$ samples that is polynomial in $(\epsilon, \mu, \log(1/\delta))$.

In order to improve upon this simple bound, we employ the framework of Hashing-Based-Estimators instantiated with *Locality Sensitive Hashing* schemes. All our results follow a similar theme: (a) we obtain pointwise upper and lower bounds on the collision probabilities of the hash functions, and (b) we bound the variance of the estimator by invoking either Theorem 1 or Theorem 2 in cases where we are able to obtain scale-free estimators (cf. Table I-B2).

Table I
SCALE FREE ESTIMATORS FOR KDE USING LSH

Kernel	(β, M)	Complexity T
$e^{-\|x-y\|^2}$	$(\beta, e^{O(R^{\frac{4}{3}} \log\log n)})$	$e^{O(R^{\frac{4}{3}} \log\log n)}$
$e^{-\|\mathbf{x}-y\|}$	$(\beta, \sqrt{e}\,)$	$O(dR^2)$
$\frac{1}{1+\|\mathbf{x}-y\|_2^p}$	$(\frac{q}{p}, 3^q)$	$O(dp)$

Theorem 3 (Informal)**.** *There exist scale-free HBE for the Gaussian, Exponential and Polynomial kernel.*

Using our theorem for scale-free estimators and the adaptive algorithm, we arrive at our main result for the KDE problem.

Theorem 4. *For a kernel k and dataset P for which there exists a $(\frac{1}{2}, M)$-scale-free estimator with complexity T, there exists a data structure that solves the (μ, ϵ, δ)-KDE problem $\forall \mu \in [\tau, 1]$ using $O(M^3 \frac{1}{\epsilon^2} \frac{1}{\sqrt{\mu}} \log(1/\delta) T)$ time and $O(M^3 \frac{1}{\epsilon^2} \frac{1}{\sqrt{\tau}} \log(1/\delta) T \cdot n)$ space.*

As an application, we show that one can use such a data structure to get an *approximate vector-matrix multiplication algorithm* for Kernel matrices using time that is adaptive to the vector and is always bounded by $\tilde{O}(\frac{n^{1+o(1)}}{\sqrt{\tau}} 1/\epsilon^2)$ where ϵ, τ indicate respectively the relative and additive error per coordinate (cf. Theorem 12). This result is important as it improves on the main bottleneck of Kernel Ridge Regression [8], that is, multiplying a dense Kernel matrix with a vector and requires time $O(n^2)$ in general.

3) Lower Bounds for KDE problem: We also complement our results by providing a reduction between hard instances for the Approximate Nearest Neighbor Search problem to the (μ, ϵ, δ)-KDE problem. We show that the latter is at least as hard as the (r, c)-ANNS with $n = \frac{1}{\mu}$ points and $c = O(\frac{\log(n)}{\log(1/\epsilon)})$. Combined with the results of Panigrahy, Talwar, Wieder [28] and Andoni et al [5], we get non-trivial lower bounds in the *cell-probe model* with a *single probe* that captures an interesting class of algorithms based on *adaptive coresets*.

C. Related Work

The problem of Kernel Density Estimation although widely studied in low-dimensions [20] has largely been unexplored in high dimensions [26]. In recent parallel and independent work, Spring and Shrivastava [34] introduced the idea of using locality sensitive hashing as

a sampling scheme to estimate the *partition function* of log-linear models, albeit without theoretical guarantees.

Coresets: The problem of KDE has mostly been investigated in the context of *coresets*. The first theoretical work we are aware of is that of Phillips [29]. Given a kernel k and a set $P \subset \mathbb{R}^d$, a subset $S \subset P$ is an ϵ-coreset if $|\mathrm{KDE}_P(x) - \mathrm{KDE}_S(x)| \leq \epsilon$ for all $x \in \mathbb{R}^d$. Phillips uses techniques from discrepancy theory to show that one can construct an ϵ-coreset of size $O\left(1/\epsilon^{2-\frac{4}{d+2}} \log^{1-\frac{2}{d+2}}(1/\epsilon\delta)\right)$ with probability at least $1-\delta$. For large d the bound deteriorates and becomes similar to what one gets by simple random sampling $O(\frac{1}{\epsilon^2}\log(1/\delta))$ which is known to be tight. For relative error, recent work [37] uses random sampling to give a similar guarantee that roughly requires $O(\frac{1}{\epsilon^2}\frac{1}{\mu})$ samples. Our lower bound against the $(m, w, 1)$-cell probe models encompasses algorithms of this kind, and shows that any set S (not necessarily a subset of P) that can be used to answer the (μ, ϵ, δ)-problem must have size at least $\Omega(\frac{1}{\mu})$. The implication is that in terms of constructing a *coreset* for KDE in high dimensions, Random Sampling is essentially optimal.

Kernel Matrix Approximation: A closely related idea to that of coresets, is that of Nyström approximation. In this method, given a kernel matrix K a set of s columns (points) is selected and subsequently K is projected on their span. Musco and Musco [27] propose a method based on recursive leverage-score sampling that, using $s = \Theta(k \log k)$ points, $\tilde{O}(nk^2)$ time and $\tilde{O}(nk)$ space, outputs a matrix $\tilde{K}$ such that $\|\tilde{K} - K\|_2 \leq \lambda$ with $\lambda = \frac{1}{k}\sum_{i=k+1}^{n} \sigma_i(K)$. One can use this algorithm to obtain an approximate Kenrel-Matrix Vector Multiplication algorithm $\hat{y} \approx Kz$. For kernel matrices of large rank, like the ones corresponding to the equilateral metric of $r = \sqrt{n} = \tau^{-1}$ clusters consisting of $\sqrt{n}$ identical points, their algorithm requires $\tilde{O}(n^2)$ time and space $\tilde{O}(n^{3/2})$ whereas our algorithm requires $\tilde{O}(n^{\frac{5}{4}})$ time and space for the Exponential and Polynomial kernel, and $O(n^{\frac{5}{4}+o(1)})$ for the Gaussian kernel. However, the algorithm of Musco and Musco applies to any kernel and gives guarantees that hold simultaneously for any test vector z.

D. Open questions

Our work leaves open a few intriguing directions.

Data-dependent Hashing: There is a recent line of work [6], [5] that designs data-dependent LSH schemes that are optimal within a certain class of hashing-based algorithms. We believe that modifications of such schemes can be used for the purposes of KDE.

Batch Setting: The study of the offline or batch setting for Nearest Neighbor Search has received renewed interest over the past years and has brought a wealth of techniques into light [35], [2]. Given the connection between KDE and the Nearest Neighbor Search problem, it would be of practical and theoretical interest to design data-structures that offer provable speedups for the offline setting of the KDE problem. In the regime where $\epsilon = \exp(-\omega(\log^2(n)))$ there is recent work [9] that shows that under SETH no significant improvements can be made beyond quadratic time.

Applications of HBE: It would be interesting to explore other extensions of HBE, e.g. get theoretical guarantees for estimating the *partition function* of log-linear models [34] or analyze the performance of multi-probe schemes.

II. Preliminaries

Notation: For a vector $x \in \mathbb{R}^n$ let $\|x\|_p^p := \sum_{i=1}^n |x_i|^p$ for $p \geq 1$ denote the ℓ_p-norm. For a strictly positive vector $w \in \mathbb{R}^n_{++}$, we denote by $\|x\|_{w,1} := \sum_i w_i|x_i|$ the weighted ℓ_1-norm. Given a number $\tau > 0$ let $(x_i)_\tau := \max\{x_i, \tau\}$. Given a probability distribution ν, we write $Y \sim \nu$ to denote that Y is sampled from ν. For a set S, $U(S)$ denotes the uniform distribution over S and $S^{\otimes k}$ denotes the Cartesian product of S with itself $k \geq 1$ times. Similarly, $\nu^{\otimes k}$ denotes the product distribution $\nu \times \ldots \times \nu$. We use $N(0, I_d)$ to denote the standard multivariate normal distribution and $\Phi(\cdot)$ its CCDF. Throughout the paper $P = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ will denote a set of n points from $\mathbb{R}^d$ and we shall assume that $\mathrm{diam}(P \cup \{x\}) \leq R$ for any query $x \in \mathbb{R}^d$ where $R = R(n) \geq 1$.

A. Unbiased Estimators and Median-of-Means

For $\epsilon, \delta > 0$, given a query point $x \in \mathbb{R}^d$ and weights $w_1(x), \ldots, w_n(x)$ induced by a set P, our goal is to estimate $\mu(x) := \frac{1}{n}\sum_{i=1}^n w_i(x)$ within a multiplicative $(1 \pm \epsilon)$ accuracy with probability at least $1 - \delta$. When it is clear from the context we will often drop the dependence on x and simply write w_i and μ. An estimator $Z \sim \nu$ is called *unbiased* for our problem if $\mathbb{E}[Z] = \mu$. The basis of our approach is to design an unbiased estimator that has small variance relative to μ.

Definition 4. *Given a non-increasing function $V : \mathbb{R} \to \mathbb{R}_+$, we call an unbiased estimator Z, V-bounded if $\mathbb{E}[Z^2] \leq \mu^2 \cdot V(\mu)$ and $\mu^2 V(\mu)$ is non-decreasing.*

The function V is intimately related to the number of samples need to estimate μ and is often referred to as (a bound on) the *relative variance*. The principal

approach to use such an estimator is the Median-of-Means (MoM) technique [3], that allows one to get an estimate $Z_{\epsilon,\delta}$ such that $\mathbb{P}\left[|Z_{\epsilon,\delta} - \mu| \geq \epsilon \cdot \mu\right] \leq \delta$ using $O(\frac{1}{\epsilon^2} V(\mu) \log(\frac{1}{\delta}))$ samples.

Algorithm 1 Median-of-Means (MoM)

1: **Input:** Estimator $Z \sim \nu$, $V \geq 0$, accuracy $\epsilon \in (0,1)$, success prob. $\delta \in (0,1)$.
2: $m(\epsilon, V) \leftarrow \lceil \frac{4}{\epsilon^2} V \rceil$, $L(\delta) \leftarrow \lceil 3\log(1/\delta) \rceil$.
3: $Z_j^{(i)} \overset{iid}{\sim} \nu$ for $j = 1, \ldots, m$, and $i = 1, \ldots, L$.
4: $Z^{(i)} \leftarrow \text{mean}\{Z_1^{(i)}, \ldots, Z_m^{(i)}\}$ for $i = 1, \ldots, L$.
5: **Output:** $Z_{\epsilon,\delta} \leftarrow \text{median}\{Z^{(1)}, \ldots, Z^{(L)}\}$

Most of our effort in this paper goes into obtaining efficient V-bounded unbiased estimators with V being of the form $V(\mu) = n^{o(1)} \cdot \mu^{-\Delta}$ for some $0 \leq \Delta \leq 2$. Once we have such an estimator, we will be able to combine it with the MoM technique and eventually get an estimation algorithm. Hence, the main challenge is bounding the variance.

B. Bounding the Variance

Given an unbiased estimator Z in order to bound the variance we first obtain a simple data-dependent upper bound on $\mathbb{E}[Z^2] \leq F(w,P)$ and then for a class of datasets $\mathcal{P}$, we aim to show that $\sup_{P \in \mathcal{P}} \{F(w,P) | \sum_i w_i = \mu\} \leq \mu^2 V(\mu)$. As a warm up, we present two simple unbiased estimators for the Kernel Density Problem and bound their variance. Let $Y \sim U[P]$, the *random sampling* estimator is given by $Z_{\text{RS}}(x) = k(x, Y)$. Let $\theta \sim U[0, \pi]$ and $g \sim N(0, I_d)$, the *Random Fourier Features* [30] estimator is given by $Z_{RFF}(x) = \frac{2}{|P|} \sum_{y \in P} \left(\cos(g^\top x + \theta) \cos(g^\top y + \theta)\right)$.

Proposition 1. *The RS and RFF estimators are unbiased and satisfy respectively* $\mathbb{E}[Z_{RS}^2] \leq \mu^2 \cdot \mu^{-1}$ *and* $\mathbb{E}[Z_{RFF}^2] \leq \mu^2 \cdot 4\mu^{-2}$. *Moreover, the bounds are tight up to constants in the worst case.*

Proof Sketch: The fact that the RS estimator is unbiased is trivial, whereas the fact that RFF is unbiased was shown by Rahimi and Recht [30] and follows from Bochner's theorem and trigonometric identities. We next bound the second moment

$$\mathbb{E}[Z_{RS}^2] \leq \max_{y \in P}\{k(x,y)\} \cdot \frac{1}{|P|} \sum_{y \in P} K(x,y) \leq \mu^2 \cdot \mu^{-1}$$

$$\mathbb{E}[Z_{RFF}^2] \leq \left(\frac{2}{|P|} \sum_{y \in P} 1\right)^2 = 4 \leq \mu^2 \cdot 4\mu^{-2}$$

To see that these bounds are tight up to constants consider for the RS estimator a dataset with $n\mu$ points located at x and the rest $n(1-\mu)$ points at distance $\sqrt{\log(1/\mu)}$ from x. For the RFF estimator the worst case is when all points are all located at the same point y_0 at distance $\sqrt{\log(1/\mu)}$. ∎

The result on Random Sampling shows that KDE problem is solvable in time $O\left(\min\left\{\frac{1}{\epsilon^2}\frac{1}{\mu}\log(\frac{1}{\delta}), n\right\}\right)$. Despite the simplicity of the above result, some salient features of the problem are revealed. Firstly, the quality of the initial upper bound $F(w,P)$ can differ dramatically between two different estimators. Secondly, despite the simplicitly of the analysis, often the resulting bounds are tight up to constants. Lastly, in both cases we used Hölder's inequality to get the bound. This is going to be a general theme as behind all our bounds on the variance are increasingly sophisticated consequences of Hölder's inequality. We state below the two main inequalities used to bound the variance of *Hashing-Based-Estimators* that we introduce in the next section.

Lemma 1 (2-sided Hölder)**.** $\forall x \in \mathbb{R}^n,\ u, v \in \mathbb{R}^n_{++}$, $|\sum_{ij} A_{ij} x_i x_j| \leq \|x\|_{u,1} \|x\|_{v,1} \cdot \max_{ij}\left\{\frac{|A_{ij}|}{u_i v_j}\right\}$.

Lemma 2 (Monotone Hölder)**.** $\forall n \geq 1,\ \beta \in [\frac{1}{2}, 1]$*, and* $\forall x \in \mathbb{R}^n$ *such that* $|x_1| \geq |x_2| \geq \ldots \geq |x_n|$ *, we have* $\sum |x_i|^{\frac{2-\beta}{\beta}} \left(i + \sum_{j>i} \frac{|x_j|}{|x_i|}\right) \leq n^\beta \cdot \left(\sum_{i=1}^n |x_i|^{\frac{1}{\beta}}\right)^{2-\beta}$ *with equality holding for* $x^* = c\mathbf{1}$ *for any* $c \neq 0$.

III. Importance Sampling through Hashing Based Estimators

Given a distribution ν over a collection of hash functions $\mathcal{H}$, let $h \in \mathcal{H}$ be an element sampled according to ν. For a point $x \in \mathbb{R}^d$ let $H(x) := \{y \in P : h(y) = h(x)\}$ be the set of elements in P that have the same hash value as x. Also, let $I(x)$ be a uniform random element out of $H(x)$ or $\perp$ if $H(x)$ is empty. Further, define $p_i := \mathbb{P}[i \in H(x)]$ and $\tilde{P} := \{i \in P | p_i > 0\}$. We also set $p_\perp := 1$ and $w_\perp := 0$. An $(\mathcal{H}, \nu)$-hashing based estimator (HBE) is defined as $Z_h = Z_h(x) := \frac{w_{I(x)}}{p_{I(x)}} \cdot \frac{|H(x)|}{n}$.

Lemma 3 (Moments)**.** *Let* Z_h *be a* $(\mathcal{H}, \nu)$*-HBE and* P *a set of points. Let* $p_1 \geq p_2 \geq \ldots \geq p_n$*, then*

$$\mathbb{E}[Z_h] = \frac{1}{n} \sum_{i \in \tilde{P}} w_i \tag{3}$$

$$\mathbb{E}[Z_h^2] \leq \frac{1}{n^2} \sum_{i \in \tilde{P}} \frac{w_i^2}{p_i} \left(i + \sum_{j>i} \frac{p_j}{p_i}\right) \tag{4}$$

The proof is straightforward and is based on applying Bayes rule to condition on the random variable $|H(x)|$. Observe that the estimator is unbiased iff $w_i > 0 \Rightarrow$

$p_i > 0$ for all $i \in [n]$. The quantity in the parenthesis plays the role of $F(w,P)$ and expresses a pessimistic upper bound on $\mathbb{E}\left[|H(x)|\middle|i \in H(x)\right]$. This is perhaps the single most important aspect of the method in that it allows us to *express the variance purely in terms of known collision probabilities* p_i that are amenable to analytic manipulations. Theorem 1 provides a characterization of HBE under worst case assumptions

Proof of Theorem 1: Fix n distinct weights, that might be considered. Without loss of generality we assume that the weights are in decreasing order. Let f_i be the fraction of points that are assigned to weight i. We have the following:

$$\begin{aligned}\mathbb{E}[Z_h^2] &\leq \frac{1}{n^2}\sum_i (f_i n)\frac{w_i^2}{p_i}\left(n\sum_{j\leq i} f_j + \frac{1}{p_i}\sum_{j>i} p_j(nf_j)\right)\\ &= \sum_{i,j} f_i f_j \left(\frac{w_i^2}{p_i}\mathbb{I}_{j\leq i} + \mathbb{I}_{j>i}\frac{w_i^2}{p_i^2}p_j\right)\\ &\leq \sup_{\substack{\|f\|_{w,1}\leq \mu\\ \|f\|_1 \leq 1}} \left\{f^\top A f\right\}\end{aligned}$$

where in the last step we set $A_{ij} := \frac{w_i^2}{p_i}\mathbb{I}_{j\leq i} + \mathbb{I}_{j>i}\frac{w_i^2}{p_i^2}p_j$. Let $k(\mu)$ be the largest index i such that $w_i \leq \mu$, to obtain an upper bound, we split the weights into two sets $S_\leq := \{i \leq k\}$ and $S_> := \{i > k\}$. We define accordingly $x_\leq := (x_1, \ldots, x_k)$ and $x_> := (x_{k+1}, \ldots, x_n)$ as well as the matrices $A_{\leq\leq}, A_{\leq>}, A_{>\leq}, A_{>>}$ in the natural way. We have that:

$$\begin{aligned}f^\top A f = f_\leq^\top A_{\leq\leq} f_\leq + f_\leq^\top A_{\leq>} f_> \\ + f_>^\top A_{>\leq} f_\leq + f_>^\top A_{>>} f_>\end{aligned}$$

We next compute the supremum of the above quantity separately for each term. The main idea is to always enforce the tightest set of constraints. For instance, for all indices with weights at least μ the constraint on $\|f\|_{w,1}$ is the tightest. Using the two-sided Hölder inequality (Lemma 1) appropriately we get:

$$\begin{aligned}\mathbb{E}[Z_h^2] \leq \mu^2 \max_{i,j\leq k}\left\{\frac{A_{ij}}{w_i w_j}\right\} + \mu\cdot\max_{i\leq k, j>k}\left\{\frac{A_{ij}}{w_i}\right\}\\ + \mu\cdot\max_{i>k, j\leq k}\left\{\frac{A_{ij}}{w_j}\right\} + \max_{i>k,j>k}\{A_{ij}\}\end{aligned}$$

Let us define the diagonal matrix $D := D(w,\mu)$ as $D_{ii} = \begin{cases}\frac{\mu}{w_i}, & \text{for } i \leq k(\mu)\\ 1, & \text{for } i > k(\mu)\end{cases}$. Then, we have the following bound $\mathbb{E}[Z_h^2] \leq 4\sup_{ij}\left\{e_i^T DAD e_j\right\}$. ∎

The theorem assumes nothing about HBE besides unbiasedness and therefore is applicable in an arbitrary setting. If more structure is assumed stronger statements can be made. We give a refined analysis of *scale-free* estimators based on whether a large fraction of the mean comes from points with relatively large weights.

Definition 5. *For a query x and $\tau \in [\mu, 1]$ let $B_{\tau,\mu}(x) := \{i \in P | w_i \geq \frac{\mu}{\tau}\}$. For $\gamma \in [0,1]$ the query is said to be* (τ,γ)-localized *if $\sum_{i\in B_{\tau,\mu}} w_i \geq (1-\gamma)\sum_i w_i$.*

Theorem 5 (Restatement of Theorem 2)**.** *Let Z_h be a (β, M)-*scale free *estimator with $\beta \in [1/2, 1]$. For every (τ,γ)-localized query x,*

$$\mathbb{E}[Z_h^2] \leq \mu^2 \cdot M^3\left\{2\tau^\beta + \gamma^{2-\beta} + \tau^{2\beta-1}\gamma^\beta\right\}\mu^{-\beta}$$

The optimal choice of β in the case where no assumptions are made ($\gamma = \tau = 1$) is $\beta^* = \frac{1}{2}$. Even if β is not selected depending on the structure of the query, significant gains can be obtained, in cases where the parameters τ, γ are small enough. As an example, for $\beta = \frac{1}{2}$, $\tau = \mu^{\frac{1}{2}}$, $\gamma = \mu^\delta$ the theorem implies that the variance is bounded by $\mu^2 \cdot 4M^3\mu^{-\Delta(\delta)}$ where $\Delta(\delta) = \max\{\frac{1}{4} - \delta, \frac{1-3\delta}{2}\} < \frac{1}{2}$ for all $\delta > 0$.

Proof Sketch: We start by using the scale-free property and (4):

$$\mathbb{E}[Z_h^2] \leq M^3 \cdot \frac{1}{n^2}\sum_{i=1}^n w_i^{2-\beta}\left(i + \sum_{j>i}\left(\frac{w_j}{w_i}\right)^\beta\right)$$

Let A be the summation term, we break up the terms according to $B_{\tau,\mu}$. Let J be the maximal index in $B_{\tau,\mu}$:

$$\begin{aligned}A = \sum_{i\leq J} w_i^{2-\beta}(i + \sum_{J\geq j>i}(\frac{w_j}{w_i})^\beta) + \sum_{i\leq J} w_i^{2-2\beta}\cdot\sum_{j>J} w_j^\beta\\ + \sum_{i>J} w_i^{2-\beta}(i - J + \sum_{j>i}(\frac{w_j}{w_i})^\beta) + |B_{\tau,\mu}|\sum_{i>J} w_i^{2-\beta}\end{aligned}$$

We next use Hölder-type inequalities to bound each term. For the first two terms we invoke Lemma 2 and use the fact that $|B_{\tau,\mu}| \leq \tau n$. To bound the remaining terms, we use the following consequences of Hölder's inequality $\|x\|_\beta^\beta \leq \|x\|_1^\beta \cdot n^{1-\beta}$ and $\|x\|_p^p \leq \|x\|_q^q \cdot \|x\|_\infty^{p-q}$ for all $\beta \in [0,1]$, $p \geq q > 0$, $x \in \mathbb{R}^n$. By combining all the inequalities, we get $A \leq n^2\mu^{2-\beta}(\tau^\beta + \gamma^{2-\beta} + \tau^{2\beta-1}\gamma^\beta + \tau^\beta\gamma)$. ∎

This concludes the presentation of the general framework of Hashing-Based-Estimators, that given a set of weights $\{w_i\}$ and collisions probabilities $\{p_i\}$ computes a function V such that the resulting HBE is V-bounded.

IV. Adaptive Estimation through Mean Relaxation

Our goal in this section is, given a dataset $P \in \mathbb{R}^d$ and a V-bounded unbiased estimator, to build a data-structure that can efficiently approximate the mean

$\mu(x)$ for a given query $x \in \mathbb{R}^d$. Thus, presenting a complete algorithmic framework that can be instantiated for different problems.

We first address the problem of obtaining a constant factor approximation to the mean. We exploit two facts: (i) monotonicity of the variance in terms of μ, (ii) concentration of measure. Monotonicity suggests that we can start with an over estimate of the mean and keep refining it until we come very close to the truth. Concentration of measure allows us to come up with a simple consistency check that recognizes when our estimate is close enough to the query. Based on this we propose the following adaptive algorithm for which we get strong guarantees.

Algorithm 2 Adaptive Mean Relaxation (AMR)

Input: V-bounded unbiased estimator $Z \sim \nu$, query $x \in \mathbb{R}^d$, accuracy $\alpha \in (0, 1]$, threshold $\tau \in (0, 1)$, failure prob. $\chi \in (0, 1)$.
$\epsilon \leftarrow \frac{2}{7}\alpha, c \leftarrow \frac{\epsilon}{2}, \gamma \leftarrow \frac{\epsilon}{7}$, $\delta \leftarrow \frac{2\alpha}{49\log(1/\tau)}\chi$, $i \leftarrow -1$.
repeat
  $i \leftarrow i + 1$, $\mu_i \leftarrow (1-\gamma)^i$,
  $Z_i \leftarrow \mathrm{MoM}_{\frac{\epsilon}{3},\delta}(\nu, V(\mu_i))$
until $|Z_i - \mu_i| \le c \cdot \mu_i$ or $i > \frac{7^2\log(1/\tau)}{2\alpha}$.
Output: if $i \le \frac{7^2\log(1/\tau)}{2\alpha}$ return Z_i else return 0.

Lemma 4. *Given a V-bounded estimator $Z \sim \nu$, let $Z_i := \mathrm{MoM}_{\frac{\epsilon}{3}}(\nu, V(\mu_i))$, with $\mu_i \in [0, 1]$ then*

(i) *for all i such that $\mu_i \ge (1-(c+\epsilon))^{-1}\mu$, it holds that $\mathbb{P}[|Z_i - \mu_i| \le c \cdot \mu_i] \le \delta$.*
(ii) *for all $\frac{1+\frac{\epsilon}{3}}{1+c}\mu \le \mu_i \le \mu$, it holds that $\mathbb{P}[|Z_i - \mu_i| \le c \cdot \mu_i] \ge 1 - \delta$.*

In fact, there is no need for the samples used in different calls of the MoM routine to be independent and we can implement the MoM routing by keeping $L(\delta) = \lceil 3\log(1/\delta)\rceil$ running sums. We call the resulting algorithm AMR*.

Theorem 6 (Mean Relaxation)**.** *Let $Z \sim \nu$ be a V-bounded estimator with $\mathbb{E}[Z] = \mu \in [0, 1]$ and $\hat{Z}$ be the output of the AMR* algorithm with parameters (α, τ, χ). If $\mu \ge \tau$ then $\mathbb{P}[|\hat{Z} - \mu| \le \alpha\mu] \ge 1 - \chi$ otherwise if $\mu < \tau$, $\mathbb{P}[\hat{Z} = 0] \ge 1 - \chi$. The total number of samples used is bounded by $O(\alpha^{-3}\log(1/\chi)) \cdot V((\mu)_\tau)$.*

With this algorithm in hand we are ready to state our main result in the abstract setting of HBE.

Theorem 7 (Main Result)**.** *Given a V-bounded HBE with complexity T , there exists a data structure that can answer any query in time $O(\frac{1}{\epsilon^2}V((\mu)_\tau)\log(\frac{1}{\chi})T)$ using space $O(\frac{1}{\epsilon^2}V(\tau)\log(\frac{1}{\chi}) \cdot nT)$ with success probability at least $1 - \chi$.*

Proof: We begin by describing the preprocessing phase. We sample $N = O(\log(1/\chi)\frac{1}{\epsilon^2}V(\tau))$ hash functions $h_1, \ldots, h_N \overset{i.i.d.}{\sim} \nu$ from $\mathcal{H}$ and evaluate them on the dataset P. This can be done in $NT \cdot n$ time and space. The query algorithm interacts with the data-structure by making calls to hash functions. The data-structure always keeps the index of the last hash function called and increments it in a cyclic fashion after each call, thus for a given query it never evaluates the same hash function twice and the samples obtained are independent. When the query arrives, the query algorithm first runs a stage of the adaptive mean relaxation algorithm with $\alpha = 1$ and probability $\chi/2$. Every time a sample is needed a call is made to the data-structure. After $O(V((\mu)_\tau\log(1/\chi)))$ calls with probability at least $1 - \chi/2$ we either have a constant factor approximation or we know that $\mu < \tau$ (if AMR* outputs 0). In the first case, we apply one level of MoM algorithm using an underestimate of μ that uses $O(\frac{1}{\epsilon^2}\log(1/\chi)V((\mu)_\tau))$ more calls and gets a $(1 \pm \epsilon)$ multiplicative approximation with probability at least $1 - \chi$. In the latter, case we simply output 0. ■

The proof of Theorem 4 follows by invoking Theorem 7 for specific V-bounded estimators that we derive in the next two sections.

V. KDE through Euclidean LSH

In this section, we instantiate the framework of HBE for the problem of KDE using Euclidean LSH of Datar et al. [11]. At a high level, given a kernel k, the goal is to design a hashing scheme such that the probability that two points hash at the same bucket is as similar as possible to $k(x, y)$. We consider three such kernels: the Exponential, the Generalized t-Student (polynomial) and the Gaussian kernel. We show that this specific LSH scheme can be used to construct scale-free estimators for the first two, while for the Gaussian case, although we cannot get a scale-free estimator, we are still able to analyze the variance of the estimator using Theorem 1. The family of hash functions is given by:

$$\mathcal{H}_1(w) := \left\{ h(x) = \left\lceil \frac{g^\top x + \beta}{w} \right\rceil \middle| \, g \in \mathbb{R}^d, \beta \in [0, w] \right\}$$

for some fixed $w > 0$. We define a distribution ν_1 over $\mathcal{H}_1$ by sampling $g \sim \mathcal{N}(0, I_d)$ and $\beta \sim U[0, w]$. The important quantity to control is the collision probability $p_1(c) := \mathbb{P}_{h\sim\nu}[h(x) = h(y)]$ of two points x, y at

distance $\|x-y\| = cw$ and is given by [11]:

$$p_1(c) = 1 - 2\Phi(c^{-1}) - \sqrt{\frac{2}{\pi}}c\left(1 - \exp\{-\frac{c^{-2}}{2}\}\right) \quad (5)$$

Lemma 5 (Pointwise bounds)**.** *For all* $c > 0$

$$p_1(c) = \sqrt{\frac{2}{\pi}}\sum_{k=0}^{\infty}\frac{(-1)^k}{2^k k!(2k+2)(2k+1)}\frac{1}{c^{2k+1}} \quad (6)$$

while for $\delta \leq \frac{1}{2}$ *and* $c \leq \min\left\{\delta, \frac{1}{\sqrt{2\ln(1/\delta)}}\right\}$

$$e^{-\sqrt{\frac{2}{\pi}}(1+\delta)\cdot c} \leq p_1(c) \leq e^{-\sqrt{\frac{2}{\pi}}(1-\delta^3)\cdot c} \quad (7)$$

We see that this family gives bounds exponentially decreasing with the distance. This leads naturally to our first result for this scheme.

Theorem 8 (Exponential Kernel)**.** *For* $\beta \in (0,1]$ *there exists a* $(\beta, \sqrt{e})$*-scale free HBE for the exponential kernel* $e^{-\|x-y\|}$ *that has complexity* $O(dR^2)$*.*

Proof: Set $D = 3\lceil R^2\rceil$, $w = \frac{D}{\beta\sqrt{\frac{\pi}{2}}}$ and consider the HBE resulting from the family $\mathcal{H}_1^{\otimes D}(w)$ with probability measure $\nu_1^{\otimes D}$. We first see that for a pair of points that are distance $\|x-y\| = r$ apart, we have that $c = \frac{r}{w} = \beta\sqrt{\frac{\pi}{2}}\frac{r}{3\lceil R^2\rceil} \leq \frac{\beta\sqrt{\frac{\pi}{2}}}{3R} \leq \sqrt{\frac{\pi}{18}} \leq \frac{1}{2}$. Where we used the fact the maximum distance of interest is bounded by $R \geq 1$. Additionally, for these parameters it holds that $e^{-\sqrt{\frac{2}{\pi}}cD} = e^{-\beta\cdot r}$. Using the second part of Lemma 5 with $\delta = \frac{\beta\sqrt{\frac{\pi}{2}}}{3R}$ we get

$$e^{-\sqrt{\frac{2}{\pi}}(\frac{\beta\sqrt{\frac{\pi}{2}}}{3R})^2 D} \leq \frac{p_1^D(c)}{e^{-\beta\cdot r}} \leq e^{+\sqrt{\frac{2}{\pi}}(\frac{\beta\sqrt{\frac{\pi}{2}}}{3R})^4 D} \quad (8)$$

Which gives us that $\frac{1}{\sqrt{e}}\cdot e^{-\beta r} \leq p_1^D(c) \leq \sqrt{e}\cdot e^{-\beta r}$. To implement the estimator we require $O(R^2)$ hash functions that each can be evaluate in time $O(d)$ and requires space $O(d+n)$. ∎

Although, this is to be expected given the derived pointwise bounds on the collision probabilities, and seems to suggest a correspondence between kernels and hashing schemes, we show next that one can construct a *scale-free estimator for another very different kernel using the same LSH scheme*.

Theorem 9 (Generalized t-Student Kernel)**.** *For integers* $p, q \geq 1$*, there exists a* $(\frac{q}{p}, 3^q)$*-scale free HBE for the kernel* $\frac{1}{1+\|x-y\|^p}$ *that has complexity* $T = O(dp)$*.*

Proof: Set $w = \sqrt{2\pi}$ and consider the HBE resulting from the family $\mathcal{H}_1^{\otimes p}(w)$. We obtain bounds for the collision probabilities for two points at distance $\|x-y\| = r = c\cdot w$ apart. For $c > 1$, using the first part of Lemma 5 and dropping terms appropriately

$$\frac{1}{\sqrt{2\pi}}(1 - \frac{1}{12c^2})\frac{1}{c} \leq p_1(c) \leq \frac{1}{\sqrt{2\pi}}\frac{1}{c} \quad (9)$$

Setting $\beta = \frac{q}{p}$ and raising to the q-th power we get that for all $c > 1$:

$$\left(\frac{11}{12}\right)^q \frac{(1+r^p)^\beta}{r^q} \leq \frac{p_1^q(c)}{\frac{1}{(1+r^p)^\beta}} \leq \frac{(1+r^p)^\beta}{r^q} \quad (10)$$

whereas for $c \leq 1$ we have from (9) and monotonicity of $p_1(c)$:

$$\left(\frac{11}{12\sqrt{2\pi}}\right)^q (1+r^p)^\beta \leq \frac{p_1^q(c)}{\frac{1}{(1+r^p)^\beta}} \leq (1+r^p)^\beta \quad (11)$$

Since $2^{-1} \geq 11/(12\sqrt{2\pi}) \geq 3^{-1}$ this shows that the resulting estimator is $(\frac{q}{p}, 3^q)$*-scale-free* for the Generalized t-student Kernel for all distances. ∎

These two results combined with Theorem 5 and Theorem 7 can be used to construct a data structure for the KDE problem under the exponential or t-Student kernel. More importantly, this is done effortlessly by appealing to the the general case of HBE and only required showing that our hashing schemes produces scale-free estimators.

We show next that the framework of HBE can be useful beyond the ideal scenarios where a scale-free estimator can be derived. We do so by showing that one can use the exponential drop-off of the collision probabilities to simulate the Gaussian kernel and then appeal to the more general Theorem 1 to bound its variance.

Theorem 10. *For any* $t \in [1, R]$ *there exists a HBE* Z_t *for the Gaussian kernel* $e^{-\|x-y\|^2}$ *with* $\mathbb{E}[Z_t^2] \leq \mu^2 \cdot 4e^{\frac{3}{2}}\mu^{-\gamma^2+\gamma-1}$ *where* $\gamma(t,\mu) := t/\sqrt{\log(1/\mu)}$*, that has complexity* $T = O(dt^2R^2)$*.*

Proof Sketch: Fix $t \geq 1$ and set $D = 3\lceil(tR)^2\rceil$, $w = D\sqrt{\frac{2}{\pi}}$. We consider once more the HBE resulting from $\mathcal{H}_1^D(w)$ and probability measure $v_1^{\otimes D}$. For a pair of points at distance $r = c\cdot w \leq R$ we have that $c \leq \delta = \frac{R}{w} \leq \frac{R}{3t^2R^2}\sqrt{\frac{\pi}{2}} \leq \frac{1}{2}$. By utilizing Lemma 5 one again we obtain $\frac{1}{\sqrt{e}}\cdot e^{-r\cdot t} \leq p_1^D(c) \leq \sqrt{e}\cdot e^{-r\cdot}$. To bound the variance due to Theorem 1 we only need to consider what happens for datasets supported only on two points. It will be useful to consider that one point is at distance $r_1 = \sqrt{\alpha\log(1/\mu)}$ away from the query and the other at $r_2 = \sqrt{\alpha'\log(1/\mu)}$ with $0 \leq \alpha \leq \alpha'$. Setting $\gamma(t,\mu) = \frac{t}{\sqrt{\log(1/\mu)}}$ the pointwise bounds

become

$$\frac{1}{\sqrt{e}} \cdot \mu^{\gamma\sqrt{\alpha}} \leq p_1^D(c) \leq \sqrt{e} \cdot \mu^{\gamma\sqrt{\alpha}} \tag{12}$$

A case analysis of the upper bound given by Theorem 1 reveals that the worst-case distances of the two points are given by picking $\alpha = \gamma^2$ and $\alpha^{'} = 1$. For which, the upper bound on the second moment becomes $\mu^2 \cdot 4e^{\frac{3}{2}} \cdot \mu^{-\gamma^2+\gamma-1}$. ∎

Even though that we were not able to get the ideal behavior using this hashing scheme for the Gaussian Kernel, still the estimator has improved variance compared to random sampling for all $0 < \gamma < 1$ and achieves it's best performance when $t = \frac{1}{2}\sqrt{\log(1/\mu)} \Rightarrow \gamma = \frac{1}{2}$.

VI. Scale-free estimator based on Andoni-Indyk LSH

Our framework of scale-free esimators is a natural desideratum when trying to approximate the ideal function for importance sampling. In this section, we show how to use the "Ball-Carving" LSH introduced by Andoni-Indyk [4] for Euclidean distance to get a *scale-free* estimator for the Gaussian Kernel.

Andoni and Indyk [4] introduced a family of hash functions $\mathcal{H}_t(w)$ parametrized by an integer $t \geq 2$, and a width $w > 0$ such that the evaluation cost is bounded by $U_t = d2^{O(t \log t)} \log n$ and space usage by $O(U_t + dn)$. Essentially their scheme partitions the space by randomly projecting into t-dimensions and then carving out balls of radius w centered at random points. We refer to the resulting probability measure as ν_t. Using a similar analysis as the one in [4] we show the following bounds on the collision probability $p_t(c) := \mathbb{P}_{h\sim\nu_t}[h(x) = h(y)]$ of two points x, y at distance cw.

Lemma 6 (Pointwise Bounds)**.** *The function $p_t(c)$ is non-increasing for all $c \geq 0, t \geq 1$. Furthermore, for all $t \geq 12$ and $\frac{16}{t} \leq c^2 \leq 1$*

$$p_t(c) \geq \frac{1}{4\sqrt{t}}\frac{1}{c}(1 - 2e^{-\frac{t}{4}})e^{-\frac{t-1}{8}\frac{1}{2-c^2}c^4}e^{-\frac{t-1}{8}c^2} \tag{13}$$

$$p_t(c) \leq \frac{3}{\sqrt{t}}\frac{1}{c}\left(1 + \frac{\sqrt{t}c}{3}e^{-\frac{9}{64}t}\right)e^{\frac{t-1}{8^2}c^4}e^{-\frac{t-1}{8}c^2} \tag{14}$$

Observe that for small $c = O(1/\mathrm{poly}(\log n))$ and large enough $t = \mathrm{poly}(\log n)$, the dominant term is $e^{-\frac{t-1}{8}c^2}$ and drops exponentially with the squared of the distance as desired. However, in order for the time U_t to compute the hash function on query to be $n^{o(1)}$, we must have $t = o(\log(n))$. This is the main bottleneck that complicates a bit the application of the Andoni-Indyk hashing scheme for our purposes.

Theorem 11 (Gaussian Kernel)**.** *For all $\beta \in (0, 1]$ there exists a $(\beta, e^{O(R^{\frac{4}{3}} \log\log n)})$-scale free HBE for the Gaussian kernel $e^{-\|x-y\|^2}$ that has complexity $T = e^{O(R^{\frac{4}{3}} \log\log n)}$.*

Proof: Let $t := \max\{R^{\frac{4}{3}}, 12\}$, $w = \sqrt[4]{t}R, D = \lceil \frac{8w^2}{t-1}\beta \rceil$ and consider the HBE resulting from using the family $\mathcal{H}_t^D(w)$. The HBE can be evaluated in time $D2^{O(t \log\log(n))}$ and takes space $D2^{O(t \log\log(n))} \cdot n$. Next, we will show that for this selection of parameters we get a scale-free estimator for the Gaussian kernel. Consider two points at distance $r = c \cdot w$. First, we see that $c = \frac{r}{w} \leq \frac{R}{w} \leq \frac{1}{\sqrt[4]{t}} \leq 1$. For distances $r \geq \frac{4}{\sqrt[4]{t}}R \Rightarrow c^2 \geq \frac{16}{t}$, our selection of D results in the dominant term of the bounds given by Lemma 6 being $e^{-\frac{t-1}{8w^2}D \cdot r^2} = e^{-\beta r^2}$. Moreover,

$$e^{-\Theta(D\log(t))}e^{-\beta r^2} \leq p_t^D(c) \leq e^{\Theta(D\log(t))}e^{-\beta r^2} \tag{15}$$

Thus, we see that for the range $4R/\sqrt[4]{t} \leq r \leq R$ we have a scale-free estimator with $M = e^{O(D\log(t))}$. To get a bound for $0 \leq r \leq R/\sqrt[4]{t}$ we use monotonicity of $p_t(c)$ to obtain:

$$M^{-1} \cdot e^{-16\beta\frac{R^2}{\sqrt{t}}} < p_t^D(c) \leq e^{\beta r^2} \cdot e^{-\beta r^2} \tag{16}$$

Since $D = \Theta(R^2/\sqrt{t})$ we see that we have constructed an $(\beta, e^{O(\log\log(n)R^2/\sqrt{t})})$-scale free estimator. Our selection of $t = R^{4/3}$ balances the complexity of evaluating the hashing function with the deviation from the ideal collision probabilities. ∎

The above theorem shows that as long as $R = O(\log^\gamma(n))$ with $0 < \gamma < \frac{3}{4}$ the estimator is $(\beta, n^{o(1)})$-scale free. The regime of most interest is when $\gamma = 1/2$, where polynomially small values of $\mu = n^{-\Omega(1)}$ are permissible. In this case, our estimator is $(\beta, e^{O(\log^{\frac{2}{3}}(n)\log\log n)})$-scale free.

VII. Fast Kernel-Matrix Vector Multiplication

Given a kernel function k and a set of points $P = \{x_1, \ldots, x_n\}$, let $K = \{K(x_i, x_j)\}_{i,j\leq n}$ denote the matrix of the pairwise evaluations of the kernel. Given a vector $z \in \mathbb{R}^n$, the problem of *Approximate Kernel-Matrix Vector Multiplication* is to obtain an approximation $\hat{y}$ to $y^* = Kz$. Due to linearity, we can always rescale the vectors without changing the problem, thus we may assume that $\|z\|_1 = 1$. This problem is important as many machine learning applications involve the multiplication of a vector with a dense Kernel matrix and very often this operation is the computational bottleneck. We show how one can adapt

the techniques from this paper to provide a solution to this problem.

Theorem 12. *Given a V-bounded HBE for a kernel k with complexity T, there exists an algorithm that given a dataset P and a weight vector z can compute a vector $\hat{y}$ in time $\tilde{O}(\frac{1}{\epsilon^2}V(\epsilon\tau)\cdot nT)$ using space $\tilde{O}(\frac{1}{\epsilon^2}V(\epsilon\tau)\cdot nT)$ such that with probability at least $1-n^{-1}$ for all $i \in [n]$ it holds $|\hat{y}_i - y_i^*| \leq 3\epsilon\tau + \epsilon|y_i^*|$ and*

$$\|\hat{y} - y^*\|_p \leq \epsilon \cdot (3\tau n^{1/p} + \|y^*\|_p) \tag{17}$$

Proof Sketch: We may assume that all elements of z are positive otherwise we apply our algorithm to z_+ and z_- separately. Set $\tau^{'} = \epsilon\tau$, $L = \log_2(n/\tau^{'})$ and $\chi = n^{-1}$. We then geometrically partition the vector z in groups $S_1, \ldots, S_L$ such that all elements in each group differ by at most a factor of two. For coordinates that are too small (less than $\tau^{'}/n$) we just ignore them. This can be done with a linear pass over the vector and in this manner we may express our problem as a weighted version of L KDE problems, where the ℓ-th problem asks to compute an approximation to $\mathrm{KDE}^z_{S_\ell}(x) = \frac{1}{Z_\ell}\sum_{i\in S_\ell} k(x, x_i)z_i$ with $Z_\ell = \sum_{i\in S_\ell} z_i$ for all $x \in P$. Given a hashing scheme $\mathcal{H}$ with collision probabilities p_i we evaluate the hash function during the preprocessing step only on S_ℓ and then for a query point $x \in P$ we define the following estimator.

$$Z_{h,\ell} = \frac{z_I}{Z_\ell}\frac{k(x, x_I)}{p_I}|H(x)| \tag{18}$$

where as before I is a random index from $H(x) \subseteq S_\ell$. It is easy to see that his estimator is unbiased and that the variance is at most 4 times larger than if we would be trying to estimate $\mathrm{KDE}_{S_\ell}(x)$. Hence, given any V-bounded HBE for the kernel k and set S_ℓ we can use the above modification to get a $4V$-bounded HBE for $\mathrm{KDE}^z_{S_\ell}(x)$. Invoking Theorem 7 with parameters $(\epsilon, \tau^{'}, \chi/(nL))$ we can get a data structure that can estimate $\mathrm{KDE}^z_{S_\ell}(x_i)$ with probability at least $1 - \frac{\chi}{nL}$ either within multiplicative accuracy ϵ (when AMR* has non-zero output) or with absolute accuracy $\tau^{'}$ (when AMR* outputs 0) for all $i \in [n]$ (by union bound). Let $\mathcal{L}^{'}$ denote the set of indices of $[L]$ such that $Z_\ell \geq \frac{\tau^{'}}{|L|}$. For all $\ell \in \mathcal{L}$ we instantiate the data structure given by Theorem 7 for the set S_ℓ and use it to query all points in P. The overhead per-query of the whole process is at most a multiplicative factor $L = O(\log(n/\tau^{'}))$ compared to the case that we were creating a single data-structure for the same problem. A straightforward analysis of the estimation error of the algorithm givers that for all $i \in [n]$ with probability at least $1-\chi$ it holds $|\hat{y}_i - y_i^*| \leq 3\epsilon\tau + \epsilon|y_i^*|$. Summing over all indices and using triangle inequality gives $\|\hat{y} - y^*\|_p \leq \epsilon \cdot (3\tau n^{1/p} + \|y^*\|_p)$. ∎

VIII. Lower Bound for Kernel Density Estimation

Our upper bounds on the KDE problem seem to suggest that the complexity of the problem should depend inverse polynomially with μ and ϵ. In this section, we give evidence that this type of dependence is necessary. The basic observation that motivates our lower bound is that the Gaussian kernel is rapidly decreasing and can be thought of as being an approximation to the indicator function $\mathbb{I}\left\{\|x - y\|^2 \leq \sigma^2\right\}$. In particular, for two distances $r_1 = \sigma, r_2 > \sqrt{C}\sigma$ the kernel value varies from $e^{-1} = \Omega(1)$ to $e^{-C} = o(1)$ for any $C = \omega(1)$. Thus, our strategy is to reduce the *Approximate Nearest Neighbor Search* (ANNS) problem to the KDE problem.

The (r, c)-ANNS problem asks, for a dataset P and query x in some metric space, to distinguish between two cases: (A) there is a point at distance at most $r > 0$ from the query and (B) all points are at distance at least $c \cdot r$ with $c > 1$. The complexity of ANNS has been a topic of ongoing research over the past two decades. The most popular model of computation to prove lower bounds for is the cell-probe model.

In this model, we are allowed an arbitrary amount of preprocessing but can only store m cells each with w bits of information. The query algorithm then queries (adaptively) t cells and is required to produce the output. We refer to this model as the (m, w, t)-*cell probe model*. For ANNS, lower bounds in this model impose constraints on m, w, t depending on n, c. The most general result in this area is given by the following theorem proved by Panigrahy, Talwar and Wieder [28], whose estimates where improved by Andoni et al. [5].

Theorem 13 ([28], [5])**.** *There exists a distribution over (r, c)-ANNS instances and $\gamma \in [0, 1]$ such that any randomized algorithm in the (m, w, t)-cell probe model which is correct with probability at least a half on these instances, satisfies:*

$$\frac{m^t w}{n} \geq \sup_{(q-1)(p-1)=(1-\frac{1}{c})^2, p,q\geq 1} \left\{(\frac{\gamma}{t})^q m^{t(1+\frac{q}{p}-q)}\right\}$$

The distribution over ANNS instances is defined by picking n points uniformly at random from the d-dimensional boolean hypercube and then generating a query by picking one of the n points and keeping each bit with probability $\rho = 1 - \frac{1}{c}$. We show that for the specific distribution one can use an algorithm for KDE that would solve the corresponding ANNS problem with more than $1/2$ probability.

Theorem 14. *Any algorithm that solves the (μ, ϵ, δ)-KDE problem solves also the hard instances of (r, c)-ANNS with $n = \frac{1}{\mu}$, $d = \Theta(\log^3(n))$, and $c = \Theta(\frac{\log(n/\epsilon)}{\log(1/\epsilon)})$ as long as $\epsilon \in [\frac{1}{n}, \frac{1}{4})$ and $\delta < \frac{1}{2} - 2n^{-1}$.*

Proof Sketch: Let $\epsilon_1 = \epsilon$, $\epsilon_2 = 2\epsilon$, $r = \log(\frac{n}{\sqrt{(1-\epsilon_1)(1-\epsilon_2)}})$. Using this parametrization we define: $\sigma^2 = \frac{9\log(n)}{\log^2\left(\sqrt{\frac{1-\epsilon_1}{1-\epsilon_2}}\right)} \cdot r$, $d = 2\sigma^2 \cdot r$, $u = \sqrt{\frac{6\log(n)}{\sigma^2}}$ and $c = \frac{(1+u)}{\log(1/4\epsilon)} \log(n/\sqrt{\epsilon_1\epsilon_2})$. Using Chernoff bounds we can show that a no-instance of ANNS has KD less than $(1-\epsilon)\mu$ and a yes-instance has KD more than $(1+\epsilon)\mu$ with probability at least $1-2n^{-1}$. We therefore, get that any data structure for the (μ, ϵ, δ)-KDE problem can also solve the c-ANN problem for random instances with probability at least $1 - \delta - 2n^{-1}$. ∎

The quality of the bounds obtained by Theorem 13 deteriorate as t increases. Nevertheless, the bound for $t = 1$ is optimal and captures ANNS algorithms such as ones based on data-independent LSH. In our setting, the $(m, 1, w)$-probe model encompasses an interesting class of estimation algorithms, that we call *adaptive coresets*. Given P, such algorithms may perform an arbitrary amount of preprocessing and store m sets $S_1, \ldots, S_m$ (of arbirary points) each of size at most w/d. Given a query x, the algorithm picks one of those sets $i = i(x) \in [m]$ and produces an approximation to kernel density as a function of (S_i, x). This includes estimates such as $\mathrm{KDE}_{S_i}(x)$ or weighted versions thereof.

Corollary 1. *Any algorithm that solves the (μ, ϵ, δ)-KDE problem in the $(m, 1, w)$-cell probe model must satisfy: $(m \cdot w) \geq e^{-O(1)} \frac{1}{\mu} \left(\frac{1}{4\epsilon}\right)^{2(1-\frac{\log(w)}{\log(\frac{1}{\mu})})(1-o(1))}$.*

Proof Sketch: Using the parameters defined in Theorem 14 where $n = 1/\mu$, we invoke Theorem 13 with $p = \frac{1}{1-\frac{\log(1/\gamma)}{\log(1/w\mu)\zeta}}$, $q = 1 + \frac{\rho^2}{\log(1/\gamma)}(\log(\frac{n}{w})\zeta - \ln(1/\gamma))$ and $\zeta = \frac{1}{\rho^2}\sqrt{\frac{\log(1/\gamma)(1-\rho^2)}{\log(1/w\mu)}}$. The final bound follows by plugging the above parameters and using standard inequalities like $\frac{1}{1-x} \geq 1 + x$. ∎

The above lower bound shows that either we must have many such sets (large space) or each set must be large itself (query time). Of particular interest is the case $m = 1$ where there is a single set. In that case, we have a lower bound on the size of coresets that shows that random sampling has the optimal dependence in terms of μ in the 1-cell probe model.

ACKNOWLEDGMENT

We would like to thank Peter Bailis for valuable conversations that initiated our research on this problem. We are also grateful to the anonymous reviewers for their comments that helped improved the presentation of our results. This research was supported by NSF grants CCF-1617577, CCF-1302518 and a Simons Investigator Award. The second author is partially supported by an Onassis Foundation Scholarship.

REFERENCES

[1] J. Acharya, I. Diakonikolas, J. Li, and L. Schmidt. Sample-optimal density estimation in nearly-linear time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1278–1289. SIAM, 2017.

[2] J. Alman and R. Williams. Probabilistic polynomials and hamming nearest neighbors. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 136–150. IEEE, 2015.

[3] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.

[4] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

[5] A. Andoni, T. Laarhoven, I. Razenshteyn, and E. Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 47–66. SIAM, 2017.

[6] A. Andoni and I. Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 793–801. ACM, 2015.

[7] E. Arias-Castro, D. Mason, and B. Pelletier. On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *Journal of Machine Learning Research*, 2015.

[8] H. Avron, K. L. Clarkson, and D. P. Woodruff. Faster kernel ridge regression using sketching and preconditioning. *arXiv preprint arXiv:1611.03220*, 2016.

[9] A. Backurs, P. Indyk and L. Schmidt. On the Fine-Grained Complexity of Empirical Risk Minimization: Kernel Methods and Neural Networks. *http://arxiv.org/abs/1704.02958*, 2017.

[10] S.-O. Chan, I. Diakonikolas, R. A. Servedio, and X. Sun. Efficient density estimation via piecewise polynomial approximation. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 604–613. ACM, 2014.

[11] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

[12] L. Devroye and G. Lugosi. *Combinatorial methods in density estimation*. Springer Science & Business Media, 2012.

[13] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Robust estimators in high dimensions without the computational intractability. In *57th IEEE Annual Symposium on Foundations of Computer Science (FOCS),* , pages 655–664. IEEE, 2016.

[14] J. Dongarra and F. Sullivan. Guest editors introduction: The top 10 algorithms. *Computing in Science & Engineering*, 2(1):22–23, 2000.

[15] J. Fan and I. Gijbels. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, volume 66. CRC Press, 1996.

[16] C. R. Genovese, M. Perone-Pacifico, I. Verdinelli, L. Wasserman, et al. Nonparametric ridge estimation. *The Annals of Statistics*, 42(4):1511–1545, 2014.

[17] A. Goldenshluger and O. Lepski. Bandwidth selection in kernel density estimation: oracle inequalities and adaptive minimax optimality. *The Annals of Statistics*, pages 1608–1632, 2011.

[18] A. G. Gray and A. W. Moore. Nonparametric density estimation: Toward computational tractability. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 203–211. Society for Industrial and Applied Mathematics, 2003.

[19] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.

[20] L. Greengard and J. Strain. The fast gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991.

[21] K. Jisu, Y.-C. Chen, S. Balakrishnan, A. Rinaldo, and L. Wasserman. Statistical inference for cluster trees. In *Advances in Neural Information Processing Systems*, pages 1831–1839, 2016.

[22] S. Joshi, R. V. Kommaraji, J. M. Phillips, and S. Venkatasubramanian. Comparing distributions and shapes using the kernel distance. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 47–56. ACM, 2011.

[23] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 273–282. ACM, 1994.

[24] K. A. Lai, A. B. Rao, and S. Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 665–674. IEEE, 2016.

[25] D. Lee, A. Gray, and A. Moore. Dual-tree fast gauss transforms. *Advances in Neural Information Processing Systems*, 18:747, 2006.

[26] W. B. March, B. Xiao, and G. Biros. Askit: Approximate skeletonization kernel-independent treecode in high dimensions. *SIAM Journal on Scientific Computing*, 37.

[27] C. Musco and C. Musco. Provably useful kernel matrix approximation in linear time. *arXiv preprint arXiv:1605.07583*, 2016.

[28] R. Panigrahy, K. Talwar, and U. Wieder. Lower bounds on near neighbor search via metric expansion. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 805–814. IEEE, 2010.

[29] J. M. Phillips. ε-samples for kernels. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1622–1632. Society for Industrial and Applied Mathematics, 2013.

[30] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

[31] A. Rinaldo and L. Wasserman. Generalized density clustering. *The Annals of Statistics*, pages 2678–2722, 2010.

[32] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[33] E. Schubert, A. Zimek, and H.-P. Kriegel. Generalized outlier detection with flexible kernel density estimates. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 542–550. SIAM, 2014.

[34] R. Spring and A. Shrivastava. A new unbiased and efficient class of lsh-based samplers and estimators for partition function computation in log-linear models. *arXiv preprint arXiv:1703.05160*, 2017.

[35] G. Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 11–20. IEEE, 2012.

[36] S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

[37] Y. Zheng, and J.M. Phillips. Coresets for Kernel Regression. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 645–654. ACM, 2017.