Proceedings of Machine Learning Research vol 99:1–19, 2019 32nd Annual Conference on Learning Theory

Discrepancy, Coresets, and Sketches in Machine Learning

Zohar Karnin ZKARNIN@AMAZON.COM

Edo Liberty LIBERTYE@AMAZON.COM

Editors: Alina Beygelzimer and Daniel Hsu

Abstract

This paper defines the notion of class discrepancy for families of functions. It shows that low discrepancy classes admit small offline and streaming coresets. We provide general techniques for bounding the class discrepancy of machine learning problems. As corollaries of the general technique we bound the discrepancy (and therefore coreset complexity) of logistic regression, sigmoid activation loss, matrix covariance, kernel density and any analytic function of the dot product or the squared distance. Our results prove the existence of ϵ-approximation $O(\sqrt{d}/\epsilon)$ sized coresets for the above problems. This resolves the long-standing open problem regarding the coreset complexity of Gaussian kernel density estimation. We provide two more related but independent results. First, an exponential improvement of the widely used merge-and-reduce trick which gives improved streaming sketches for any low discrepancy problem. Second, an extremely simple deterministic algorithm for finding low discrepancy sequences (and therefore coresets) for any positive semi-definite kernel. This paper establishes some explicit connections between class discrepancy, coreset complexity, learnability, and streaming algorithms.

Acknowledgments: The authors sincerely thank Nikhil Bansal, Nikhil Srivastava, Jeff Phillips, Wai Ming Tai, and Camron Musco for generously sharing their time and ideas. They helped us uncover the usefulness of Banaszczyk's theorem for proving Lemma 16, compare to other results on coresets and discrepancy (specifically on Gaussian Kernel Density estimation), and understand the connection to graph sparsification and matrix column subset selection results.

1. Introduction

The study of coresets in optimization as a whole and in machine learning specifically has a long history. The basic setup is as follows. Suppose you are trying to optimize an expression over a set of items, data points, or examples. The optimization problem is difficult. Its running time dependence on the input set size is square, cubic, or even exponential. As a result, there is a strong incentive to reduce the cardinality of that set. The goal is, therefore, to pinpoint a small subset of data items which approximates the entire input set with respect to the optimization at hand. Such small sets are called coresets. This idea is very general and applies to geometric properties of the data Agarwal et al. (2005), clustering Har-Peled and Kushal (2005) Feldman and Langberg (2011), classification Har-Peled et al. (2007), regression Munteanu et al. (2018a) machine learning Bachem et al. (2017), density estimation Phillips and Tai (2018b), and many other problems.

Obtaining small coresets and understanding the coreset complexity (the size of the minimal coreset) of different problems is of significant theoretical and practical importance. While some problems obviously do not admit small coresets, others do. There are several results that connect the simplicity of the measure and its coreset complexity. In this manuscript, we focus solely on

sums of functions applied to the input items. That is, for $\{x_i, \ldots, x_n\} \subset \mathcal{X}$ we measure $F(q) = \sum_i f(x_i, q)$ for $q \in \mathcal{Q}$ which is either some model parameters or a query. For example, one could consider the sum of sigmoid activation losses $F(q) = \sum_{i=1}^{n} 1/(1 + \exp(\langle x_i, q \rangle))$ and $x, q \in \mathbb{R}^d$. Using Chernoff's inequality and a union bound already shows that sampling $\tilde{O}(d/\epsilon^2)$ items gives an ϵn approximation to this sum.[1] In general, for families of VC dimension v, a sample of $(v + \log(1/\delta))/\epsilon^2$ suffices Talagrand (1994). For logistic regression and many other problems $O(d/\epsilon)$ samples are enough due to fast rate generalization results Van Erven et al. (2015). For Gaussian kernel density, it is known that a sample size of $O(1/\epsilon^2)$ suffices independently of d Lopez-Paz et al. (2015). These results require different analyses and seem to stem from different mathematical underpinnings. This paper provides a general framework for obtaining and improving on these results.

Rademacher complexity (see for example Bartlett and Mendelson (2003)) is a standard measure of generalization. In other words, bounding the Rademacher complexity is a good way to upper bound the sample complexity. A sample is an instance of a coreset which is chosen i.i.d. from the data (or the underlying distribution). A carefully selected coreset can, at least potentially, be better than a uniformly sampled one. It can be smaller and still give the same generalization power or give better generalization with the same number of data points. There are papers such as Langberg and Schulman (2010); Tolochinsky and Feldman (2018) and references therein that tie the coreset size to the VC dimension of the function family and the average sensitivity of the dataset. These relationships come up as tools for constructing coresets rather than complexity measures aimed to characterize generalization ability. This paper defines the analog to Rademacher complexity that aims to characterize the coreset complexity, i.e., the generalization ability of the best possible coreset of a fixed size.

We show that our result applies to any analytic function of the dot product. These include Logistic Regression $F(q) = \sum_i \log(1 + \exp(\langle y_i x_i, q \rangle))$, Covariance or matrix approximation $F(q) = \sum_i \langle x_i, q \rangle^2$, sigmoid activation loss $F(q) = \sum_i 1/(1 + \exp(\langle y_i x_i, q \rangle))$, linear regression $F(q) = \sum_i (\langle y_i x_i, q \rangle - y_i)^2$ and many others. For all the aforementioned $x, q \in R^d$ and $y_i \in \{-1, 1\}$. By bounding the class discrepancy of all such functions we prove the existence of coresets of size $O(\sqrt{d}/\epsilon)$ for all of them.

We note that while we obtain a universal *additive guarantee* it is often much harder to get a *multiplicative guarantee*. For Logistic regression, for example, a recent paper Munteanu et al. (2018b) provides a coreset with a *multiplicative guarantee* that is based on an average sensitivity property of the dataset. They provide a lower bound for the size of a multiplicative error coreset proving in particular that in general, it is not possible to achieve $m \ll n$. The coreset they build is of cardinality $m \approx \mu\sqrt{nd^3}/\epsilon^2$ where $\mu \geq 1$ is a complexity measure of the dataset. Tolochinsky and Feldman (2018) give a generic multiplicative coreset construction for any monotonic function with ℓ_2^2 regularization. The dependence they get is $\tilde{O}(d/\epsilon^2)$ ignoring logarithmic factors. Additive approximation coresets are also studies in the ϵ-approximation literature which is also related to the discrepancy of the problem Mustafa and Varadarajan (2017). In Braverman et al. (2016), some connections are drawn between additive and multiplicative guarantees by providing a method to use an additive guarantee along with sensitivity scores in order to provide a multiplicative guarantee.[2]

1. Using $\tilde{O}(\cdot)$ to suppress poly-logarithmic terms.
2. These methods might be combined with our results to obtain improved multiplicative guarantees, but this would not be a trivial result and we defer it to future research.

We show that our result also applies to any analytic function of the squared distance. A prime example of that is Gaussian kernel density estimation. Kernel density estimation is a popular tool in data analysis aimed to estimate a continuous distribution with a finite set of points. Among other applications, this tool is used for outlier detection Schubert et al. (2014), regression Fan (2018), and clustering Rinaldo et al. (2010). A thorough survey could be found in Silverman (2018). Given a set of n data points $\{x_1, \ldots, x_n\}$ and a query q, the Gaussian density estimate at point q is given by $\sum_i K(x_i, q) = \sum_i e^{-\|x-q\|^2}$. Obtaining the smallest possible coreset for this problem has been open for several years. The state-of-the-art is given by Phillips and Tai (2018b) (see references within). They achieve coresets of size $O(\sqrt{d\log(1/\epsilon)}/\epsilon)$ where d is the dimension of the original data points. Their result holds for any Lipchitz bounded positive semi-definite kernels. The result is constructive though it is polynomial rather than (quasi-)linear in the data size. The authors give an almost matching lower bound of $\sqrt{d}/\epsilon$ and pose an open question for closing the gap between the bounds. In this paper we resolve the open question by Phillips and Tai (2018b) and prove that the coreset complexity of Gaussian kernel density is indeed $O(\sqrt{d}/\epsilon)$, matching the lower bound. In fact, we show that this is the coreset complexity for any bounded analytic function of the squared distance $f(x, q) = f(\|x - q\|^2)$.

In high dimensions $\sqrt{d}/\epsilon$ could be large. It is known (see Lopez-Paz et al. (2015), Theorem 1) that a uniform random sample of $\log(1/\delta)/\epsilon^2$ points gives a coreset w.p. $1-\delta$ for some kernel types. Phillips and Tai (2018b) provide an algorithm based on the Frank-Wolf method that achieves a $1/\epsilon^2$ sized coreset. In section 3.1 we provide (as a stand-alone result) a very simple and deterministic algorithm for constructing coresets of size $1/\epsilon^2$ for any positive semi-definite kernel. The worst-case coreset lower bound is $\Omega(1/\epsilon^2)$ which matches the coreset achieved by sampling. Yet, for real data, the deterministic algorithm outperforms random sampling significantly (experiments not included in this manuscript).

2. Class Discrepancy and Coreset Complexity

In both machine learning and in streaming and sketching problems our goal is (often) to approximate sums or expectations of well-behaved functions. Specifically, we need to approximate $\mathbb{E}_x f(x)$ or $\frac{1}{n}\sum_{i=1}^n f(x_i)$ for every $f \in \mathcal{F}$ where $\mathcal{F}$ is a family of functions and $x_i \in \mathcal{X}$ are either sampled training examples or an arbitrary set of stream items. Standard generalization results show that for a large enough value of n the average approximates the mean if the complexity of $\mathcal{F}$ is bounded and the samples x_i are drawn i.i.d. from an underlying distribution. We therefore focus on approximating the average, or rather, the sum $\sum_{i=1}^n f(x_i)$. For notational convenience, we use a parameter $q \in \mathcal{Q}$ to index into $\mathcal{F}$ explicitly. In other words, there is a bijective mapping between $\mathcal{Q} \equiv \mathcal{F}$ such that $f(x) \in \mathcal{F}$ iff there exists $q \in \mathcal{Q}$ such that $f(x, q) = f(x)$. We keep using the two different functions $f : \mathcal{X} \to \mathbb{R}$ and $f : \mathcal{X}, \mathcal{Q} \to \mathbb{R}$ interchangeably. One should think about q as either the model parameters or a query for the sketch.

The goal is to produce a coreset. This is a small set $S \subset [n]$ and weights $w \in \mathbb{R}_+^n$ such that $\tilde{F}(q) = \sum_{i\in S} w_i f(x_i, q)$ approximates $F(q)$. Approximation here means that $|\tilde{F}(q) - F(q)| \leq \epsilon n$ for all $q \in \mathcal{Q}$ simultaneously. There are more complicated formulations such as weak coresets which we will not touch upon in this manuscript. Generating a concise representation $\tilde{F}$ for F allows one to optimize over $\tilde{F}$ instead of F which is more efficient. Moreover, if the resulting coresets are mergeable, this could be done on separate streams without the need for communication or assuming randomness in the partitioning.

For bounded functions f, uniform sampling of $O(\log(1/\delta)/\epsilon^2)$ combined with a union bound over $|\mathcal{Q}|$ always provides a valid solution using $O(\log(|\mathcal{Q}|)/\epsilon^2)$ items. While $|\mathcal{Q}|$ is often infinite it can be replaced by a finite (albeit usually exponentially large) epsilon net Q_ϵ. We present a mechanism for producing coresets which are much smaller than those achieved by sampling for a large class of problems in a unified manner. Moreover, our solutions create streaming algorithms with fully mergeable sketches. The size of the optimal coreset appears to be intimately tied to the class discrepancy properties of the associated functions.

2.1. Class Discrepancy

We begin by giving three equivalent definitions of complexity based on discrepancy for sets, functions, and function families. We will use all three interchangeably throughout the manuscript. Our notation is intentionally similar to the definition of the Rademacher complexity for reasons that will become clear later.

Definition 1 (Class Discrepancy) *Let $A \subset \mathbb{R}^m$ and $\sigma \in \{-1,1\}^m$ the class discrepancy of A is $D_m(A) = \min_\sigma \max_{a\in A} \left|\frac{1}{m}\sum_{i=1}^m \sigma_i a_i\right|$.*

Definition 2 (Class Discrepancy) *Let $f : \mathcal{X}, \mathcal{Q} \to \mathbb{R}$ and $\sigma \in \{-1,1\}^m$. The class discrepancy of f w.r.t. $\{x_1,\ldots,x_m\} \subset \mathcal{X}$ is $D_m(f) = \min_\sigma \max_{q\in\mathcal{Q}} \left|\frac{1}{m}\sum_{i=1}^m \sigma_i f(x_i,q)\right|$.*

Definition 3 (Class Discrepancy) *Let $\mathcal{F}$ be a family of functions $f : \mathcal{X} \to \mathbb{R}$ and $\sigma \in \{-1,1\}^m$. The class discrepancy of $\mathcal{F}$ w.r.t. $\{x_1,\ldots,x_m\} \subset \mathcal{X}$ is $D_m(\mathcal{F}) = \min_\sigma \max_{f\in\mathcal{F}} \left|\frac{1}{m}\sum_{i=1}^m \sigma_i f(x_i)\right|$.*

The class discrepancy of f or $\mathcal{F}$ without a reference a set $\{x_1,\ldots,x_m\}$ is the upper bound on any subset of $\mathcal{X}$ of size m. Throughout the manuscript, we assume a bijective mapping between $\mathcal{F}$ and $\mathcal{Q}$. Specifically, any function in $\mathcal{F}$ can be written as f_q and has a unique $q \in \mathcal{Q}$ such that $f_q(x) = f(x,q)$. In the context of machine learning, one should think about $f(x,q)$ as the loss associated with example x and model parameters q. The set A should be thought of as the set of all possible induced loss vectors. Namely, $a \in A$ if there is a model q such $a_i = f(x_i,q)$.

To understand our motivation, consider the following informal explanation of the Rademacher Complexity applied to ML problems. In PAC learning there exists a set of examples (often with labels). We aim to find a regressor/classifier from a given family that suffers the least loss on the set. Having a low Rademacher complexity means that we can optimize over a sample of roughly half the examples at random (each w.p. $1/2$). Low Rademacher complexity guarantees that, in expectation, twice the loss on the sample is roughly the same as the loss on the entire set. This translates to a generalization bound. In other words, the Rademacher complexity gives a guarantee for the loss of coresets chosen uniformly at random.

Coming back to discrepancy. Having the ability to choose the signs arbitrarily lets us choose an advantageous subset of examples. We can algorithmically choose to minimize the induced error and guarantee to have (roughly) the same performance on the entire set. This set is, in fact, a coreset. The class discrepancy of a problem helps us determine the obtainable coreset size. We will show in the following sections several examples for which a coreset can be significantly smaller than the random sample while maintaining the same guarantees. This will be done by showing that for a wide range of interesting problems in machine learning $D_m(\mathcal{F}) = o(R_m(\mathcal{F}))$. This intuition is restated more explicitly in the next section.

2.2. Coreset Complexity

In this section, we point out a direct connection between coreset complexity and class discrepancy. The connection is a simple application of the folklore argument know as the "the halving trick". For simplicity, in what follows we focus on functions f whose range is $[0, 1]$.

Definition 4 (Coreset Complexity) *For a function $f : \mathcal{X}, \mathcal{Q} \to \mathbb{R}$ let $F(q) = \sum_{i=1}^{m} f(x_i, q)$ for any set $\{x_1, \ldots, x_m\} \subset \mathcal{X}$. For a set $S \subset [m]$ let $\tilde{F}(q) = \sum_{i \in S} w_i f(x_i, q)$ for some $w \in \mathbb{R}_+^m$ which is independent of q. The coreset complexity of f is the size of the smallest set S such that $\forall q \in \mathcal{Q}\ |F(q) - \tilde{F}(q)| \leq \epsilon m$.*

The following facts are true for the common cases where $D_m = O(c/m)$ or $D_m = O(c/\sqrt{m})$. Although they were previously known (see e.g., Phillips (2009), Theorem 1.1) we give their proof here for completeness.

Fact 5 *Any function f with class discrepancy $D_m = O(c/m)$ has coreset complexity of $O(c/\epsilon)$.*

Fact 6 *Any function f with class discrepancy $D_m = O(c/\sqrt{m})$ has coreset complexity $O(c^2/\epsilon^2)$.*

Proof For a set of n points $x_1, \ldots, x_n$ and arbitrary query q, consider the signed-sum error function $E(q) = \sum_{i=1}^{n} \sigma_i f(x_i, q)$ where $\sigma_i \in \{-1, 1\}$. Recalling $F(q) = \sum f(x_i, q)$, we consider $\tilde{F}_+(q) = F(q) + E(q) = \sum_{i|\sigma_i=1} 2f(x_i, q)$ and similarly $\tilde{F}_-(q) = F(q) - E(q) = \sum_{i|\sigma_i=-1} 2f(x_i, q)$. We have that both $\tilde{F}_+(q)$ and $\tilde{F}_-(q)$ are approximations for $F(q)$ obtained by coresets of item-weight 2. The error is at most $|\tilde{F}_\pm(q) - F(q)| = |E(q)|$, and one of the coresets are of cardinality of at most $n/2$. The above is true for any choice of signs σ, specifically, for those minimizing $\max_q |E(q)|$. By definition we can select signs such that $|E(q)| \leq mD_n$.

Naturally, one could iterate this process. Starting with n items and ending with m. Let F_t denote the (unweighted) sum of functions f after t iterations and n_t denote the cardinality of the coreset.[3]

$$F = F_0 = 2F_1 \pm n_0 D_{n_0} = 4F_2 \pm 2n_1 D_{n_1} \pm n_0 D_{n_0} = \ldots = 2^T F_T \pm \sum_{t=0}^{T-1} 2^t n_t D_{n_t}$$

Here T stand for the total number of iterations. Let us analyze the error term. Given $n_t \leq n/2^t \approx m$ and the polynomial dependence of D_m on m we have

$$\sum_{t=0}^{T-1} 2^t n_t D_{n_t} \leq n \sum_{t=0}^{T-1} D_{n/2^t} = n \cdot O(D_m).$$

Setting m for which $D_m = \epsilon$ gets coresets with appropriate cardinalities and completes the proof. ■

Fact 7 *Class discrepancy bounds are tight asymptoticly for unweighted coreset complexity.*

3. The expression $a = b \pm c$ means $|a - b| \leq c$

Proof Taking for example the bound of Fact 5, if we can guarantee the existence of an unweighted coreset of size c/ϵ, then for m items a coreset of size $m/2$ provides a sign assignment with an error of $\epsilon = 2c/m$, leading to an upper bound of $O(c/m)$ on the class discrepancy. ■

The following it a straight forward fact which is provided mainly for convenience. It loosely says that optimizing models on coresets generalizes. In other words, ERM works as expected.

Fact 8 *Let $f(x, q)$ be the loss suffered by model q on example x. Let $R(q) = \frac{1}{n}F(q) = \frac{1}{n}\sum_{i=1}^{n} f(x_i, q)$ be the empirical risk associated with it. Let q^* denote the best empirical risk minimizer on the data ($q^* = \arg\min_q F(q)$). Let $\tilde{q}$ be the minimizer of q over an optimal weighted coreset of size m ($\tilde{q} = \arg\min_q \tilde{F}(q)$). Then $R(\tilde{q}) \leq R(q^*) + O(D_m)$.*

Proof This fact follows from the standard argument about empirical risk minimization.

$$R(\tilde{q}) = \frac{1}{n}F(\tilde{q}) \leq \frac{1}{n}\tilde{F}(\tilde{q}) + O(D_m) \leq \frac{1}{n}\tilde{F}(q^*) + O(D_m) \leq \frac{1}{n}F(q^*) + O(D_m) = R(q^*) + O(D_m)$$

The first and last transitions are by definition. The second and fourth are by the approximation bounds above. The third transition is due to the optimality of $\tilde{q}$ for $\tilde{F}$ ■

2.3. Streaming Coreset Complexity

We claim that low class discrepancy implies concise streaming mergeable coresets as well.

Definition 9 (Streaming Coreset Algorithm) *A streaming coreset algorithm for $f : \mathcal{X}, \mathcal{Q} \to \mathbb{R}$ receives and arbitrary set $\{x_1, \ldots, x_m\} \subset \mathcal{X}$ one item after the other. At time $t \leq m$, the algorithm maintains a subset $S_t \subset \{x_1, \ldots, x_t\}$ and uses at most $O(|S_t|)$ auxiliary memory. At the end of the stream, the algorithm must output S and w such that $\forall q \in \mathcal{Q}$ $|F(q) - \tilde{F}(q)| \leq \epsilon m$ where $F(q) = \sum_{i=1}^{m} f(x_i, q)$ and $\tilde{F}(q) = \sum_{i \in S} w_i f(x_i, q)$. The size of the streaming coreset is $\max_t |S_t|$.*

Definition 10 (Streaming Coreset Complexity) *The streaming coreset complexity for $f : \mathcal{X}, \mathcal{Q} \to \mathbb{R}$ is the minimal streaming coreset size among all possible streaming coreset algorithms for f.*

The following statements upper bound streaming coreset complexities for functions. We note that these bounds are only poly-logarithmically larger than their offline counterparts.

Theorem 11 *Any function f with class discrepancy $D_m(f) = O(c/m)$ has streaming coreset complexity of $O\left(c \log^2(\epsilon n/c)/\epsilon\right)$.*

Theorem 12 *Any function f with class discrepancy $D_m = O(c/\sqrt{m})$ has streaming coreset complexity of $O\left(c^2 \log^3(\epsilon^2 n/c)/\epsilon^2\right)$.*

Theorems 11 and 12 are achieved by deterministic algorithms. They could be thought of extensions of the MRL algorithm Manku et al. (1999) for streaming quantile sketching. Quantile sketching falls into this framework since it corresponds to $f(x, q) = 1$ if $x > q$ and 0 else. The techniques of the above Theorems could also be associated with Matousek (1995), providing a merge-reduce framework for additive coresets. More details and the proof of correctness are given in Appendix A.

Recently, Karnin et al. (2016) provided an improved (optimal) streaming quantile coreset algorithm by improving the merge-reduce technique in a way tailored to the quantile problem. In the typical merge-reduce framework, the algorithm is based on finding an ϵ-coreset on subsets of size dependent on ϵ rather than on n. The novelty of Karnin et al. (2016) is in suggesting a way to use different values of ϵ for these local coreset constructions. This ends up providing a randomized algorithm with no dependence on n and doubly logarithmic dependence on the failure probability. Generalizing their construction requires more work and the main ideas are as follows. We argued above that $\tilde{F}_+$ and $\tilde{F}_-$ are both good approximations for F. We can also take $\tilde{F}_\pm$ which is $\tilde{F}_+$ or $\tilde{F}_-$ with equal probability. Clearly $|\tilde{F}_\pm - F| \leq |E|$ as before. But now, $\mathbb{E}[\tilde{F}_\pm] = F$ as well. In the streaming algorithm, we apply this compaction (converting F to $\tilde{F}_\pm$) many times to small subsets of items from the stream. This allows us to use concentration results to bound the overall error. So far, analogous ideas where used in Karnin et al. (2016); the main departure is that $\tilde{F}_\pm$ has half the support of F only in expectation.

Theorem 13 *Any function f with class discrepancy $D_m(f) = O(c/m)$ has streaming coreset complexity of $O\left(c \log^2 \log(|Q_\epsilon|/\delta)/\epsilon\right)$. Q_ϵ is an epsilon net for f on $\mathcal{Q}$. The streaming coreset algorithm is randomized and fails with probability at most δ.*

Theorem 14 *Any function f with class discrepancy $D_m(f) = O(c/\sqrt{m})$ has streaming coreset complexity of $O\left(c^2 \log^3 \log(|Q_\epsilon|/\delta)/\epsilon^2\right)$. Q_ϵ is an epsilon net for f on $\mathcal{Q}$. The streaming coreset algorithm is randomized and fails with probability at most δ.*

The set Q_ϵ is an ϵ-net for $\mathcal{Q}$. It is a finite subset of $\mathcal{Q}$ such that for every $q \in \mathcal{Q}$ there exist some $\tilde{q} \in Q_\epsilon$ for which $\sup_{x \in \mathcal{X}} |f(q,x) - f(\tilde{q},x)| < \epsilon$. We note that the size $|Q_\epsilon|$ is often exponential in the problem parameters. Nevertheless, our dependence on the failure probability is *doubly* logarithmic. This means the dependence on the problem parameters is still only polylogarithmic. The above improves on the well-known merge-and-reduce tree construction by Bentley and Saxe (1980). Moreover, it is likely that a uniform ϵ-net for $\mathcal{Q}$ is not required for the sake of minimization (ERM on the final sketch). See literature on weak coresets (e.g. Feldman et al. (2007)) and concentration results based on doubling dimensions in classification/query space Bshouty et al. (2009). The refinement of the above results is left for future work.

3. Class Discrepancy of Analytic Functions of Dot Products

Now that we proved the usefulness of low class discrepancy, we move to upper bound it for common family functions. We provide a coreset suitable for analytical functions of the inner product $\langle q, x \rangle$ or squared Euclidean distance $\|q - x\|^2$. The idea is to find a set of signs that simultaneously balance $\langle q, x \rangle^k$ for all powers k and unit vectors q.[4] By controlling all powers of $\langle q, x \rangle$ we control any sum of these powers. It follows that this coreset can be used to control, for example, the logistic loss function $L(q,x) = \log(1 + \exp(\langle q, x \rangle))$, the gaussian Kernel $K(q,x) = \exp(-\lambda \|q - x\|^2)$, or the sigmoid activation loss $1/(1 + \exp(\langle q, x \rangle))$.

We start with some notation and trivial properties. For a vector $q \in \mathbb{R}^d$ let $q^{\otimes k}$ represent the k-dimensional tensor obtained from the outer product of q with itself k times. For a k dimensional tensor with d^k entries X we consider the measure $\|X\|_{T_k} = \max_{q \in \mathbb{R}^d, \|q\|=1} \left|\langle X, q^{\otimes k} \rangle\right|$.

4. We Assume that $\|x\|, \|q\| \leq 1$ for ease of presentation. As above our results extend to generic bounds on the radius of q

Fact 15 *$\|X\|_{T_k}$ is a norm*

Proof We prove the claim directly from the definition of a norm. Notice that for any $X \neq 0$, $\left\langle X, q^{\otimes k} \right\rangle$ is a non-zero polynomial in q. It follows that there must be q for which its value is non-zero, meaning that $\|X\|_{T_k} = 0$ iff $X = 0$. For a scalar a, we clearly have by definition that $\|aX\|_{T_k} = |a|\|X\|_{T_k}$. Lastly, by the max definition we have $\|X+Y\|_{T_k} = \max_q \left|\langle X+Y, q^{\otimes k}\rangle\right| \leq \max_q \left|\langle X, q^{\otimes k}\rangle\right| + \max_q \left|\langle Y, q^{\otimes k}\rangle\right| = \|X\|_{T_k} + \|Y\|_{T_k}$ ■

We are now ready for the lemma controlling all powers of inner products simultaneously.

Lemma 16 *For any set of vectors $x_i \in \mathbb{R}^d$ with $\|x_i\| \leq 1$ there exist a set of signs σ_i such that for all k simultaneously $\left\|\sum_i \sigma_i x_i^{\otimes k}\right\|_{T_k} \leq O(\sqrt{dk\log^3 k})$ (the 3 power of the term $\log(k)$ can be reduced to any constant power larger than 2).*

Proof The proof will use Banaszczyk's theorem Banaszczyk (1998). Let $\mathcal{K}$ be a convex body in Euclidean space with Gaussian measure at least 1/2 ($\Pr[g \in \mathcal{K}] \geq 1/2$ when g is i.i.d. Gaussian). Let $x_1, \ldots, x_n$ be vectors with $\|x_i\| \leq 1$. Then, there exist signs σ such that $\sum \sigma_i x_i \in C\mathcal{K}$ for some constant C.

To use Banaszczyk's theorem we begin with defining our convex body. Define the norm $\|\psi\|_T$ of a vector ψ as follows. Look at the first d coordinates of ψ as a vector ψ_1, the next d^2 coordinates of ψ as a matrix ψ_2 the next d^3 coordinates as a three tensor ψ_3 etc. We define $\|\psi\|_T = \max_k \|\psi_k\|_{T_k}/\sqrt{\log(k)}$. Here, $\|\cdot\|_{T_k}$ is the special spectral norm defined at the beginning of the section. The maximum over norms of subvectors is clearly a norm in itself, meaning that $\|\cdot\|_T$ is indeed a norm. It follows that the set $\mathcal{K} = \{\psi \mid \|\psi\|_T \leq c\sqrt{d}\}$ is convex.

We now need to show that the Gaussian measure of $\mathcal{K}$ is at least $1/2$. That is, with probability at least $1/2$ a vector of random Gaussian entrees g belongs to $\mathcal{K}$. Consider a random i.i.d. Gaussian Tensor $g_k \in \mathbb{R}^{d^k}$.

A trivial modification of Theorem 1 from Tomioka and Suzuki (2014) shows that $\Pr[\|g_k\|_{T_k} \geq c\sqrt{d\log(k)}] \leq 1/10k^2$ for some constant c. The only change needed in the proof is the size of the epsilon net which changes from $(2\log(3/2)/k)^{kd}$ for Tomioka and Suzuki (2014) to $(2\log(3/2)/k)^d$. The reason we require a net over a smaller space is due to us bounding the inner product with a rank one tensor rather than rank k. Union bounding on all values of k we get $\sum_k 1/10k^2 \leq 1/2$ which shows $g = [g_1, \mathrm{flat}(g_2), \mathrm{flat}(g_3), \ldots]$ belongs to $\mathcal{K}$ with probability at least $1/2$, where $\mathrm{flat}(g_k)$ is the flattening of the tensor into a one dimensional vector. We now define a mapping $\psi(x)$ of $x \in \mathbb{R}^d$ to a high dimensional space.

$$\psi(x) = \left[x, \frac{\mathrm{flat}(x^{\otimes 2})}{\sqrt{2\log^2(2)}}, \frac{\mathrm{flat}(x^{\otimes 3})}{\sqrt{3\log^2(3)}}, \ldots, \frac{\mathrm{flat}(x^{\otimes k})}{\sqrt{k\log^2(k)}}, \ldots\right]$$

Note that for $\|x\| \leq 1$ we have $\|\psi(x)\|_2 = (\sum_k 1/k\log^2(k))^{1/2} = O(1)$.

We are now ready to apply Banaszczyk's theorem. There exist signs σ_i such that $\psi = \sum_i \sigma_i \psi(x_i) \in C\mathcal{K}$, meaning $\|\psi\|_T \leq C$. Since $\psi_k = \sum_i \sigma_i x_i^{\otimes k}/\sqrt{k\log^2 k}$ we get that

$$\max_k \frac{\|\sum_i \sigma_i x_i^{\otimes k}\|_{T_k}}{\sqrt{k\log^3(k)}} \leq O\left(\sqrt{d}\right)$$

This concludes the proof of the statement. ■

Lemma 17 *Let f be a function of the inner product $f(x,q) = f(\langle x,q\rangle)$ and let $f = \sum_k \alpha_k \langle x,q\rangle^k$ be its Taylor expansion. The class discrepancy of f indexed by $\|q\| \leq 1$ is bounded by*

$$D_m = \min_\sigma \sum_i \sigma_i f(x_i,q) = O\left(\sqrt{d}\sum_k |\alpha_k|\sqrt{k\log^3(k)}\right)$$

For general $\|q\| \leq R$ we get

$$D_m = \min_\sigma \sum_i \sigma_i f(x_i,q) = O\left(\sqrt{d}\sum_k |\alpha_k| R^k\sqrt{k\log^3(k)}\right)$$

Proof The proof follows from combining the above.

$$\sum_i \sigma_i f(x_i,q) = \sum_k \alpha_k \sum_i \sigma_i \langle x_i,q\rangle^k = \sum_k \alpha_k \left\langle \sum_i \sigma_i x_i^{\otimes k}, q^{\otimes k}\right\rangle \leq$$

$$\sum_k |\alpha_k| \cdot \left\|\sum_i \sigma_i x_i^{\otimes k}\right\|_{T_k} \cdot \|q\|^k$$

By Lemma 16 we can find signs σ such that $\left\|\sum_i \sigma_i x_i^{\otimes k}\right\|_{T_k} \leq c\sqrt{dk\log^3(k)}$. Substituting into the above, the lemma follows. ■

Theorem 18 *Let $f : \mathbb{R} \to \mathbb{R}$ be analytic. There exist a radius R such that functions $f = f(\langle q,x\rangle)$, indexed by $\|q\| \leq R$, have class discrepancy $O(\sqrt{d}/m)$.*

Proof Recall that for analytic functions f we have $\left|\frac{d^k f}{dz^k}(z)\right| \leq C^{k+1}k!$ for some constant C. Considering the taylor expansion of f near zero, for $R < 1/C$ the sum $\sum_k |\alpha_k| R^k\sqrt{k\log^3(k)} \leq C\sum_k (CR)^k\sqrt{k\log^3(k)}$ corresponding to Lemma 17 converges to a constant. The result follows. ■

The following two corollaries apply to the Logistic function and sigmoid activation loss function. They are easy to obtain by noticing the coefficients of the functions' Taylor expansion.

Corollary 19 *The class discrepancy of the Logistic function $f(\langle q,x\rangle) = \log(1+\exp(\langle q,x\rangle))$ in dimension d, for $\|q\| \leq 1$ is $O(\sqrt{d}/m)$.*

Corollary 20 *The class discrepancy of the sigmoid activation loss function $f(\langle q,x\rangle) = 1/(1+\exp(\langle q,x\rangle))$ in dimension d, for $\|q\| \leq 1$ is $O(\sqrt{d}/m)$.*

Corollary 21 *The class discrepancy of the covariance function $f(\langle q,x\rangle) = \langle q,x\rangle^2$ in dimension d, for $\|q\| \leq 1$ is $O(\sqrt{d}/m)$. This gives coresets for matrix column subset selection such that $\|XX^T - \tilde{X}\tilde{X}^T\| \leq \epsilon n$ where $\tilde{X}$ contains only $O(\sqrt{d}/\epsilon)$ rescaled columns of the matrix X.*

Theorem 22 *Let $f : \mathbb{R} \to \mathbb{R}$ be analytic. There exist a radius R such that the function $f(\|x-q\|^2)$, indexed by $\|q\| \leq R$, has class discrepancy $O(\sqrt{d}/m)$.*

Proof By transforming x to $\tilde{x} = (1, \sqrt{2}x, \|x\|^2)$ and q to $\tilde{q} = (\|q\|^2, -\sqrt{2}q, 1)$ we get $\langle \tilde{x}, \tilde{q} \rangle = \|q - x\|^2$. Moreover, $\|q\| \leq R$ gives $\|\tilde{q}\| \leq R^2 + 1$. The result follows from applying Theorem 18 to $f(\langle \tilde{q}, \tilde{x} \rangle) = f(\|q - x\|^2)$. ■

Corollary 23 *For cases where $\|q - x\| \leq 1$ for all q, x, the class discrepancy of the Gaussian kernel $K(q, x) = \exp(-\gamma\|x - q\|^2)$ in dimension d is $O(\gamma \exp(\gamma)\sqrt{d}/m)$.*

This improves upon the recent result of Phillips and Tai (2018b) by proving the existence of ϵ approximation corsets of size $\sqrt{d}/\epsilon$ for Gaussian kernel density, in the case where γ is constant. This also resolves the open problem raised by Phillips and Tai (2018b) and matches their lower bound. For non-constant γ assume w.l.o.g. $\|q - x\| \leq 1$. The Taylor series of the Gaussian kernel K exhibits $|\alpha_k| \leq \gamma^k/k!$. Plugging into the equation in the proof of Theorem 22 we get that the sum determining the constant is upper bounded by

$$\sum_{k=1}^{\infty} \gamma^k (k \log^3(k))^{1/2}/k! = O\left(\sum_{k=1}^{\infty} \gamma^k/(k-1)!\right) = O\left(\gamma \exp(\gamma)\right)$$

3.1. Towards an Efficient Algorithm

From section 3 we know that the class discrepancy of the Gaussian kernel is $D_m = O(\sqrt{d}/m)$. Here, we provide a computationally efficient bound that can be achieved with a straightforward algorithm of complexity $O(m^2)$. Together with the results of Section 2.3 this provides an efficient sketching algorithm for Kernel Density Estimation. In fact, we show that for any positive kernel $D_m = O(1/\sqrt{m})$. This bound is superior to that of the previous section for high dimensions $d > m$. More importantly, there is a very simple, intuitive, and deterministic algorithm for computing the signs σ. Given a collection of data points $X = \{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$ the density function $f : \mathbb{R}^d \to \mathbb{R}$ of a point q is defined as $F(q) = \sum_{i=1}^{n} K(x_i, q)$. Here, K is any *positive semi-definite kernel* function. The most frequent examples include

$$K(x, q) = \exp(-\|x-q\|_2^2/\lambda^2), \quad K(x, q) = \exp(-\|x-q\|/\lambda), \quad K(x, q) = (1+\|x-q\|/_2^2/\lambda^2)^{-1}$$

where λ is a scaling parameter. For simplicity, we assume that $K(x, x) \leq 1$ for all data points. Notice that for any kernel based on the distance we have $K(x, x) = 1$ exactly for all $x \in \mathbb{R}^d$.

Algorithm 1 Low discrepancy algorithm for positive semi-definite kernels

input: Kernel function $K : (\mathbb{R}^d, \mathbb{R}^d) \to [0, 1]$, points $\{x_1, \ldots, x_m\}$
output: $\sigma \in \{-1, 1\}^m$ such that $\max_q |\sum_i \sigma_i K(x_i, q)| \leq \sqrt{m}$
$\sigma_1 = 1$
for $i = 2, \ldots, m$ **do**
  $\sigma_i = -\operatorname{sign}(\sum_{j=1}^{i-1} \sigma_j K(x_j, x_i))$

Theorem 24 *Algorithm 1 achieves $\max_q |\sum_i \sigma_i K(x_i, q)| \leq \sqrt{m}$.*

Proof For any positive semi-definite kernel K there exist a mapping $\phi : \mathbb{R}^d \to \mathcal{V}$ to an inner product space $\mathcal{V}$ such that $K(x,q) = \langle \phi(x), \phi(q) \rangle$. Using this function ϕ our objective function becomes

$$|\sum_{i=1}^{m} \sigma_i K(x_i, q)| = |\sum_{i=1}^{m} \sigma_i \langle \phi(x_i), \phi(q) \rangle| = \left| \left\langle \sum_{i=1}^{m} \sigma_i \phi(x_i), \phi(q) \right\rangle \right| \leq \|\phi(q)\| \cdot \left\| \sum_{i=1}^{m} \sigma_i \phi(x_i) \right\|$$

Since $\|\phi(q)\| \leq 1$ we reduced the problem to bounding the norm of $\sum_{i=1}^{m} \sigma_i \phi(x_i)$. We show by induction on i that $\left\| \sum_{j=1}^{i} \sigma_j \phi(x_j) \right\|^2 \leq \sum_{j=1}^{i} \|\phi(x_j)\|^2 \leq i$. This is trivially true for $i = 1$ since $\|\phi(x)\| \leq 1$. Using our induction assumption we get

$$\begin{aligned}
\left\| \sum_{j=1}^{i} \sigma_j \phi(x_j) \right\|^2 &= \left\| \sum_{j=1}^{i-1} \sigma_j \phi(x_j) \right\|^2 + \|\phi(x_i)\|^2 + 2 \left\langle \sum_{j=1}^{i-1} \sigma_j \phi(x_j), \sigma_i \phi(x_i) \right\rangle \\
&\leq \sum_{j=1}^{i-1} \|\phi(x_j)\|^2 + \|\phi(x_i)\|^2 + 2\sigma_i \sum_{j=1}^{i-1} \sigma_j K(x_j, x_i) \\
&= \sum_{j=1}^{i} \|\phi(x_j)\|^2 - 2 \left| \sum_{j=1}^{i-1} \sigma_j K(x_j, x_i) \right| \leq \sum_{j=1}^{i} \|\phi(x_j)\|^2
\end{aligned}$$

The first equality simply unpacks the squared vector norm, the second transition is due to the induction assumption and the last substitutes our choice of σ (and $\text{sign}(z) \cdot z = |z|$). This completes the proof that $|\sum_{i=1}^{m} \sigma_i K(x_i, q)| \leq \sqrt{m}$ for all q. ■

Using the framework above provides a deterministic coreset construction for kernel density estimation of size $O(1/\epsilon^2)$ such that $\forall\ q\ \ |\tilde{F}(q) - F(q)| \leq \epsilon n$. This matches and simplifies the results achieved by Phillips and Tai (2018a) and Phillips and Tai (2018b). Theorem 12 leads to a deterministic streaming algorithm with a memory complexity of $O(\log^3(\epsilon^2 n)/\epsilon^2)$. For L-Lipchitz kernels, meaning K such that $|K(x, q+h) - K(x,q)|/\|h\| \leq L$ for all $h \neq 0$, Theorem 14 leads to a randomized streaming algorithm with a memory complexity of $O\left(\log^3\left(d \log\left(RLn/\delta\epsilon\right)\right)/\epsilon^2\right)$ that succeeds in finding a coreset with probability $1 - \delta$. The parameter R is the maximum norm of a query. The argument goes through a union bound over an ϵ/L-net over vectors of norm at most R, the size of which is $(RL/\epsilon)^{O(d)}$.

Note Theorem 24 provides an upper bound of $\sqrt{m}$ for the sign discrepancy. This upper bound is tight since there exist sets of vectors in high dimensions that require it. For data that lends itself to density estimation, however, one should expect input vectors to be clustered together. In such cases, the algorithm above performs much better than the worst-case bound predicts. We leave it to future work to define properties of the data that ensure better guarantees for Algorithm 1.

References

Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*, 2017.

Wojciech Banaszczyk. Balancing vectors and gaussian measures of n-dimensional convex bodies. *Random Struct. Algorithms*, 12(4):351–360, July 1998. ISSN 1042-9832. doi: 10.1002/(SICI)1098-2418(199807)12:4⟨351::AID-RSA3⟩3.0.CO;2-S. URL http://dx.doi.org/10.1002/(SICI)1098-2418(199807)12:4<351::AID-RSA3>3.0.CO;2-S.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944944.

Jon Louis Bentley and James B Saxe. Decomposable searching problems i. static-to-dynamic transformation. *Journal of Algorithms*, 1(4):301 – 358, 1980. ISSN 0196-6774. doi: https://doi.org/10.1016/0196-6774(80)90015-2. URL http://www.sciencedirect.com/science/article/pii/0196677480900152.

Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *arXiv preprint arXiv:1612.00889*, 2016.

Nader H. Bshouty, Yi Li, and Philip M. Long. Using the doubling dimension to analyze the generalization of learning algorithms. *Journal of Computer and System Sciences*, 75(6):323 – 335, 2009. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2009.01.003. URL http://www.sciencedirect.com/science/article/pii/S0022000009000130.

Jianqing Fan. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*. Routledge, 2018.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 569–578. ACM, 2011. ISBN 978-1-4503-0691-1. doi: 10.1145/1993636.1993712. URL https://doi.org/10.1145/1993636.1993712.

Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A ptas for k-means clustering based on weak coresets. In *Proceedings of the Twenty-third Annual Symposium on Computational Geometry*, SCG '07, pages 11–18, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-705-6. doi: 10.1145/1247069.1247072. URL http://doi.acm.org/10.1145/1247069.1247072.

Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. In Joseph S. B. Mitchell and Günter Rote, editors, *Proceedings of the 21st ACM Symposium on Computational Geometry, Pisa, Italy, June 6-8, 2005*, pages 126–134. ACM, 2005. ISBN 1-58113-991-8. doi: 10.1145/1064092.1064114. URL https://doi.org/10.1145/1064092.1064114.

Sariel Har-Peled, Dan Roth, and Dav Zimak. Maximum margin coresets for active and noise tolerant learning. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint*

Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, pages 836–841, 2007. URL http://ijcai.org/Proceedings/07/Papers/134.pdf.

Zohar S. Karnin, Kevin J. Lang, and Edo Liberty. Optimal quantile approximation in streams. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 71–78. IEEE Computer Society, 2016. ISBN 978-1-5090-3933-3. doi: 10.1109/FOCS.2016.17. URL https://doi.org/10.1109/FOCS.2016.17.

Michael Langberg and Leonard J Schulman. Universal ε-approximators for integrals. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 598–607. SIAM, 2010.

David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Iliya Tolstikhin. Towards a learning theory of cause-effect inference. In *International Conference on Machine Learning*, pages 1452–1461, 2015.

Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, pages 251–262, New York, NY, USA, 1999. ACM. ISBN 1-58113-084-8. doi: 10.1145/304182.304204. URL http://doi.acm.org/10.1145/304182.304204.

Jiri Matousek. Approximations and optimal geometric divide-and-conquer. *Journal of Computer and System Sciences*, 50(2):203–208, 1995.

Alexander Munteanu, Chris Schwiegelshohn, Christian Sohler, and David P. Woodruff. On coresets for logistic regression. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 6562–6571, 2018a. URL http://papers.nips.cc/paper/7891-on-coresets-for-logistic-regression.

Alexander Munteanu, Chris Schwiegelshohn, Christian Sohler, and David P. Woodruff. On coresets for logistic regression. *CoRR*, abs/1805.08571, 2018b. URL http://arxiv.org/abs/1805.08571.

Nabil H. Mustafa and Kasturi R. Varadarajan. Epsilon-approximations and epsilon-nets. *CoRR*, abs/1702.03676, 2017. URL http://arxiv.org/abs/1702.03676.

Jeff M Phillips. *Small and stable descriptors of distributions for geometric statistical problems*. PhD thesis, 2009.

Jeff M. Phillips and Wai Ming Tai. Improved coresets for kernel density estimates. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2718–2727. SIAM, 2018a. ISBN 978-1-61197-503-1. doi: 10.1137/1.9781611975031.173. URL https://doi.org/10.1137/1.9781611975031.173.

Jeff M. Phillips and Wai Ming Tai. Near-optimal coresets of kernel density estimates. *CoRR*, abs/1802.01751, 2018b. URL http://arxiv.org/abs/1802.01751.

Alessandro Rinaldo, Larry Wasserman, et al. Generalized density clustering. *The Annals of Statistics*, 38(5):2678–2722, 2010.

Erich Schubert, Arthur Zimek, and Hans-Peter 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.

Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

Michel Talagrand. Sharper bounds for gaussian and empirical processes. *The Annals of Probability*, pages 28–76, 1994.

Elad Tolochinsky and Dan Feldman. Coresets for monotonic functions with applications to deep learning. *arXiv preprint arXiv:1802.07382*, 2018.

Ryota Tomioka and Taiji Suzuki. Spectral norm of random tensors. *arXiv preprint arXiv:1407.1870*, 2014.

Tim Van Erven, Peter D Grünwald, Nishant A Mehta, Mark D Reid, and Robert C Williamson. Fast rates in statistical and online learning. *Journal of Machine Learning Research*, 16:1793–1861, 2015.

Appendix A. Proofs for Section 2.3, Sketching Coresets

The proofs of Theorems 11, 13, 12, and 14 all use the basic concept of a compactor. A compactor consumes a stream of items and outputs another stream. The output stream contains at most half the items from the input stream with double the weight. It does so by keeping a buffer of a certain capacity m. When a new item is inserted into the compactor it is added to its buffer. If the buffer is full, a compaction operation takes place. The compaction takes the elements in the buffer $x_1, \ldots, x_m$ and finds a low discrepancy assignment σ such that $\max_q |\sum_i \sigma_i f(x_i, q)| \leq mD_m$. Note that such a sequence is guaranteed to exist by the definition of the class discrepancy. For cases where an algorithm for finding this sequence σ is not known, our result applies for the guarantee of the σ sequence obtained by the algorithm. That is, if it is possible to obtain a bound of D_m yet we can only find signs obtaining a bound of $\tilde{D}_m > D_m$, our results for the obtainable signs apply for $\tilde{D}_m$. Given the sign vector σ, the compactor appends either $\{x_i | \sigma_i = 1\}$ or $\{x_i | \sigma_i = -1\}$ to the output stream.

Consider a stream of data points $x_1, \ldots, x_n$ and the output stream of a compactor $z_1, \ldots, z_{\tilde{n}}$. The error associated with the new stream w.r.t. a query q is defined as

$$\sum_{i=1}^{n} f(x_i, q) - 2 \sum_{j} f(z_j, q) \, .$$

This is the difference between the value of q on the original stream and the output stream. For a compactor we would like to bound both the length of the output stream, and the absolute value of its error.

Lemma 25 *A deterministic compactor output the smaller of the two sets* $\{x_i|\sigma_i = 1\}$ *or* $\{x_i|\sigma_i = -1\}$. *Given an input of length* n, *the output has at most* $n/2$ *items, and the error of the output stream is bounded in absolute value by* nD_m

Proof We note that the argument about the length is obvious, so we proceed to bound the error. Consider a single compaction operation done on m vectors $x_1, \ldots, x_m$. For a query q, let $F(q) = \sum_{i=1}^{m} f(x_i, q)$ be the evaluation on the items of the buffer. Let $\tilde{F}_+$ denote the function evaluated on $\{x_i|\sigma_i = 1\}_{i=1}^{m}$ (similarly $\tilde{F}_-$ defined for negative signs). Also, let $E(q) = \sum_{i=1}^{m} \sigma_i f(x_i, q)$ for the signs σ computed by the algorithm above. We have that

$$\tilde{F}_+(q) = \sum_{i,\ \sigma_i=1} 2f(x_i, q) = \sum_i f(x_i, q) + \sum_i \sigma_i f(x_i, q) = F(q) + E(q)$$

$$\tilde{F}_-(q) = \sum_{i,\ \sigma_i=-1} 2f(x_i, q) = \sum_i f(x_i, q) - \sum_i \sigma_i f(x_i, q) = F(q) - E(q)$$

meaning that the error for the items of the single compaction is bounded by

$$|\tilde{F}_\pm(q) - F(q)| = |E(q)| \leq \max_q |\sum_i \sigma_i f(x_i, q)| = mD_m$$

Summing over all n/m compactions we get that the overall error is bounded, in absolute value, by nD_m. ■

Lemma 25 alone already allow us to prove Theorems 11 and 12. The algorithms are a direct extension the well know MRL algorithm Manku et al. (1999) for quantile sketching. Note that for quantiles, $f(x, q) = 1$ if $q > x$ and 0 else. A low discrepancy sequence is achieved simply by sorting the values and assigning $\sigma_i = 1$ for all evenly positioned values in the sorted order and $\sigma_i = -1$ to the odd positions. The above gives class discrepancy of $1/m$ for quantile approximation. Theorems 11 and 12 below generalize this algorithm to any low discrepancy class.

Theorem 11 For any function family $\mathcal{F}$ with a corresponding class discrepancy $D_m = O(c/m)$ there exists an fully-mergeable streaming coreset deterministic algorithm of size $O\left(c \log^2(\epsilon n/c)/\epsilon\right)$ whose error is at most ϵn.

Proof Consider feeding the output of the first compactor into a second one etc. Specifically, we start with a single compactor and open a second once it produced any output, then open a third compactor once the second produced output, etc. Number the compactors $0, \ldots, H$. The weight of items given to compactors h have weight $w_h = 2^h$. The length of the input stream seen by compactor is $n_h \leq n/2^h$.

Each compactor contributes at most $w_h n_h D_m \leq nD_m$ error. Moreover since the $H - 1$ layer had outputs, we must have $m \leq n_{H-1}$ and

$$\log_2(m) \leq \log_2(n_{H-1}) \leq \log_2(n) - (H - 1)$$

leading to a bound $H \leq \lfloor log_2(n/m) \rfloor + 1$. The total error is therefore $HnD_m \leq O(log(n/m)nD_m)$. Setting $m \geq m_0 = O(c \log(\epsilon n/c)/\epsilon)$ and replacing $D_m = c/m$ we get that the error is at most $O(log(n/m)nD_m) \leq \epsilon n$. Since we have $H = O(log(\epsilon n/c))$ compactors the overall space complexity is $O(c \log^2(\epsilon n/c)/\epsilon)$. ■

Theorem 12 For any function family $\mathcal{F}$ with a corresponding class discrepancy $D_m = O(c/\sqrt{m})$ there exists a fully-mergeable streaming coreset deterministic algorithm of size $O\left(c^2 \log^3(\epsilon^2 n/c)/\epsilon^2\right)$ whose error is at most ϵn.

Proof The proof is identical to the one above except for the variable setting of Setting $m \geq m_0 = O(c^2 \log^2(\epsilon^2 n/c^2)/\epsilon^2)$ and replacing $D_m = c/\sqrt{m}$. We get that the error is at most $O(log(n/m)nD_m) \leq \epsilon n$. Since we have $H = O(log(\epsilon^2 n/c^2))$ such compactors the overall space complexity is $O\left(c^2 \log^3(\epsilon^2 n/c^2)/\epsilon^2\right)$. ■

We proceed to prove Theorem 14. To understand the motivation consider first an easier setting where the overall stream length n is known to us in advance. Since $|f(x,q)| \leq 1$, standard concentration bounds will show that by sampling each item w.p. $\log(1/\delta)/n\epsilon^2$ we get an output stream of length $\log(1/\delta)/\epsilon^2$, that for any fixed query q, with probability at least $1-\delta$ suffers an error of ϵn for that query. We can feed this output stream into a deterministic sketch, and given that the input length for the deterministic sketch is $\log(1/\delta)/\epsilon^2$, Theorem 12 leads to the required guarantee.

Because we do not know the stream length in advance, we operate as in the deterministic case with compactors. The difference will be that each compactor will keep a count of how many items it has seen. Once a compactor observed more than $\tilde{n} = O(\log(1/\delta)/\epsilon^2)$ items, it will no longer use a buffer of size m but rather a buffer of size 2. For every two items observed it will output one of them uniformly at random. It is easy to see that a sequence of such compactors can, in fact, be implemented with $O(1)$ memory via reservoir sampling. The memory of this process is therefore identical, at least asymptotically, to the above.

Theorem 14 For any function family $\mathcal{F}$ with a corresponding class discrepancy $D_m = O(c/\sqrt{m})$ there exists a fully-mergeable streaming coreset randomized algorithm of size $O\left(c^2 \log^3 \log(n/\delta)/\epsilon^2\right)$ whose error for any fixed function $f \in \mathcal{F}$ is at most ϵn with probability at least $1-\delta$.

Proof As in the deterministic setting we maintain a sequence of compactors of levels $h = 0, \ldots, H$. Notice that the value of H is increasing as the stream grows longer. Recall that a compactor of level h observes elements of weight 2^h and outputs elements of weight 2^{h+1}. As before we use a buffer of m and get that $H \leq \lfloor log_2(n/m) \rfloor + 1$. The difference is that for a compactor of level h, once $h \leq H' = H - \log(\tilde{n}/m)$, where $\tilde{n} = O(\log(1/\delta)/\epsilon^2)$ with a constant in the $O()$ term that will be determined later, we change the mode of operation for this compactor. Notice that the requirement for h ensures that the number of items observed by the h'th compactor is at least $n_h \geq \tilde{n}$. Rather than using a buffer of size m the compactor uses a buffer of size 2 and for every two observed items, it outputs one of them uniformly at random.

To analyze the memory requirement, notice that the compactors of levels $h = 0, \ldots, H'$ are in fact performing reservoir sampling for every $2^{H'+1}$ items, meaning that they can be implemented in $O(1)$ memory. This means that the overall memory requirement is $O(m \log(\log(1/\delta)/m\epsilon^2))$; for $m \geq 1/\epsilon^2$ this is $O(m \log\log(1/\delta))$.

We continue to bound the error. For the top compactors of level $h = H'+1, \ldots, H$ we get as in the deterministic case that the error for each is nD_m. Since we will use $m \geq 1/\epsilon^2$ we get that the error for all top compactors is $O(nD_m \log\log(1/\delta))$. Consider now a compactor of level $h \leq H'$. For the first $\tilde{n}$ items it observed, the error is bounded by $2^h \tilde{n} D_m \leq 2^{h-H'} nD_m$. Fix a query q; for the items following the first $\tilde{n}$ items the compactor is operating in the sampling mode. For every pair, the associated error w.r.t q is a random variable, of mean zero and absolute value of at most $w_h = 2^{h+1}$. There are $(n_h - \tilde{n})/2 \leq n_h$ such pairs and the overall error w.r.t. q is the sum of these independent random variables. Chernoff bound implies that with probability $1-\delta$, the overall error

is bounded by $E_h = O(w_h\sqrt{n_h \log(1/\delta)})$. Since $n_h \geq \tilde{n}2^{H'-h} = O(2^{H'-h}\log(1/\delta)/\epsilon^2)$ we get that

$$E_h = O\left(2^h n_h \frac{\epsilon}{2^{(H'-h)/2}}\right) = O(\epsilon n 2^{(h-H')/2})$$

We get that the sum of errors associated with the compactors of level $h = 0, \ldots, H'$ form a geometric sequence dominated by the error of the H' compactor, which is in turn $O(\epsilon n)$. For proper constants in $\tilde{n}$ we get a bound of $\epsilon n/2$ for the bottom compactors. For a budget of $m = \Omega(c^2 \log^2(\log(1/\delta))/\epsilon^2)$ for the buffers of the top compactors we guarantee an overall error of $\epsilon n/2$ for the top compactors.

To conclude, we get an error of ϵn w.p. $1-\delta$ for any fixed q with a memory budget of

$$O(m \log\log(1/\delta)) = O\left(c^2 \log^3(\log(1/\delta))/\epsilon^2\right)$$

as required. ■

We are now ready for the proof of Theorem 13. Here we extend the idea of Karnin et al. (2016) applied for quantiles to general coresets. To explain the high-level idea consider again the easier setting where we know n, the length of the stream in advance. As in the $D_m = c/\sqrt{m}$ case, we will split the compactors into the top $\log\log(1/\delta)$ ones acting deterministically and bottom compactors yielding random outputs. The issue comes from the choice of m. To handle the error of the top compactors it suffices to set $m = c/\epsilon \ll 1/\epsilon^2$. The fact that $m \ll 1/\epsilon^2$ means that the top random compactors observe a stream that is shorter than before and having a buffer of size 2 will result in a large error. We can mitigate this by adding $\log(1/\epsilon)$ more deterministic compactors and replace the $\log\log(1/\delta)^2$ term in the memory requirement with $(\log\log(1/\delta)/\epsilon)^2$. If $\log(1/\delta) \gg 1/\epsilon$ then this is a good solution. However, for cases where ϵ is small we can avoid the $\log(1/\epsilon)$ term altogether. To do that, the random compactors will not have a buffer of size 2, but a buffer size of m_h depending on their level. Specifically the sequence of m_h starting from the top random level $h = H - \log\log(1/\delta)$ and ending with $h = 0$ is exponentially decreasing until hitting the minimal buffer size of 2.

The memory requirement is now $O(m)$ and a careful analysis of the error will lead to an ϵn term coming from the bottom layers. One subtle issue we will need to take into account is that for random compactors with budget $m_h > 2$ the output stream is only half as long as the input stream in expectation. Luckily, the output stream length is sharply concentrated around its mean so a union bound can ensure that w.p. $1-\delta$ the output stream is not much longer than its expectation.

Theorem 13 Any function f with class discrepancy $D_m(f) = O(c/m)$ has streaming coreset complexity of $O\left(c \log^2 \log(|Q_\epsilon|/\delta)/\epsilon\right)$. Q_ϵ is an epsilon net for f on $\mathcal{Q}$. The streaming coreset algorithm is randomized and fails with probability at most δ.

Proof We start by describing the algorithm, from the perspective of a compactor of level h. The compactor observes an input stream of items with weight 2^h and outputs a stream of weight 2^{h+1}. When created the compactor has a budget of $m_h = m$. Once it outputs items to an output stream for the first time, a new compactor of level $h+1$ is created. We keep track of H, the level of the top compactor, that did not yet output any items. When H is updated, compactors of level $h < H$ might restrict their budget. Specifically, for some $H' = H - O(\log\log(n/\delta))$ where we set the constant of the $O()$ term later, a compactor of level $h \leq H'$ sets its buffer size to

$$m_h = \max\left\{2, \left\lceil (2/3)^{h-H'} m \right\rceil\right\}$$

compactors of level $h > H'$ have a buffer size of m. We note that although n is present in the definition of H' we can use a crude upper bound. Given that the dependence is doubly logarithmic the upper bound can be extremely crude. Furthermore, δ is typically set to be exponentially small, so we ignore this issue.

Compactors of level $h > H'$ act in a deterministic manner. Namely, once the buffer is full with items $x_1, \ldots, x_m$ we find the sign assignment σ giving $|\max_q \sum \sigma_i f(x_i, q)| < m_h D_{m_h} = m D_m$ and output the smallest of the sets $X_+ = \{x_i | \sigma_i > 0\}$, $X_- = \{x_i | \sigma_i < 0\}$. Compactors of level $h \leq H'$ act in a random manner; they output either the items of X_- or X_+ with equal probability. When the stream is finished the coreset consists of all the items in the buffers, along with their corresponding weight.

Let's begin by analyzing the memory complexity of the algorithm. The top layers each require a buffer of size m, and there are $\log\log(n/\delta)$ such buffers. It follows that they require $O(\log\log(n/\delta)m)$ memory. The bottom layers are exponentially decreasing until hitting $m_h = 2$. All layers with $m_h = 2$ are stacked in a consecutive way so they are in fact doing reservoir sampling and can be implemented with $O(1)$ memory. The layers with $m_h > 2$ are have exponentially growing weights ending at m, so the overall memory they require is $O(m)$. Concluding, the overall memory requirement is $O(\log\log(n/\delta)m)$.

We are now ready to bound the error, starting with the bottom layers. Fix a query q. For a layer h we will provide a high probability bound to both $E_h(q)$, the error associated to its output stream and the length of the output stream. Let n_h be the overall number of items layer h observes. Let m_h be the buffer size of level h at the end of the stream. Since having a larger buffer size only improves the error bound, we analyze the error as if the budget was set as m_h to begin with.

With the assumption of all compactions being done with a buffer of size m_h, the number of compactions is n_h/m_h and the error associated with each compaction is a zero mean random variable, with an absolute value of $2^h m_h D_{m_h}$. The overall error $E_h(q)$ is the sum of these independent random variables. It follows from Chernoff-Hoeffding bound that for any $\epsilon_h > 0$,

$$\Pr\left[E_h(q) > 2^h m_h D_{m_h} \epsilon_h n_h\right] = \exp\left(-\Omega\left(\epsilon_h^2 n_h m_h\right)\right) \tag{1}$$

For a bound on the output length we will analyze the behavior of the compactor with the assumption that all compactions are done to m elements. This is not the case but an upper bound for this scenario also bounds the scenario where m_h is decreasing with time. Every compaction outputs a random number of items between 0 and m, with an expected value of $m/2$. Again, using Chernoff-Hoeffding we get

$$\Pr\left[n_{h+1} > n_h(1/2 + 1/\log(n))\right] = \exp\left(-\Omega\left(\frac{n_h}{m\log^2(n)}\right)\right) \tag{2}$$

To bound this expression we derive a lower bounding on n_h. Notice that the compactors of levels $H'+1, \ldots, H$ are acting in a deterministic manner meaning that

$$n_h \geq n_{H'} \geq 2^{H-H'-1} n_{H-1} \geq 2^{H-H'-1} m = \Omega(\log^2(n)\log(\log(n)/\delta)m)$$

where the constant in the Ω term can be controlled via constant defining H'. Plugging into Equation (2) leads to

$$\Pr\left[n_{h+1} > n_h(1/2 + 1/\log(n))\right] \leq \delta/2(\log_2(n) + 3)$$

A union bound over $h = 0, \ldots, \log_2(n) + 2$ indicates that w.p. $1 - \delta/2$, $n_h \leq 3n/2^h$ for all mentioned h values. In particular this means that $H \leq \log_2(n) + 2$ meaning that

$$\Pr\left[\forall h, n_h \leq 3n/2^h\right] \geq 1 - \delta/2 \tag{3}$$

We can now plug the upper bound for n_h to Equation 1 and achieve

$$\Pr\left[E_h(q) > m_h D_{m_h} \epsilon_h n\right] = \exp\left(-\Omega\left(\frac{\epsilon_h^2 m_h n}{2^h}\right)\right) \tag{4}$$

Recall that $2^{H-H'-1} = \Omega(\log^2(n)\log(\log(n)/\delta))$ and $n \geq 2^{H-1}m$. Combining the two leads to $n = \Omega(2^{H'}\log(n/\delta)m)$. Now, since $m_h \geq (2/3)^{H'-h}m$ we have that

$$\frac{\epsilon_h^2 n m_h}{2^h} = \Omega\left((2/3)^{H'-h}\frac{\epsilon_h^2 2^{H'}\log(n/\delta)m^2}{2^h}\right) = \Omega\left((4/3)^{H'-h}\epsilon_h^2\log(n/\delta)m^2\right)$$

Plugging this into Equation (4), with $\epsilon_h = (3/4)^{h-H'}/m$ and using $m_h D_{m_h} \leq c$ we get

$$\Pr\left[E_h(q) > \frac{(3/4)^{h-H'}c}{m}n\right] \leq \delta/2n \tag{5}$$

Since $H' < n$ we get that w.p. $1 - \delta/2$

$$\sum_{h=1}^{H} E_h(q) \leq (4c/m)n$$

Concluding the analysis for the bottom H' layers, w.p. at least $1 - \delta$ their error is $(4c/m)n$ and the output stream of the H' compactor outputs at most $3n/2^{H'+1}$ items, each having a weight of $2^{H'+1}$. With the length of the output stream we use the fact that the top layers are deterministic and can apply Lemma 25 to bound their error of each of these layers by

$$3D_m n \leq (3c/m)n$$

Since there are $O(\log\log(n/\delta))$ such layers, we conclude that for $m = \Omega(\log\log(n/\delta)c/\epsilon)$ with appropriate constant it holds for a fixed q, w.p. at least $1 - \delta$ that the overall error of the sketch is bounded by ϵn. The resulting memory requirement $O(\log^2\log(n/\delta)c/\epsilon)$, as claimed. ■