

## 49 CORESETS and SKETCHES

Jeff M. Phillips

---

### INTRODUCTION

Geometric data summarization has become an essential tool in both geometric approximation algorithms and where geometry intersects with big data problems. In linear or near-linear time large data sets can be compressed into a summary, and then more intricate algorithms can be run on the summaries whose results approximate those of the full data set. Coresets and sketches are the two most important classes of these summaries.

A *coreset* is a reduced data set which can be used as proxy for the full data set; the same algorithm can be run on the coreset as the full data set, and the result on the coreset approximates that on the full data set. It is often required or desired that the coreset is a subset of the original data set, but in some cases this is relaxed. A *weighted coreset* is one where each point is assigned a weight, perhaps different than it had in the original set. A *weak coreset* associated with a set of queries is one where the error guarantee holds for a query which (nearly) optimizes some criteria, but not necessarily all queries; a *strong coreset* provides error guarantees for all queries.

A *sketch* is a compressed mapping of the full data set onto a data structure which is easy to update with new or changed data, and allows certain queries whose results approximate queries on the full data set. A *linear sketch* is one where the mapping is a linear function of each data point, thus making it easy for data to be added, subtracted, or modified.

These definitions can blend together, and some summaries can be classified as either or both. The overarching connection is that the summary size will ideally depend only on the approximation guarantee but not the size of the original data set, although in some cases logarithmic dependence is acceptable.

We focus on five types of coresets and sketches: shape-fitting (Section 49.1), density estimation (Section 49.2), high-dimensional vectors (Section 49.3), high-dimensional point sets / matrices (Section 49.4), and clustering (Section 49.5). There are many other types of coresets and sketches (e.g., for graphs [AGM12] or Fourier transforms [IKP14]) which we do not cover for space or because they are less geometric.

---

### COMPUTATIONAL MODELS and PRIMITIVES

Often the challenge is not simply to bound the size of a coreset or sketch as a function of the error tolerance, but to also do so efficiently and in a restricted model. So before we discuss the specifics of the summaries, it will be useful to outline some basic computational models and techniques.

The most natural fit is a *streaming model* that allows limited space (e.g., the size of the coreset or sketch) and where the algorithm can only make a single scan over

the data. More generally, data may be distributed across different locations, and it will be useful to first compress the data into a coresets or sketch, and then share only these summaries across sites. Some of the summaries (e.g., linear sketches) can easily be updated with the addition of a new data point or added together. The other most common approach relies on analysis surrounding the merge-reduce paradigm [CM96, BS80], which may require an extra log factor in space over direct methods.

Given two summaries  $S_1$  and  $S_2$  with error bounds  $\varepsilon_1$  and  $\varepsilon_2$  (specific definitions vary widely, and will be discussed subsequently), typically one can show that the *merge* (generically denoted  $\oplus$ , perhaps just the union) of the two summaries  $S = S_1 \oplus S_2$  has an error bound at most  $\varepsilon = \varepsilon_1 + \varepsilon_2$ . Summaries which admit a process  $\oplus$  for which  $S$  has size  $\max\{\text{size}(S_1), \text{size}(S_2)\}$  and has error bound  $\varepsilon = \max\{\varepsilon_1, \varepsilon_2\}$  are said to be *mergeable* [ACH<sup>+</sup>13]. Otherwise, if one summarizes (or *reduces*) a summary  $S$  with error bound  $\varepsilon$  to a smaller summary  $S'$  that incurs additional error  $\varepsilon'$  (compared to  $S$ ), then  $S'$  typically can be shown to have an error bound  $\varepsilon + \varepsilon'$  with respect to the original data.

Putting these *merge* and *reduce* operations together, an efficient merge-reduce framework obtains a summary of size  $g$  (asymptotically, perhaps up to log factors) from a data set of size  $n$ , as follows. First arbitrarily divide the data into  $n/g$  subsets, each of size  $g$  (assume  $n/g$  is a power of 2, otherwise pad the data with dummy points). Then in  $\log(n/g)$  rounds until there is one remaining set, perform each of the next two steps. First pair up all remaining sets, and merge each pair using an  $\oplus$  operator. Second, reduce each remaining set to be a summary of size  $g$ .

Notice that even if the merge or reduce step requires some polynomial  $m^c$  time to process  $m$  data points, this is only applied to sets of size at most  $2g$ , hence the full runtime is dominated by the first round as  $(n/g) \cdot (2g)^c = O(n \cdot g^{c-1})$ . The log factors increases in error can be folded into the size  $g$ , or in many cases removed by delaying some reduce steps, and careful bookkeeping [CM96].

In a streaming setting this framework is adapted by mapping data points to the  $n/g$  subsets in the order they arrive, and then always completing as much of the merge-reduce process as possible given the data seen. Another  $\log(n/g)$  space factor is incurred for those many summaries which can be active at any given time.

---

---

## 49.1 SHAPE FITTING

In this section we will discuss problems where given an input point set  $P$ , the goal is to find the best fitting shape from some class to  $P$ . The two central problems in this area are the minimum (or smallest) enclosing ball, which has useful solutions in high dimensions, and the  $\varepsilon$ -kernel coresets for directional width which approximates the convex hull but also can be transformed to solve many other problems.

---

### GLOSSARY

***minimum enclosing ball (MEB):*** Given a point set  $P \subset \mathbb{R}^d$ , it is the smallest ball  $B$  which contains  $P$ .

***$\varepsilon$ -approximate minimum enclosing ball problem:*** Given a point set  $P \subset \mathbb{R}^d$ , and a parameter  $\varepsilon > 0$ , the problem is to find a ball  $B$  whose radius is no larger

than  $(1 + \varepsilon)$  times the radius of the MEB of  $P$ .

**directional width:** Given a point set  $P \subset \mathbb{R}^d$  and a unit vector  $u \in \mathbb{R}^d$ , then the *directional width* of  $P$  in direction  $u$  is  $\omega(P, u) = \max_{p \in P} \langle p, u \rangle - \min_{p \in P} \langle p, u \rangle$ .

**$\varepsilon$ -kernel coreset:** An  $\varepsilon$ -kernel coreset of a point set  $P \subset \mathbb{R}^d$  is subset  $Q \subset P$  so that for all unit vectors  $u \in \mathbb{R}^d$

$$0 \leq \omega(P, u) - \omega(Q, u) \leq \varepsilon \omega(P, u).$$

**functional width:** Given a set  $\mathcal{F} = \{f_1, \dots, f_n\}$  of functions each from  $\mathbb{R}^d$  to  $\mathbb{R}$ , the width at a point  $x \in \mathbb{R}^d$  is defined  $\omega_{\mathcal{F}}(x) = \max_{f_i \in \mathcal{F}} f_i(x) - \min_{f_i \in \mathcal{F}} f_i(x)$ .

**$\varepsilon$ -kernel for functional width:** Given a set  $\mathcal{F} = \{f_1, \dots, f_n\}$  of functions each from  $\mathbb{R}^d$  to  $\mathbb{R}$ , an  $\varepsilon$ -kernel coreset is a subset  $\mathcal{G} \subset \mathcal{F}$  such that for all  $x \in \mathbb{R}^d$  the functional width  $\omega_{\mathcal{G}}(x) \geq (1 - \varepsilon)\omega_{\mathcal{F}}(x)$ .

**faithful measure:** A measure  $\mu$  is faithful if there exists a constant  $c$ , depending on  $\mu$ , such that for any point set  $P \subset \mathbb{R}^d$  any  $\varepsilon$ -kernel coreset  $Q$  of  $P$  is a coreset for  $\mu$  with approximation parameter  $c\varepsilon$ .

**diameter:** The diameter of a point set  $P$  is  $\max_{p, p' \in P} \|p - p'\|$ .

**width:** The width of a point set  $P$  is  $\min_{u \in \mathbb{R}^d, \|u\|=1} \omega(P, u)$ .

**spherical shell:** For a point  $c \in \mathbb{R}^d$  and real numbers  $0 \leq r \leq R$ , it is the closed region  $\sigma(c, r, R) = \{x \in \mathbb{R}^d \mid r \leq \|x - c\| \leq R\}$  between two concentric spheres of radius  $r$  and  $R$  centered at  $c$ . Its *width* is defined  $R - r$ .

### Smallest Enclosing Ball Coreset

Given a point set  $P \subset \mathbb{R}^d$  of size  $n$ , there exists a  $\varepsilon$ -coreset for the smallest enclosing ball problem of size  $\lceil 2/\varepsilon \rceil$  that runs in time  $O(nd/\varepsilon + 1/\varepsilon^5)$  [BC03b]. Precisely, this finds a subset  $S \subset P$  with smallest enclosing ball  $B(S)$  described by center point  $c$  and radius  $r$ ; it holds that if  $r$  is expanded to  $(1 + \varepsilon)r$ , then the ball with the same center would contain  $P$ .

The algorithm is very simple and iterative: At each step, maintain the center  $c_i$  of the current set  $S_i$ , add to  $S_i$  the point  $p_i \in P$  furthest from  $c_i$ , and finally update  $S_{i+1} = S_i \cup \{p_i\}$  and  $c_{i+1}$  as the center of smallest enclosing ball of  $S_{i+1}$ . Clarkson [Cla10] discusses the connection to the Frank-Wolfe [FW56] algorithm, and the generalizations towards several sparse optimization problems relevant for machine learning, for instance support vector machines [TKC05] and polytope distance [GJ09].

These algorithms do not work in a streaming setting, as they require  $\Omega(1/\varepsilon)$  passes over the data, but the runtime can be improved to  $O((d/\varepsilon + n/\varepsilon^2) \log(n/\varepsilon))$  with high probability [CHW10]. Another approach [AS15] maintains a set of  $O((1/\varepsilon^3) \log(1/\varepsilon))$  points in a stream that handles updates in  $O((d/\varepsilon^2) \log(1/\varepsilon))$  time, but is not a pure coreset since in order to handle updates, it needs to maintain these points as  $O((1/\varepsilon^2) \log(1/\varepsilon))$  different groups.

### Epsilon-Kernels Coreset for Width

Given point sets  $P \subset \mathbb{R}^d$  of size  $n$ , an  $\varepsilon$ -kernel coreset for directional width exists of size  $O(1/\varepsilon^{(d-1)/2})$  [AHPV04] and can be constructed in  $O(n + 1/\varepsilon^{d-(3/2)})$

time [Cha06, YAPV08]. These algorithms are quite different than those for MEB, and the constants have heavy dependence on  $d$  (in addition to it being in the exponent of  $1/\varepsilon$ ). They first estimate the rough shape of the points so that they can be made fat (so width and diameter are  $\Theta(1)$ ) through an affine transform that does not change which points form a coreset. Then they carefully choose a small set of points in the extremal directions.

In a stream in  $\mathbb{R}^d$ , the  $\varepsilon$ -kernel coreset can be computed using  $O((1/\varepsilon^{(d-1)/2}) \cdot \log(1/\varepsilon))$  space with  $O(1 + (1/\varepsilon^{(d-3)/2}) \log(1/\varepsilon))$  update time, which can be amortized to  $O(1)$  update time [ZZ08]. In  $\mathbb{R}^2$  this can be reduced to  $O(1/\sqrt{\varepsilon})$  space and  $O(1)$  update time [YA07].

Similar to  $\varepsilon$ -kernels for directional width, given a set of  $n$   $d$ -variate linear functions  $\mathcal{F}$  and a parameter  $\varepsilon$ , then an  $\varepsilon$ -kernel for functional width can be computed of size  $O(1/\varepsilon^{d/2})$  in time  $O(n + 1/\varepsilon^{d-(1/2)})$  [AHPV04, Cha06].

Many other measures can be shown to have  $\varepsilon$ -approximate coresets by showing they are *faithful*; this includes diameter, width, minimum enclosing cylinder, and minimum enclosing box. Still other problems can be given  $\varepsilon$ -approximate coresets by linearizing the inputs so they represent a set of  $n$  linear functions in higher dimensions. Most naturally this works for creating an  $\varepsilon$ -kernel for the width of polynomial functions. Similar linearization is possible for a slew of other shape-fitting problems including the minimum width spherical shell problem, overviewed nicely in a survey by Agarwal, Har-Peled and Varadarajan [AHPV07].

These coresets can be extended to handle a small number of outliers [HPW04, AHPY08] or uncertainty in the input [LPW14]. A few approaches also extend to high dimensions, such as fitting a  $k$ -dimensional subspace [HPV04, BHPR16].

---

---

## 49.2 DENSITY ESTIMATION

Here we consider a point set  $P \in \mathbb{R}^d$  which represents a discrete density function. A coreset is then a subset  $Q \subset P$  such that  $Q$  represents a similar density function to  $P$  under a restricted family of ways to measure the density on subsets of the domain, e.g., defined by a range space.

---

### GLOSSARY

**range space:** A range space  $(P, \mathcal{A})$  consists of a ground set  $P$  and a family of ranges  $\mathcal{R}$  of subsets from  $P$ . In this chapter we consider ranges which are defined geometrically, for instance when  $P$  is a point set and  $\mathcal{R}$  are all subsets defined by a ball, that is any subset of  $P$  which coincides with  $P \cap B$  for any ball  $B$ .

**$\varepsilon$ -net:** Given a range space  $(P, \mathcal{R})$ , it is a subset  $Q \subset P$  so for any  $R \in \mathcal{R}$  such that  $|R \cap P| \geq \varepsilon|P|$ , then  $R \cap Q \neq \emptyset$ .

**$\varepsilon$ -approximation (or  $\varepsilon$ -sample):** Given a range space  $(P, \mathcal{R})$ , it is a subset  $Q \subset P$  so for all  $R \in \mathcal{R}$  it implies  $\left| \frac{|R \cap P|}{|P|} - \frac{|R \cap Q|}{|Q|} \right| \leq \varepsilon$ .

**VC-dimension:** For a range space  $(P, \mathcal{R})$  it is the size of the largest subset  $Y \subset P$  such that for each subset  $Z \subset Y$  it holds that  $Z = Y \cap R$  for some  $R \in \mathcal{R}$ .

---

## RANDOM SAMPLING BOUNDS

Unlike the shape fitting coresets, these can be constructed by simply selecting a large enough random sample of  $P$ . The best such size bounds typically depend on VC-dimension  $\nu$  [VC71] (or shattering dimension  $\sigma$ ), which for many geometrically defined ranges (e.g., by balls, halfspaces, rectangles) is  $\Theta(d)$ . A random subset  $Q \subset P$  of size  $O((1/\varepsilon^2)(\nu + \log(1/\delta)))$  [LLS01] is an  $\varepsilon$ -approximation of any range space  $(P, \mathcal{R})$  with VC-dimension  $\nu$ , with probability at least  $1 - \delta$ . A subset  $Q \subset P$  of size  $O((\nu/\varepsilon) \log(1/\varepsilon\delta))$  [HW87] is an  $\varepsilon$ -net of any range space  $(P, \mathcal{R})$  with VC-dimension  $\nu$ , with probability at least  $1 - \delta$ .

These bounds are of broad interest to learning theory, because they describe how many samples are needed to learn various sorts of classifiers. In machine learning, it is typical to assume each data point  $q \in Q$  is drawn iid from some unknown distribution, and since the above bounds have no dependence on  $n$ , we can replace  $P$  by any probability distribution with domain  $\mathbb{R}^d$ . Consider that each point in  $Q$  has a value from  $\{-, +\}$ , and a separator range (e.g., a halfspace) should ideally have all  $+$  points inside, and all  $-$  points outside. Then for an  $\varepsilon$ -approximation  $Q$  of a range space  $(P, \mathcal{A})$  the range  $R \in \mathcal{R}$  which misclassifies the fewest points on  $Q$ , misclassifies at most an  $\varepsilon$ -fraction of points in  $P$  more than the optimal separator does. An  $\varepsilon$ -net (which requires far fewer samples) can make the same claim as long as there exists a separator in  $\mathcal{A}$  that has zero misclassified points on  $P$ ; it was recently shown [Han15] a weak coreset for this problem only requires  $O((1/\varepsilon)(\nu + \log(1/\delta)))$  samples.

The typical  $\varepsilon$ -approximation bound provides an additive error of  $\varepsilon$  in estimating  $|R \cap P|/|P|$  with  $|R \cap Q|/|Q|$ . One can achieve a stronger *relative*  $(\rho, \varepsilon)$ -approximation such that

$$\max_{R \in \mathcal{R}} \left| \frac{|R \cap P|}{|P|} - \frac{|R \cap Q|}{|Q|} \right| \leq \varepsilon \max \left\{ \rho, \frac{|R \cap P|}{|P|} \right\}.$$

This requires  $O((1/\rho\varepsilon^2)(\nu \log(1/\rho) + \log(1/\delta)))$  samples [LLS01, HPS11] to succeed with probability at least  $1 - \delta$ .

---

## DISCREPANCY-BASED RESULTS

The basic idea is to build a coloring on the ground set  $\chi : X \rightarrow \{-1, +1\}$  to minimize  $\sum_{x \in R} \chi(x)$  over all ranges. Then we can plug this into the merge-reduce process where merging takes the union and reducing discards the points colored  $-1$ . Chazelle and Matousek [CM96] showed how slight modifications of the merge-reduce framework can remove extra log factors in the approximation.

Based on discrepancy results (see Chapters 14 and 48) we can achieve the following bounds. These assume  $d \geq 2$  is a fixed constant, and is absorbed in  $O(\cdot)$  notation. For any range space  $(P, \mathcal{R})$  with VC-dimension  $\nu$  (a fixed constant) we can construct an  $\varepsilon$ -approximation of size  $g = O(1/\varepsilon^{2-\nu/(\nu+1)})$  in  $O(n \cdot g^{w-1})$  time. This is tight for range spaces  $\mathcal{H}_d$  defined by halfspaces in  $\mathbb{R}^d$ , where  $\nu = d$ . For range spaces  $\mathcal{B}_d$  defined by balls in  $\mathbb{R}^d$ , where  $\nu = d+1$  this can be improved slightly to  $g = O(1/\varepsilon^{2-\nu/(\nu+1)} \sqrt{\log(1/\varepsilon)})$ ; it is unknown if the log factor can be removed. For range spaces  $\mathcal{T}_d$  defined by axis-aligned rectangles in  $\mathbb{R}^d$ , where  $\nu = 2d$ , this can be greatly improved to  $g = O((1/\varepsilon) \log^{d+1/2}(1/\varepsilon))$  with the best lower bound

as  $g = O((1/\varepsilon) \log^{d-1}(1/\varepsilon))$  for  $d \geq 2$  [Lar14, MNT15]. These colorings can be constructed adapting techniques from Bansal [Ban10, BS13]. Generalization to various other settings (typically following one of these patterns) can be found in books by Matousek [Mat10] and Chazelle [Cha00]. Similar bounds are available in a stream adapting the merge-reduce framework [BCEG04, STZ04].

Discrepancy based results also exist for constructing  $\varepsilon$ -nets. However often the improvement over the random sampling bounds are not as dramatic; the main advantage is that the algorithms are deterministic. For halfspaces in  $\mathbb{R}^3$  and balls in  $\mathbb{R}^2$  we can construct  $\varepsilon$ -nets of size  $O(1/\varepsilon)$  [MSW90, CV07, PR08]. For axis-aligned rectangles and fat objects we can construct  $\varepsilon$ -nets of size  $O((1/\varepsilon) \log \log(1/\varepsilon))$  [AES10]. Pach and Tardos [PT13] then showed these results are tight, and that similar improvements cannot exist in higher dimensions.

---

## GENERALIZATIONS

One can also replace the set of ranges  $\mathcal{R}$  with a family of kernels  $\mathcal{K} = \{K(x, \cdot) \mid x \in \mathbb{R}^d\}$  where  $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1]$ ; that is it replaces the binary  $\{0, 1\}$  notion of inclusion with a continuous map  $[0, 1]$ . A family of kernels  $\mathcal{K}$  is *linked* to a range space  $(P, \mathcal{R})$  if for every value  $\tau \in [0, 1]$  and every kernel  $K(x, \cdot)$ , the points  $\{p \in P \mid K(x, p) \geq \tau\} = R \cap P$  for some  $R \in \mathcal{R}$ . For instance, centrally symmetric, shift-invariant kernels such as Gaussians are linked to the ranges space defined by balls. Then an  $\varepsilon$ -approximation  $Q$  for  $(P, \mathcal{R})$  is also an  $\varepsilon$ -approximation for  $(P, \mathcal{K})$  [JKP11], meaning that

$$\max_{x \in \mathbb{R}^d} \left| \frac{\sum_{p \in P} K(x, p)}{|P|} - \frac{\sum_{q \in Q} K(x, q)}{|Q|} \right| \leq \varepsilon.$$

Surprisingly, the discrepancy-based  $\varepsilon$ -approximations are smaller for kernels than balls, for instance requiring only  $O((1/\varepsilon) \sqrt{\log(1/\varepsilon)})$  points in  $\mathbb{R}^2$  [Phi13]. Similarly an  $(\varepsilon - \tau)$ -net  $Q$  for  $(P, \mathcal{R})$  is also an  $(\tau, \varepsilon)$ -net for  $(P, \mathcal{K})$  [PZ15], meaning that for all  $x \in \mathbb{R}^d$  such that  $\frac{\sum_{p \in P} K(x, p)}{|P|} \geq \varepsilon$  then there exists some  $q \in Q$  such that  $K(x, q) \geq \tau$ .

Similarly we can consider a single function  $f \in \mathcal{F}$  [HP06] or smaller set of  $k$  functions  $\{f_1, \dots, f_k\} \subset \mathcal{F}$  [LS10] for various well-behaved function classes  $\mathcal{F}$ , and create a weighted coresets  $Q \subset P$ , with weight function  $w : Q \rightarrow \mathbb{R}$ , such that  $\sum_{q \in Q} \min_{i \in [k]} w(q) f_i(q)$  approximates  $\sum_{p \in P} \min_{i \in [k]} f_i(p)$ . These can be extended to handle outliers and robust loss functions like Huber and Tukey [FS12] and Gaussian mixture models [FFK11], and as we will see, these techniques build the foundation for coresets for clustering [FL11]. These approaches can probably be extended to handle some classes of kernels by using  $\mathcal{F} = \{f(\cdot) = 1 - K(x, \cdot) \mid K(x, \cdot) \in \mathcal{K}\}$ .

---

## QUANTILES SKETCH

A popular sketch for the so-called quantiles problem maps to 1-dimensional  $\varepsilon$ -approximations. Define the *rank* of  $v$  for set  $X \in \mathbb{R}$  as  $\text{rank}(X, v) = |\{x \in X \mid x \leq v\}|$ . A quantiles sketch  $S$  over a data set  $X$  of size  $n$  allows for queries such that  $|S(v) - \text{rank}(X, v)| \leq \varepsilon n$  for all  $v \in \mathbb{R}$ . This is equivalent to an  $\varepsilon$ -approximation of a range space  $(X, \mathcal{J})$  where  $\mathcal{J}$  is defined by half-open intervals of the form  $(-\infty, a]$ .

A  $\varepsilon$ -approximation coresets of size  $1/\varepsilon$  can be found by simply sorting  $X$  and taking evenly spaced points in that sorted ordering. Streaming sketches are also known;

most famously the Greenwald-Khanna sketch [GK01] which takes  $O((1/\varepsilon) \log(\varepsilon n))$  space, where  $X$  is size  $n$ . Recently a streaming sketch of size  $O((1/\varepsilon) \log(1/\varepsilon))$  was discovered [FO15] which is optimal.

---



---

## 49.3 HIGH DIMENSIONAL VECTORS

In this section we will consider high dimensional vectors  $v = (v_1, v_2, \dots, v_d)$ . When each  $v_i$  is a positive integer, we can imagine these as the counts of a labeled set (the  $d$  dimensions), and a subset of these objects is a coreset which can approximate the relative frequencies. Hence, we can consider weighted coresets over the dimensions, and consider approximating  $v$  with another vector with few non-zero coordinates. Even more generally, a sketch will compactly represent another vector  $u$  which behaves similarly to  $v$  under various norms.

---

### GLOSSARY

**$\ell_p$ -norm:** For a vector  $v \in \mathbb{R}^d$  the  $\ell_p$  norm, for  $p \in [1, \infty)$ , is defined  $\|v\|_p = (\sum_{i=1}^d |v_i|^p)^{1/p}$ . For  $p = 0$  define  $\|v\|_0 = |\{i \mid v_i \neq 0\}|$ , the number of non-zero coordinates, and for  $p = \infty$  define  $\|v\|_\infty = \max_{i=1}^d |v_i|$ .

**$k$ -sparse:** A vector is  $k$ -sparse if  $\|v\|_0 \leq k$ .

**additive  $\ell_p/\ell_q$  approximation:** A vector  $v$  has an additive  $\varepsilon$ - $(\ell_p/\ell_q)$  approximation with vector  $u$  if  $\|v - u\|_p \leq \varepsilon \|v\|_q$ .

**$k$ -sparse  $\ell_p/\ell_q$  approximation:** A vector  $v$  has a  $k$ -sparse  $\varepsilon$ - $(\ell_p/\ell_q)$  approximation with vector  $u$  if  $u$  is  $k$ -sparse and  $\|v - u\|_p \leq \varepsilon \|v - u\|_q$ .

**frequency count:** For a vector  $v = (v_1, v_2, \dots, v_d)$  the value  $v_i$  is called the  $i$ th frequency count of  $v$ .

**frequency moment:** For a vector  $v = (v_1, v_2, \dots, v_d)$  the value  $\|v\|_p$  is called the  $p$ th frequency moment of  $v$ .

---

### FREQUENCY APPROXIMATION

There are several types of coresets and sketches for frequency counts. Derived by  $\varepsilon$ -approximation and  $\varepsilon$ -net bounds, we can create the following coresets over dimensions. Assume  $v$  has positive integer coordinates, and each coordinate's count  $v_i$  represents  $v_i$  distinct objects. Then let  $S$  be a random sample of size  $k$  of these objects and  $u(S)$  be an approximate vector defined so  $u(S)_i = (\|v\|_1/k) \cdot |\{s \in S \mid s = i\}|$ . Then with  $k = O((1/\varepsilon^2) \log(1/\delta))$  we have  $\|v - u(S)\|_\infty \leq \varepsilon \|v\|_1$  (an additive  $\varepsilon$ - $(\ell_\infty/\ell_1)$  approximation) with probability at least  $1 - \delta$ . Moreover, if  $k = O((1/\varepsilon) \log(1/\varepsilon\delta))$  then for all  $i$  such that  $v_i \geq \varepsilon \|v\|_1$ , then  $u(S)_i \neq 0$ , and we can then measure the true count to attain a weighted coreset which is again an additive  $\varepsilon$ - $(\ell_\infty/\ell_1)$  approximation. And in fact, there can be at most  $1/\varepsilon$  dimensions  $i$  with  $v_i \geq \varepsilon \|v\|_1$ , so there always exists a weighted coreset of size  $1/\varepsilon$ .

Such a weighted coreset for additive  $\varepsilon$ - $(\ell_\infty/\ell_1)$  approximations that is  $(1/\varepsilon)$ -sparse can be found deterministically in a stream via the Misra-Gries sketch [MG82]

(or other rediscovered variants [MAA06, DLOM02, KSP03]). This approach keeps  $1/\varepsilon$  counters with associated labels. For a new item, if it matches a label, the counter is incremented, else if any counter is 0 it takes over that counter/label, and otherwise, (perhaps unintuitively) all counters are decremented.

The count-min sketch [CM05] also provides an additive  $\varepsilon\text{-}(\ell_\infty/\ell_1)$  approximation with space  $O((1/\varepsilon)\log(1/\delta))$  and is successful with probability  $1 - \delta$ . A count-sketch [CCFC02] provides an additive  $\varepsilon\text{-}(\ell_\infty/\ell_2)$  approximation with space  $O((1/\varepsilon^2)\log(1/\delta))$ , and is successful with probability at least  $1 - \delta$ . Both of these linear sketches operate by using  $O(\log 1/\delta)$  hash functions, each mapping  $[d]$  to one of  $O(1/\varepsilon)$  or  $O(1/\varepsilon^2)$  counters. The counters are incremented or decremented with the value  $v_i$ . Then an estimate for  $v_i$  can be recovered by examining all cells where  $i$  hashes; the effect of other dimensions which hash there can be shown bounded.

**Frequency moments.** Another common task is to approximate the frequency moments  $\|v\|_p$ . For  $p = 1$ , this is simply the count and can be done exactly in a stream. The AMS Sketch [AMS99] maintains a sketch of size  $O((1/\varepsilon^2)\log(1/\delta))$  that can derive a value  $\hat{F}_2$  so that  $|\|v\|_2 - \hat{F}_2| \leq \varepsilon\|v\|_2$  with probability at least  $1 - \delta$ .

The FM Sketch [FM85] (and its extensions [AMS99, DF03]) show how to create a sketch of size  $O((1/\varepsilon^2)\log(1/\delta))$  which can derive an estimate  $\hat{F}_0$  so that  $|\|v\|_0 - \hat{F}_0| \leq \varepsilon\|v\|_1$  with probability at least  $1 - \delta$ . This works when  $v_i$  are positive counts, and those counts are incremented one at a time in a stream. Usually these sketches and coresets have implicit assumptions that a “word” can fit  $\log n$  bits where the stream is of size  $n$ , and is sufficient for each counter. On the other hand, these  $\ell_0$  sketches operate with bits, and only have a hidden  $\log \log n$  factor for bits.

**$k$ -sparse tail approximation.** Some sketches can achieve  $k$ -sparse approximations (which are akin to coresets of size  $k$ ) and have stronger error bounds that depend only on the “tail” of the matrix; this is the class of  $k$ -sparse  $\varepsilon\text{-}(\ell_p/\ell_q)$  approximations. See survey by Gilbert and Indyk for more details [GI10].

These bounds are typically achieved by increasing the sketch size by a factor  $k$ , and then the  $k$ -sparse vector is the top  $k$  of those elements. The main recurring argument is roughly as follows: If you maintain the top  $1/\varepsilon$  counters, then the largest counter not maintained is of size at most  $\varepsilon\|v\|$ . Similarly, if you first remove the top  $k$  counters (a set  $K = \{i_1, i_2, \dots, i_k\} \subset [d]$ , let their collective norm be  $\|v_K\|$ ), then maintain  $1/\varepsilon$  more, the largest not-maintained counter is at most  $\varepsilon(\|v\| - \|v_K\|)$ . The goal is then to sketch a  $k$ -sparse vector which approximates  $v_K$ ; for instance the Misra-Gries Sketch [MG82] and Count-Min sketch [CM05] achieve  $k$ -sparse  $\varepsilon\text{-}(\ell_1/\ell_1)$ -approximations with  $O(k/\varepsilon)$  counters, and the Count sketch [CCFC02] achieves  $k$ -sparse  $\varepsilon\text{-}(\ell_2/\ell_2)$ -approximations with  $O(k^2/\varepsilon^2)$  counters [BCIS10].

---

---

## 49.4 HIGH DIMENSIONAL POINT SETS (MATRICES)

Matrix sketching has gained a lot of interest due to its close connection to scalability issues in machine learning and data mining. The goal is often to replace a matrix with a small space and low-rank approximation. However, given a  $n \times d$  matrix  $A$ , it can also be imagined as  $n$  points each in  $\mathbb{R}^d$ , and the span of a rank- $k$  approximation



is a  $k$ -dimensional subspace that approximately includes all of the points.

Many of the techniques build on approaches from vector sketching, and again, many of the sketches are naturally interpretable as weighted coresets. Sometimes it is natural to represent the result as a reduced set of rows in a  $\ell \times d$  matrix  $B$ . Other times it is more natural to consider the dimensionality reduction problem where the goal is an  $n \times c$  matrix  $S$ , and sometimes you do both! But since these problems are typically phrased in terms of matrices, the difference comes down to simply transposing the input matrix.

We will write all results as approximating an  $n \times d$  matrix  $A$  using fewer rows, for instance, with an  $\ell \times d$  matrix  $B$ . This allows one to answer questions about a unit vector  $x \in \mathbb{R}^d$ . Geometrically, we can preserve the variation along direction  $x$  ( $\|Ax\|^2$ ) and how close is  $x$  to the best rank  $k$  subspace of  $A$ .

Notoriously this problem can be solved optimally using the numerical linear algebra technique the *singular value decomposition* in  $O(nd^2)$  time. The challenges are then to compute this more efficiently in streaming and other related settings.

We will describe three basic approaches (row sampling, random projections, and iterative SVD variants), and then some extensions and applications [FT15]. The first two approaches are mainly randomized, and we will describe results with constant probability, and for the most part these bounds can be made to succeed with any probability  $1 - \delta$  by increasing the size by a factor  $\log(1/\delta)$ .

---

## GLOSSARY

**matrix rank:** The *rank* of an  $n \times d$  matrix  $A$ , denoted  $\text{rank}(A)$ , is the smallest  $k$  such that all rows (or columns) lie in a  $k$ -dimensional subspace of  $\mathbb{R}^d$  (or  $\mathbb{R}^n$ ).

**singular value decomposition:** Given an  $n \times d$  matrix  $A$ , the singular value decomposition is a product  $USV^T$  where  $U$  and  $V$  are orthogonal, and  $S$  is diagonal.  $U$  is  $n \times n$ , and  $V$  is  $d \times d$ , and  $S = \text{diag}(s_1, s_2, \dots, s_{\min\{n,d\}})$  (padded with either  $n - d$  rows or  $d - n$  columns of all 0s, so  $S$  is  $n \times d$ ) where  $s_1 \geq s_2 \geq \dots \geq s_{\min\{n,d\}} \geq 0$ , and  $s_i = 0$  for all  $i > \text{rank}(A)$ .

The  $i$ th column of  $U$  (resp. column of  $V$ ) is called the  $i$ th left (resp. right) *singular vector*; and  $s_i$  is the  $i$ th *singular value*.

**spectral norm:** The spectral norm of matrix  $A$  is denoted  $\|A\|_2 = \max_x \|Ax\|/\|x\|$ .

**Frobenius norm:** The Frobenius norm of a matrix  $A$  is  $\|A\|_F = \sqrt{\sum_{i=1}^n \|a_i\|^2}$  where  $a_i$  is the  $i$ th row of  $A$ .

**low rank approximation of a matrix:** The best rank  $k$  approximation of a matrix  $A$  is denoted  $[A]_k$ . Let  $S_k$  be the matrix  $S$  (the singular values from the SVD of  $A$ ) where the singular values  $s_i$  are set to 0 for  $i > k$ . Then  $[A]_k = US_kV^T$ . Note we can also ignore the columns of  $U$  and  $V$  after  $k$ ; these are implicitly set to 0 by multiplication with  $s_i = 0$ .

The  $n \times d$  matrix  $[A]_k$  is optimal in that over all rank  $k$  matrices  $B$  it minimizes  $\|A - B\|_2$  and  $\|A - B\|_F$ .

**projection:** For a subspace  $F \subset \mathbb{R}^d$  and point  $x \in \mathbb{R}^d$ , define the projection  $\pi_F(x) = \arg \min_{y \in F} \|x - y\|$ . For a  $n \times d$  matrix  $A$ , then  $\pi_F(A)$  defines the  $n \times d$  matrix where each row is individually projected on to  $F$ .

## ROW SUBSET SELECTION

The first approach towards these matrix sketches is to choose a careful subset of the rows (note: the literature in this area usually discusses selecting columns). The first analysis of these techniques considered sampling  $\ell = O((1/\varepsilon^2)k \log k)$  rows proportional to their squared norm as  $\ell \times d$  matrix  $B$ , and showed [FKV04, DFK<sup>+</sup>04, DKM06] one could describe a rank- $k$  matrix  $P = [\pi_B(A)]_k$  so that

$$\|A - P\|_F^2 \leq \|A - [A]_k\|_F^2 + \varepsilon \|A\|_F^2 \quad \text{and} \quad \|A - P\|_2^2 \leq \|A - [A]_k\|_2^2 + \varepsilon \|A\|_F^2.$$

This result can be extended for sampling columns in addition to rows.

This bound was then improved by sampling proportional to the *leverage scores*; If  $U_k$  is the  $n \times k$  matrix of the top  $k$  left singular values of  $A$ , then the leverage score of row  $i$  is  $\|U_k(i)\|^2$ , the norm of the  $i$ th row of  $U_k$ . In this case  $O((1/\varepsilon^2)k \log k)$  rows are needed to achieve a relative error bound [DMM08]

$$\|A - \pi_B(A)\|_F \leq (1 + \varepsilon) \|A - [A]_k\|_F^2.$$

These results can be extended to sample rows and columns, generating a so-called CUR decomposition of  $A$ . Also, similar relative error bounds can be achieved through volume sampling [DV06].

Although this matrix approximation preserves the structure of rows of the matrix, to get the best bounds via sampling, the sampling probabilities based on leverage scores are as expensive to compute as the full SVD. Instead one can approximate the leverage scores [DMIMW12, CLM<sup>+</sup>15], for instance in a stream in  $O((kd/\varepsilon^2) \log^4 n)$  bits of space [DMIMW12].

Better algorithms exist in a non-streaming settings [FVR15]; these can, for instance, achieve the strong relative error bounds with only  $O(k/\varepsilon)$  rows (and  $O(k/\varepsilon)$  columns) and only require time  $O(\text{nnz}(A) \log n + n \text{poly}(\log n, k, 1/\varepsilon))$  time where  $\text{nnz}(A)$  is the number of non-zero entries in  $A$  [BDMI14, BW14].

---

## RANDOM PROJECTIONS

The second approach to matrix sketching is based on the Johnson-Lindenstrauss (JL) Lemma [JL84], which says that projecting any vector  $x$  (independent of its dimension, for which it will be useful here to denote as  $n$ ) onto a random subspace  $F$  of dimension  $\ell = O(1/\varepsilon^2)$  preserves its norm up to  $(1 + \varepsilon)$  relative error, after rescaling:  $(1 - \varepsilon)\|x\| \leq \sqrt{n/\ell} \|\pi_F(x)\| \leq (1 + \varepsilon)\|x\|$ . Follow up work has shown that the projection operator  $\pi_F$  can be realized as an  $\ell \times n$  matrix  $S$  so that  $(\sqrt{n/\ell})\pi_F(x) = Sx$ . And in particular, we can fill the entries of  $S$  iid with Gaussian random variables [DG03], uniform  $\{-1, +1\}$  or  $\{-1, 0, +1\}$  random variables [Ach03], or any subgaussian random variable [Mat08], rescaled. Alternatively, we can make  $S$  all 0s except for one [CW13] or a constant number [NN13] of uniform  $\{-1, +1\}$  random variables in each column of  $S$ . The later construction essentially “hashes” each element of  $A$  to one or a constant number of elements in  $\pi_F(x)$ ; basically an extension of the count-sketch [CCFC02].

To apply these results to matrix sketching, we simply apply the sketch matrix  $S$  to  $A$  instead of just a single “dimension” of  $A$ . Then  $B = SA$  is our resulting  $\ell \times d$  matrix. However, unlike in typical uses of the JL Lemma on a point set of size  $m$ , where it can be shown to preserve all distances using  $\ell = O((1/\varepsilon^2) \log m)$  target

dimensions, we will strive to preserve the norm over all  $d$  dimensions. As such we use  $\ell = O(d/\varepsilon^2)$  for iid JL results [Sar06], or  $\ell = O(d^2/\varepsilon^2)$  for hashing-based approaches [CW13, NN13]. As was first observed by Sarlos [Sar06] this allows one to create an oblivious subspace embedding so that for *all*  $x \in \mathbb{R}^d$  guarantees

$$(1 - \varepsilon)\|Ax\|_2^2 \leq \|SAx\|_2^2 \leq (1 + \varepsilon)\|Ax\|_2^2.$$

The obliviousness of this linear projection matrix  $S$  (it is created independent of  $A$ ) is very powerful. It means this result can be not only performed in an update-only stream, but also one that allows deletions, or arbitrary updates to an individual entry in a matrix. Moreover, given a matrix  $A$  with only  $\text{nnz}(A)$  non-zero entries, it can be applied in roughly  $O(\text{nnz}(A))$  time [CW13, NN13]. It also implies bounds for matrix multiplication, and as we will discuss, linear regression.

---

## FREQUENT DIRECTIONS

This third class of matrix sketching algorithms tries to more directly replicate those properties of the SVD, and can be deterministic. So why not just use the SVD? These methods, while approximate, are faster than SVD, can be used in a stream, and can sketch two disjoint parts of a matrix then combine the sketches together.

The Frequent Directions algorithm [Lib13, GLPW15] essentially processes each row (or  $O(\ell)$  rows) of  $A$  at a time, always maintaining the best rank- $\ell$  approximation as the sketch. But this can suffer from data drift, so crucially after each such update, it also shrinks all squared singular values of  $[B]_\ell$  by  $s_\ell^2$ ; this ensures that the additive error is never more than  $\varepsilon\|A\|_F^2$ , precisely as in the Misra-Gries [MG82] sketch for frequency approximation. Setting  $\ell = k + 1/\varepsilon$  and  $\ell = k + k/\varepsilon$ , respectively, the following bounds have been shown [GP14, GLPW15] for any unit vector  $x$ :

$$0 \leq \|Ax\|^2 - \|Bx\|^2 \leq \varepsilon\|A - [A]_k\|_F^2 \quad \text{and} \quad \|A - \pi_{[B]_k}(A)\|_F^2 \leq (1 + \varepsilon)\|A - [A]_k\|_F^2.$$

Operating in a batch to process  $\Theta(\ell)$  rows at a time, this takes  $O(nd\ell)$  time. A similar approach by Feldman *et.al.* [FSS13] provides a more general bound, and will be discussed in the context of subspace clustering next.

---

## LINEAR REGRESSION and ROBUST VARIANTS

The regression problem takes as input again an  $n \times d$  matrix  $A$  and also an  $n \times w$  matrix  $T$  (most commonly  $w = 1$  and  $T$  is a vector); the goal is to find the  $d \times w$  matrix  $X^* = \arg \min_X \|AX - T\|_F$ . One can create a coreset of  $\ell$  rows (or weighted linear combination of rows): the  $\ell \times d$  matrix  $\hat{A}$  and  $\ell \times w$  matrix  $\hat{T}$  imply a matrix  $\hat{X} = \arg \min_X \|\hat{A}X - \hat{T}\|_2$  that satisfies

$$(1 - \varepsilon)\|AX^* - T\|_F^2 \leq \|\hat{A}\hat{X} - \hat{T}\|_F^2 \leq (1 + \varepsilon)\|AX^* - T\|_F^2.$$

Using the random projection techniques described above, one can sketch  $\hat{A} = SA$  and  $\hat{T} = ST$  with  $\ell = O(d^2/\varepsilon^2)$  for hashing approaches or  $\ell = O(d/\varepsilon^2)$  for iid approaches. Moreover, Sarlos [Sar06] observed that for the  $w = 1$  case, since only a single direction (the optimal one) is required to be preserved (see *weak coresets* below), one can also use just  $\ell = O(d^2/\varepsilon)$  rows. Using row-sampling, one can deterministically select  $\ell = O(d/\varepsilon^2)$  rows [BDMI13]. The above works also provide bounds for approximating the multiple-regression spectral norm  $\|AX^* - T\|_2$ .

Mainly considering the single-regression problem when  $w = 1$  (in this case spectral and Frobenius norms bounds are equivalent  $p = 2$  norms), there also exists bounds for approximating  $\|AX - T\|_p$  for  $p \in [1, \infty)$  using random projection approaches and row sampling [CDMI<sup>+</sup>13, Woo14]. The main idea is to replace iid Gaussian random variables which are 2-stable with iid  $p$ -stable random variables. These results are improved using max-norm stability results [WZ13] embedding into  $\ell_\infty$ , or for other robust loss functions like the Huber loss [CW15b, CW15a].

---

---

## 49.5 CLUSTERING

An assignment-based clustering of a data set  $X \subset \mathbb{R}^d$  is defined by a set of  $k$  centers  $C \subset \mathbb{R}^d$  and a function  $\phi_C : \mathbb{R}^d \rightarrow C$ , so  $\phi_C(x) = \arg \min_{c \in C} \|x - c\|$ . The function  $\phi_C$  maps to the closest center in  $C$ , and it assigns each point  $x \in X$  to a center and an associated cluster. It will be useful to consider a weight  $w : X \rightarrow \mathbb{R}^+$ . Then a clustering is evaluated by a cost function

$$\text{cost}_p(X, w, C) = \sum_{x \in X} w(x) \cdot \|x - \phi_C(x)\|^p.$$

When the weighting function  $w(x)$  is equal for all points  $x \in X$  (say  $w(x) = 1/|X|$ , which we assume as default), then we may simply write  $\text{cost}_p(X, C)$ . We also define  $\text{cost}_\infty(X, C) = \max_{x \in X} \|x - \phi_C(x)\|$ . These techniques extend to when the centers of the clusters are not just points, but can also be higher-dimensional subspaces.

---

## GLOSSARY

***k*-means / *k*-median / *k*-center clustering problem:** Given a set  $X \subset \mathbb{R}^d$ , find a set  $C$  of size  $k$  that minimizes  $\text{cost}_2(X, C)$  (respectively,  $\text{cost}_1(X, C)$  and  $\text{cost}_\infty(X, C)$ ).

**$(k, \varepsilon)$ -coreset for *k*-means / *k*-median / *k*-center:** Given a point set  $X \subset \mathbb{R}^d$ , then a subset  $S \subset X$  is a  $(k, \varepsilon)$ -coreset for *k*-means (respectively, *k*-median and *k*-center) if for all center sets  $C$  of size  $k$  and parameter  $p = 2$  (respectively  $p = 1$  and  $p = \infty$ ) that

$$(1 - \varepsilon)\text{cost}_p(X, C) \leq \text{cost}_p(S, C) \leq (1 + \varepsilon)\text{cost}_p(X, C).$$

**projective distance:** Consider a set  $C = (C_1, C_2, \dots, C_k)$  of  $k$  affine subspaces of dimension  $j$  in  $\mathbb{R}^d$ , and a power  $p \in [1, \infty)$ . Then for a point  $x \in \mathbb{R}^d$  the projective distance is defined  $\text{dist}_p(C, x) = \min_{C_i \in C} \|x - \pi_{C_i}(x)\|^p$ , recalling that  $\pi_{C_i}(x) = \arg \min_{y \in C_i} \|x - y\|$ .

**projective  $(k, j, p)$ -clustering problem:** Given a set  $X \subset \mathbb{R}^d$ , find a set  $C$  of  $k$   $j$ -dimensional affine subspaces of size  $k$  that minimizes  $\text{cost}_p(X, C) = \sum_{x \in X} \text{dist}_p(C, x)$ .

**$(k, j, \varepsilon)$ -coreset for projective  $(k, j, p)$ -clustering:** Given a point set  $X \subset \mathbb{R}^d$ , then a subset  $S \subset X$ , a weight function  $w : S \rightarrow \mathbb{R}^+$ , and a constant  $\gamma$ , is

a  $(k, j, \varepsilon)$ -coreset for projective  $(k, j, p)$ -clustering if for all  $j$ -dimensional center sets  $C$  of size  $k$  that

$$(1 - \varepsilon)\text{cost}_p(X, C) \leq \text{cost}_p(X, w, C) + \gamma \leq (1 + \varepsilon)\text{cost}_p(X, C).$$

In many cases the constant  $\gamma$  may not be needed; it will be 0 unless stated.

**strong coreset:** Given a point set  $X \subset \mathbb{R}^d$ , it is a subset  $S \subset X$  that approximates the distance to any  $k$ -tuple of  $j$ -flats up to a multiplicative  $(1 + \varepsilon)$  factor.

**weak coreset:** Given a point set  $X \subset \mathbb{R}^d$ , it is a subset  $S \subset X$  such that the cost of the optimal solution (or one close to the optimal solution) of  $(k, j)$ -clustering on  $S$ , approximates the optimal solution on  $X$  up to a  $(1 + \varepsilon)$  factor. So a strong coreset is also a weak coreset, but not the other way around.

### $k$ -MEANS and $k$ -MEDIAN CLUSTERING CORESETS

$(k, \varepsilon)$ -Coresets for  $k$ -means and for  $k$ -median are closely related. The best bounds on the size of these coresets are independent of both  $n$  and  $d$ , the number and dimension of points in the original point set  $X$ . Feldman and Langberg [FL11] showed how to construct a strong  $(k, \varepsilon)$ -coreset for  $k$ -median and  $k$ -means clustering of size  $O(dk/\varepsilon^2)$ . They also show how to construct a weak  $(k, \varepsilon)$ -coreset [FMS07] of size  $O(k \log(1/\varepsilon)/\varepsilon^3)$ . These bounds can generalize for any  $\text{cost}_p$  for  $p \geq 1$ . However, note that for any fixed  $X$  and  $C$  that  $\text{cost}_p(X, C) > \text{cost}_{p'}(X, C)$  for  $p > p'$ , hence these bounds are not meaningful for the  $p = \infty$  special case associated with the  $k$ -center problem. Using the Merge-Reduce framework, the weak coreset constructions can be realized in a stream using  $O((k/\varepsilon^3) \log(1/\varepsilon) \log^4 n)$  space.

Interestingly, in contrast to earlier work on these problems [HPK07, HPM04, Che06] which applied various forms of geometric discretization of  $\mathbb{R}^d$ , the above results make an explicit connection with VC-dimension-type results and density approximation [LS10, VX12]. The idea is each point  $x \in X$  is associated with a function  $f_x(\cdot) = \text{cost}_p(x, \cdot)$ , and the total cost  $\text{cost}_p(X, \cdot)$  is a sum of these. Then the mapping of these functions onto  $k$  centers corresponds to a generalized dimension, similar to the VC-dimension of a dual range space, with dimension  $O(kd)$ , and then standard sampling arguments can be applied.

### $k$ -CENTER CLUSTERING CORESETS

The  $k$ -center clustering problem is harder than the  $k$ -means and  $k$ -median ones. It is NP-hard to find a set of centers  $\tilde{C}$  such that  $\text{cost}_\infty(X, \tilde{C}) \leq (2 - \eta)\text{cost}_\infty(X, C^*)$  where  $C^*$  is the optimal center set and for any  $\eta > 0$  [Hoc97]. Yet, famously the Gonzalez algorithm [Gon85], which always greedily chooses the point furthest away from any of the points already chosen, finds a set  $\hat{C}$  of size  $k$ , such  $\text{cost}_\infty(X, \hat{C}) \leq 2 \cdot \text{cost}_\infty(X, C^*)$ . This set  $\hat{C}$ , plus the furthest point from any of these points (i.e., run the algorithm for  $k + 1$  steps instead of  $k$ ) is a  $(k, 1)$  coreset (yielding the above stated 2 approximation) of size  $k + 1$ . In a streaming setting, McCutchen and Khuller [MK08] describe a  $O(k \log k \cdot (1/\varepsilon) \log(1/\varepsilon))$  space algorithm that provides  $(2 + \varepsilon)$  approximation for the  $k$ -center clustering problem, and although not stated, can probably be interpreted as a streaming  $(k, 1 + \varepsilon)$ -coreset for  $k$ -center clustering.

To get a  $(k, \varepsilon)$ -coreset, in low dimensions, one can use the result of the Gonzalez algorithm to define a grid of size  $O(k/\varepsilon^d)$ , keeping one point from each grid cell as

a coresets of the same size [AP02], in time  $O(n + k/\varepsilon^d)$  [HP04a]. In high dimensions one can run  $O(k^{O(k/\varepsilon)})$  sequences of  $k$  parallel MEB algorithms to find a  $k$ -center coresets of size  $O(k/\varepsilon)$  in  $O(dnk^{O(k/\varepsilon)})$  time [BC03a].

---

## PROJECTIVE CLUSTERING CORESETS

Projective clustering seeks to find a set of  $k$  subspaces of dimension  $j$  which approximate a large, high-dimensional data set. This can be seen as the combination of the subspace (matrix) approximations and clustering coresets.

Perhaps surprisingly, not all shape-fitting problems admit coresets – and in particular subspace clustering ones pose a problem. Har-Peled showed [HP04b] that no coresets exist for the 2-line-center clustering problem of size sublinear in the dataset. This result can be interpreted so that for  $j = 1$  (and extended to  $j = 2$ ),  $k = 2$ , and  $d = 3$  then there is no coresets for projective  $(k, j)$ -center clustering problem sublinear in  $n$ . Moreover a result of Meggido and Tamir [MT83] can be interpreted to say for  $j \geq 2$  and  $k > \log n$ , the solution cannot be approximated in polynomial time, for any approximation factor, unless  $P = NP$ .

This motivates the study of bicriteria approximations, where the solution to the projective  $(j, k, p)$ -clustering problem can be approximated using a solution for larger values of  $j$  and/or  $k$ . Feldman and Langberg [FL11] describe a strong coresets for projective  $(j, k)$ -clustering of size  $O(djk/\varepsilon^2)$  or weak coresets of size  $O(kj^2 \log(1/\varepsilon)/\varepsilon^3)$ , which approximated  $k$  subspaces of dimension  $j$  using  $O(k \log n)$  subspaces of dimensions  $j$ . This technique can yield stronger bounds in the  $j = 0$  and  $p = \infty$  case (the  $k$ -center clustering problem) where a set of  $O(k \log n)$  cluster centers can be shown to achieve error no more than the optimal set of  $k$  centers: a  $(k, 0)$ -coresets for  $k$ -center clustering with an extra  $O(\log n)$  factor in the number of centers. Other tradeoffs are also described in their paper where the size or approximation factor varies as the required number of subspaces changes. These approaches can be made to work in a stream with an extra factor  $\log^4 n$  in space.

Feldman, Schmidt, and Sohler [FSS13] consider the specific case of  $\text{cost}_2$  and crucially make use of a non-zero  $\gamma$  value in the definition of a  $(k, j, \varepsilon)$ -coresets for projective  $(k, j, 2)$ -clustering. They show strong coresets of size  $O(j/\varepsilon)$  for  $k = 1$  (subspace approximation), of size  $O(k^2/\varepsilon^4)$  for  $j = 0$  ( $k$ -means clustering), of size  $\text{poly}(2^k, \log n, 1/\varepsilon)$  if  $j = 1$  ( $k$ -lines clustering), and of size  $\text{poly}(2^{kj}, 1/\varepsilon)$  if  $j, k > 1$  and under the assumption that the coordinates of all points are integers between 1 and  $n^{O(1)}$ . The runtime of these algorithms is dominated by computing the principle component analysis (e.g., via the singular value decomposition) which requires  $O(nd \min\{n, d\})$  time; this can likely be improved using approximate variants (c.f. [CEM<sup>+</sup>15]). These constructions also extend to streaming algorithms with extra  $\log n$  factors in space.

---

---

## 49.6 SOURCES AND RELATED MATERIAL

---

### SURVEYS

[AHPV07]: A slightly dated, but excellent survey on coresets in geometry.

[HP11]: Book on geometric approximation that covers many of the above topics, for instance chapters 5 ( $\varepsilon$ -approximations and  $\varepsilon$ -nets), Chapter 19 (dimensionality reduction), and Chapter 23 ( $\varepsilon$ -kernel coresets).

[Mut05]: On Streaming, including history, puzzles, applications, and sketching.

[Cor11]: Nice introduction to sketching and its variations.

[GI10]: Survey on  $k$ -sparse  $\varepsilon$ - $(\ell_p/\ell_q)$  approximations.

[Mah11, Woo14]: Surveys of randomized algorithms for Matrix Sketching.

---

## RELATED CHAPTERS

Chapter 9: Low-distortion embeddings of finite metric spaces

Chapter 14: Geometric discrepancy theory and uniform distribution

Chapter 48: Epsilon-Nets and epsilon-approximations

---

## REFERENCES

- [Ach03] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66:671–687, 2003.
- [ACH<sup>+</sup>13] Pankaj Agarwal, Graham Cormode, Zengfeng Huang, Jeff Phillips, Zhewei Wei, and Ke Yi. Mergeable summaries. *ACM Transactions on Database Systems*, 38(26), 2013.
- [AES10] Boris Aronov, Esther Ezra, and Micha Sharir. Small-size epsilon-nets for axis-parallel rectangles and boxes. *SIAM Journal of Computing*, 39:3248–3282, 2010.
- [AGM12] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: Sparisfication, spanners, and subgraphs. In *Proceedings ACM Symposium on Principles of Database Systems*, 2012.
- [AHPV04] Pankaj K. Agarwal, Sarel Har-Peled, and Kasturi Varadarajan. Approximating extent measure of points. *Journal of ACM*, 51(4):606–635, 2004.
- [AHPV07] Pankaj K. Agarwal, Sarel Har-Peled, and Kasturi Varadarajan. Geometric approximations via coresets. *Current Trends in Combinatorial and Computational Geometry (E. Welzl, ed.)*, 2007.
- [AHPY08] Pankaj K. Agarwal, Sarel Har-Peled, and Hai Yu. Robust shape fitting via peeling and grating coresets. *Discrete & Computational Geometry*, 39:38–58, 2008.
- [AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.
- [AP02] Pankaj K. Agarwal and Cecilia M. Procopiuc. Exact and approximate algorithms for clustering. *Algorithmica*, 33:201–226, 2002.
- [AS15] Pankaj K. Agarwal and R. Sharathkumar. Streaming algorithms for extent problems in high dimensions. *Algorithmica*, 72:83–98, 2015.
- [Ban10] Nikhil Bansal. Constructive algorithms for dsicrepancy minimization. In *Proceedings of IEEE Symposium on Foundations of Computer Science*. (arXiv:1002.2259), 2010.
- [BC03a] Mihai Bădoiu and Kenneth L. Clarkson. Optimal coresets for balls. Technical report, <http://kenclarkson.org/coresets1.pdf>, 2003.

- [BC03b] Mihai Bădoiu and Kenneth L. Clarkson. Smaller core-sets for balls. In *Proceedings 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 801–802, 2003.
- [BCEG04] A. Bagchi, A. Chaudhary, David Eppstein, and Michael Goodrich. Deterministic sampling and range counting on geometric data streams. In *Proceedings 20th Annual ACM Symposium on Computational Geometry*, pages 144–151, 2004.
- [BCIS10] R. Berinde, G. Cormode, P. Indyk, and M. Strauss. Space-optimal heavy hitters with strong error bounds. *ACM Transactions on Database Systems*, 35(4), 2010.
- [BDMI13] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal coresets for least-squares regression. *SIAM Journal on Computing*, 43:687–717, 2013.
- [BDMI14] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near optimal column-based matrix reconstruction. *SIAM Journal of Computing*, 43:687–717, 2014.
- [BHPR16] Avrim Blum, Sariel Har-Peled, and Ben Raichel. Sparse approximation via generating point sets. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*, 2016.
- [BS80] Jon Louis Bentley and James B. Saxe. Decomposable searching problems I: Static-to-dynamic transformations. *Journal of Algorithms*, 1(4), 1980.
- [BS13] Nikhil Bansal and Joel Spencer. Deterministic discrepancy minimization. *Algorithmica*, 67:451–471, 2013.
- [BW14] Christos Boutsidis and David P. Woodruff. Optimal CUR decompositions. In *Proceedings ACM Symposium on Theory of Computing*, 2014.
- [CCFC02] Moses Charikar, K. Chen, and Michael Farach-Colton. Finding frequent items in data streams. In *Proceedings International Colloquium on Automata, Languages and Programming*, 2002.
- [CDMI<sup>+</sup>13] Kenneth L. Clarkson, Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, Xiangrui Meng, and David P. Woodruff. The fast cauchy transform and faster robust linear regression. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*. (extended version | arXiv:1207.4684), 2013.
- [CEM<sup>+</sup>15] Micheal B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for  $k$ -means clustering and low rank approximation. In *Proceedings of ACM Symposium on Theory of Computing*, 2015.
- [Cha00] Bernard Chazelle. *The Discrepancy Method*. Cambridge, 2000.
- [Cha06] Timothy Chan. Faster core-set constructions and data-stream algorithms in fixed dimensions. *Computational Geometry: Theory and Applications*, 35:20–35, 2006.
- [Che06] Ke Chen. On  $k$ -median clustering in high dimensions. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*, 2006.
- [CHW10] Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. In *Proceedings 51st IEEE Symposium on Foundations of Computer Science*, 2010.
- [Cla10] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms*, 6(4), 2010.
- [CLM<sup>+</sup>15] Micheal B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings Innovations in Theoretical Computer Science*, 2015.
- [CM96] Bernard Chazelle and Jiri Matousek. On linear-time deterministic algorithms for optimization problems in fixed dimensions. *Journal of Algorithms*, 21:579–597, 1996.



- [CM05] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 55:58–75, 2005.
- [Cor11] Graham Cormode. Synposes for massive data: Samples, histograms, wavelets and sketches. In *Foundations and Trends in Databases*. Now Publishers, 2011.
- [CV07] Kenneth L. Clarkson and Kasturi Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete & Computational Geometry*, 37:43–58, 2007.
- [CW13] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings 45th ACM Symposium on Theory of Computing*, 2013.
- [CW15a] Kenneth L. Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation. In *Proceedings 47th IEEE Symposium on Foundations of Computer Science*, 2015.
- [CW15b] Kenneth L. Clarkson and David P. Woodruff. Sketching for M-estimators: A unified approach to robust regression. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, 2015.
- [DF03] M. Durand and Philippe Flajolet. Loglog counting of large cardinalities. In *Proceedings of European Symposium on Algorithms*, 2003.
- [DFK<sup>+</sup>04] Petros Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56:9–33, 2004.
- [DG03] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22:60–65, 2003.
- [DKM06] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36:158–183, 2006.
- [DLOM02] Erik Demaine, Alejandro Lopez-Ortiz, and J. Ian Munro. Frequency estimation of internet packet streams with limited space. In *Proceedings European Symposium on Algorithms*, 2002.
- [DMIMW12] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13:3441–3472, 2012.
- [DMM08] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal of Matrix Analysis and Applications*, 30:844–881, 2008.
- [DV06] Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *RANDOM*, 2006.
- [FFK11] Dan Feldman, Matthew Faulkner, and Adreas Krause. Scalable training of mixture models via coresets. In *Proceedings 25th Conference on Neural Information Processing Systems*, 2011.
- [FKV04] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of ACM*, 51:1025–1041, 2004.
- [FL11] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings ACM Symposium on Theory of Computing*, 2011.
- [FM85] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

- [FMS07] Dan Feldman, Morteza Monemizedah, and Christian Sohler. A PTAS for  $k$ -means clustering based on weak coresets. In *Symp. on Computational Geometry*, 2007.
- [FO15] David Felber and Rafail Ostrovsky. A randomized online quantile summary in  $O(1/\epsilon \cdot \log(1/\epsilon))$  words. In *RANDOM*, 2015.
- [FS12] Dan Feldman and Lenard J. Schulman. Data reduction for weighted and outlier-resistant clustering. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2012.
- [FSS13] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for  $k$ -means, PCA, and projective clustering. In *Proceedings 24th ACM-SIAM Symposium on Discrete Algorithms*, 2013.
- [FT15] Dan Feldman and Tamir Tassa. More constraints, smaller coresets: Constrained matrix approximation of sparse big data. In *Proceedings ACM Symposium on Knowledge Discovery and Data Mining*, 2015.
- [FVR15] Dan Feldman, Mikhail Volkov, and Daniela Rus. Dimensionality reduction of massive sparse datasets using coresets. arxiv:1503.01663, 2015.
- [FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.
- [GI10] Anna C. Gilbert and Piotr Indyk. Sparse recovery using sparse matrices. In *Proceedings of IEEE*, 2010.
- [GJ09] Bernd Gärtner and Martin Jaggi. Coresets for polytope distance. In *Proceedings 25th Annual Symposium on Computational Geometry*, 2009.
- [GK01] Michael Greenwald and Sanjeev Khanna. Space-efficient online computation of quantile summaries. In *Proc. ACM Symposium on Management of Data*, 2001.
- [GLPW15] Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *arXiv:1501.01711*, 2015.
- [Gon85] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.
- [GP14] Mina Ghashami and Jeff M Phillips. Relative errors for deterministic low-rank matrix approximations. In *Proceedings of 25th ACM-SIAM Symposium on Discrete Algorithms*, 2014.
- [Han15] Steve Hanneke. The optimal sample complexity of PAC learning. Technical report, arXiv:1507.00473, 2015.
- [Hoc97] Dorit Hochbaum. *Approximation Algorithms for NP-Hard problems*. PWS Publishing Company, 1997.
- [HP04a] Sarel Har-Peled. Clustering motion. *Discrete & Computational Geometry*, 31:545–565, 2004.
- [HP04b] Sarel Har-Peled. No coreset, no cry. In *Proceedings 24th Conference on Foundations of Software Technology and Theoretical Computer Science*, 2004.
- [HP06] Sarel Har-Peled. Coresets for discrete integration and clustering. In *Proceedings Foundations of Software Technology and Theoretical Computer Science*, 2006.
- [HP11] Sarel Har-Peled. *Geometric Approximation Algorithms*. AMS, 2011.
- [HPK07] Sarel Har-Peled and Akash Kushal. Smaller coresets for  $k$ -median and  $k$ -mean clustering. *Discrete & Computational Geometry*, 37:3–19, 2007.
- [HPM04] Sarel Har-Peled and Soham Mazumdar. Coresets for  $k$ -means and  $k$ -median clustering and their applications. In *Proc. ACM Symp. on Theory of Computing*, 2004.

- [HPS11] Sarel Har-Peled and Micha Sharir. Relative (p,eps)-approximations in geometry. *Discrete & Computational Geometry*, 45(3):462–496, 2011.
- [HPV04] Sarel Har-Peled and Kasturi Varadarajan. High-dimensional shape fitting in linear time. *Discrete & Computational Geometry*, 32:269–288, 2004.
- [HPW04] Sarel Har-Peled and Yusu Wang. Shape fitting with outliers. *SIAM Journal of Computing*, 33:269–285, 2004.
- [HW87] David Haussler and Emo Welzl. Epsilon nets and simplex range queries. *Discrete & Computational Geometry*, 2:127–151, 1987.
- [IKP14] Piotr Indyk, Michael Kapralov, and Eric Price. (Nearly) Space-optimal sparse Fourier transform. In *Proceedings ACM Symposium on Discrete Algorithms*, 2014.
- [JKP11] Sarang Joshi, Raj Varma Kommaraju, and Jeff M. Phillips. Comparing distributions and shapes using the kernel distance. In *Proceedings 27th Symposium on Computational Geometry*, 2011.
- [JL84] William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26:189–206, 1984.
- [KSP03] Richard M. Karp, Scott Shenker, and Christos H. Papadimitriou. A simple algorithm for finding frequent elements in streams and bags. *ACM Transactions on Database Systems*, 28:51–55, 2003.
- [Lar14] Kasper Green Larsen. On range searching in the group model and combinatorial discrepancy. *SIAM Journal on Computing*, 43:673–686, 2014.
- [Lib13] Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.
- [LLS01] Yi Li, Phil Long, and Aravind Srinivasan. Improved bounds on the samples complexity of learning. *Journal of Computer and System Sciences*, 62:516–527, 2001.
- [LPW14] Jian Li, Jeff M. Phillips, and Haitao Wang. epsilon-Kernel coresets for stochastic points. Technical report, arXiv:1411.0194, 2014.
- [LS10] Micheal Langberg and Leonard J. Schulman. Universal  $\varepsilon$ -approximators for integrals. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*, 2010.
- [MAA06] A. Metwally, D. Agrawal, and A.E. Abbadi. An integrated efficient solution for computing frequent and top-k elements in data streams. *ACM Transactions on Database Systems*, 31(3):1095–1133, 2006.
- [Mah11] Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning, NOW Publishers*, 3(2), 2011.
- [Mat08] Jiri Matousek. On variants of the Johnson-Lindenstrauss lemma. *Random Structures & Algorithms*, 33:142–156, 2008.
- [Mat10] Jiri Matousek. *Geometric Discrepancy; An Illustrated Guide, 2nd printing*. Springer-Verlag, 2010.
- [MG82] J. Misra and D. Gries. Finding repeated elements. *Sc. Comp. Prog.*, 2:143–152, 1982.
- [MK08] Rochard Matthew McCutchen and Samir Khuller. Streaming algorithms for  $k$ -center clustering with outliers and with anonymity. In *Workshop on Approximation Algorithms*, 2008.
- [MNT15] Jiri Matousek, Aleksandar Nikolov, and Kunal Talwar. Combinatorial discrepancy for boxes via the ellipsoid-infinity norm. In *Proceedings International Symposium on Computational Geometry*. (full version: 1408.1376), 2015.

- [MSW90] Jiri Matousek, Raimund Seidel, and Emo Welzl. How to net a lot with a little: Small  $\varepsilon$ -nets for disks and halfspaces. In *Proceedings ACM Symposium on Computational Geometry*. (corrected version: <http://kam.mff.cuni.cz/~matousek/enets3.ps.gz>), 1990.
- [MT83] Nimrod Meggido and Arie Tamir. Finding least-distance lines. *SIAM Journal on Algebraic and Discrete Methods*, 4:207–211, 1983.
- [Mut05] S. Muthukrishnan. Data streams: Algorithms and applications. In *Now Publishers*, 2005.
- [NN13] Jelani Nelson and Huy L. Nguyen. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proceedings of 54th IEEE Symposium on Foundations of Computer Science*, 2013.
- [Phi13] Jeff M. Phillips.  $\varepsilon$ -Samples for kernels. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*, 2013.
- [PR08] Evangelia Pyrga and Saurabh Ray. New existence proofs for  $\varepsilon$ -nets. In *Proceedings Symposium on Computational Geometry*, 2008.
- [PT13] Janos Pach and Gábor Tardos. Tight lower bounds for the size of epsilon-nets. *Journal of the American Mathematical Society*, 26:645–658, 2013.
- [PZ15] Jeff M. Phillips and Yan Zheng. Subsampling in smooth range spaces. In *Proceedings Algorithmic Learning Theory*, 2015.
- [Sar06] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.
- [STZ04] Subhash Suri, Csaba D. Tóth, and Yunhong Zhou. Range counting over multidimensional data streams. In *Proc. Symposium on Computational Geometry*, 2004.
- [TKC05] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.
- [VC71] Vladimir Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.
- [VX12] Kasturi Varadarajan and Xin Xiao. On the sensitivity of shape fitting problems. In *Proceedings International Conference on Foundations of Software Technology and Theoretical Computer Science*. arxiv:1209.4893, 2012.
- [Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10:1–157, 2014.
- [WZ13] David P. Woodruff and Qin Zhang. Subspace embeddings and lp regression using exponential random variables. In *Proceedings Conference on Learning Theory*, 2013.
- [YA07] Hai Yu and Pankaj K. Agarwal. A space-optimal data-stream algorithm for coresets in the plane. In *Proceedings 23rd Symposium on Computational Geometry*, 2007.
- [YAPV08] Hai Yu, Pankaj K. Agarwal, Roditty Poreddy, and Kasturi Varadarajan. Practical methods for shape fitting and kinetic data structures using coresets. *Algorithmica*, 52:378–402, 2008.
- [ZZ08] Hamid Zarrabi-Zadeh. An almost space-optimal streaming algorithm for coresets in fixed dimensions. In *Proceedings 16th European Symposium on Algorithms*, 2008.