Oblivious Sketching for Logistic Regression

Alexander Munteanu [1] **Simon Omlor** [2] **David P. Woodruff** [3]

Abstract

What guarantees are possible for solving logistic regression in one pass over a data stream? To answer this question, we present the first data oblivious sketch for logistic regression. Our sketch can be computed in input sparsity time over a turnstile data stream and reduces the size of a d-dimensional data set from n to only $\mathrm{poly}(\mu d \log n)$ weighted points, where μ is a useful parameter which captures the complexity of compressing the data. Solving (weighted) logistic regression on the sketch gives an $O(\log n)$-approximation to the original problem on the full data set. We also show how to obtain an $O(1)$-approximation with slight modifications. Our sketches are fast, simple, easy to implement, and our experiments demonstrate their practicality.

1. Introduction

Sketches and coresets are arguably the most promising and widely used methods to facilitate the analysis of massive data with provable accuracy guarantees (Phillips, 2017; Munteanu & Schwiegelshohn, 2018; Feldman, 2020). Sketching has become a standard tool in core research areas such as data streams (Muthukrishnan, 2005) and numerical linear algebra (Mahoney, 2011; Woodruff, 2014), and constantly paves its way into diverse areas including computational geometry (Braverman et al., 2019; Meintrup et al., 2019), computational statistics (Geppert et al., 2017; Munteanu, 2019), machine learning (Nelson, 2020) and artificial intelligence (van den Brand et al., 2020; Gajjar & Musco, 2020; Molina et al., 2018). Following the sketch-and-solve paradigm, we first apply a simple and fast dimensionality reduction technique to compress the data to a significantly smaller *sketch* of polylogarithmic size. In a second step we feed the sketch to a standard solver for the problem, that needs little or no modifications. The theoretically challenging part is to prove an approximation guarantee for the solution obtained from the sketch with respect to the original large data set.

1.1. Related work

Deficiencies of coreset constructions. Most works on logistic regression have studied *coresets* as a data reduction method. Those are small subsets of the data, often obtained by subsampling from a properly designed importance sampling distribution (Huggins et al., 2016; Tolochinsky & Feldman, 2018; Munteanu et al., 2018; Tukan et al., 2020; Samadian et al., 2020). Those results often rely on regularization as a means to obtain small coresets. This changes the sampling distribution such that they do not generally apply to the unregularized setting that we study. The above coreset constructions usually require random access to the data and are thus not directly suitable for streaming computations. Even where row-order processing is permissible, at least two passes are required, one for calculating or approximating the probabilities and another for subsampling and collecting the data, since the importance sampling distributions usually depend on the data. A widely cited general scheme for making static (or multi-pass) constructions streamable in one pass is the merge & reduce framework (Bentley & Saxe, 1980). However, this comes at the cost of additional polylogarithmic overhead in the space requirements and also in the update time. The latter is a severe limitation when it comes to high velocity streams that occur for instance in large scale physical experiments such as the large hadron collider, where up to 100 GB/s need to be processed and data rates are anticipated to grow quickly to several TB/s in the near future (Rohr, 2018). While the amortized insertion time of merge & reduce is constant for some problems, in the worst case $\Theta(\log n)$ repeated coreset constructions are necessary for the standard construction to propagate through the tree structure; see e.g. (Feldman et al., 2020). This poses a prohibitive bottleneck in high velocity applications. Any data that passes and cannot be processed in real time will be lost forever.

Another limitation of coresets and the merge & reduce

Authors listed in alphabetical order. [1]Dortmund Data Science Center, Faculties of Statistics and Computer Science, TU Dortmund University, Dortmund, Germany [2]Faculty of Statistics, TU Dortmund University, Dortmund, Germany [3]Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. Correspondence to: Alexander Munteanu <alexander.munteanu@tu-dortmund.de>, Simon Omlor <simon.omlor@tu-dortmund.de>, David Woodruff <dwoodruf@cs.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

scheme is that they work only in insertion streams, where the data is presented row-by-row. However it is unclear how to construct coresets when the data comes in column-wise order, e.g., when we first obtain the incomes of all individuals, then receive their heights and weights, etc. A similar setting arises when the data is distributed *vertically* on numerous sites (Stolpe et al., 2013). Sensor networks are another example where each sensor is recording only a single or a small subset of features (columns), e.g., each at one of many different production stages in a factory. Also the usual form of storing data in a table either row- or column-wise is not appropriate or efficient for extremely massive databases. The data is rather stored as a sequence of $(key, value)$ pairs in an arbitrary order in big unstructured databases (Gessert et al., 2017; Siddiqa et al., 2017).

The only work that can be simulated in a turnstile stream to tackle the extreme settings described above, is arguably (Samadian et al., 2020) via uniform subsampling. Their coreset size is roughly $\Theta(d\sqrt{n})$ and works only when the problem is regularized very strongly such that the loss function is within constant factors to the regularizer, and thus widely independent of the input data. Consequently, the contribution of each point becomes roughly equal and thus makes uniform sampling work. However, those arguments do not work for unconstrained logistic regression, where each single point can dominate the cost and thus no sublinear compression below $\Omega(n)$ is possible in the worst case, as was shown in (Munteanu et al., 2018). To cope with this situation, the authors of (Munteanu et al., 2018) introduced a complexity parameter $\mu(A)$ that is related to the statistical modeling of logistic regression, and is a useful measure for capturing the complexity of compressing the dataset A for logistic regression. They developed a coreset construction of size $\tilde{O}(\mu d^{3/2}\sqrt{n})$. Although calculating their sampling distribution can be simulated in a row-order stream, the aforementioned limitation to two passes is an unsolved open problem. The coreset size was reduced to $\mathrm{poly}(\mu d \log n)$ but only at the cost of even more row-order passes to compute repeatedly a coreset from a coreset, $O(\log\log n)$ times.

On the importance of data oblivious sketching. Oblivious sketching methods are much better positioned for handling high velocity streams, as well as highly unstructured and arbitrarily distributed data. Linear sketches allow efficient applications in single pass sequential streaming and in distributed environments, see, e.g. (Clarkson & Woodruff, 2009; Woodruff & Zhang, 2013; Kannan et al., 2014). Linear sketches can be updated in the most flexible dynamic setting, which is commonly referred to as the *turnstile* model, see, e.g., (Muthukrishnan, 2005) for a survey. In this model we initialize a matrix A to the all-zero matrix. The stream consists of $(key, value)$ updates of the form (i, j, v), meaning that A_{ij} will be updated to $A_{ij} + v$. A single entry can be defined by a single update or by a subsequence of not necessarily consecutive updates. For instance, a sequence $\ldots, (i, j, 27), \ldots, (i, j, -5), \ldots$ will result in $A_{ij} = 22$. Deletions are possible in this setting by using negative updates matching previous insertions. At first glance this model might seem technical or unnatural but we stress that for dealing with the aforementioned unstructured data, the design of algorithms working in the turnstile model is of high importance. We will see how any update can be calculated in $O(1)$ basic operations so it becomes applicable in high velocity real-time applications. Additionally, due to linearity, oblivious sketching algorithms can be represented as linear maps, i.e., sketching matrices S. In particular they support several operations such as adding, subtracting, and scaling databases A_j efficiently in the sketch space, since $SA = S\sum_j \alpha_j A_j = \sum_j \alpha_j SA_j$. For instance, if A_{t_1} and A_{t_2} are balances of bank accounts at time steps $t_1 < t_2$, then $SB = SA_{t_2} - SA_{t_1}$ is a sketch of the changes in the period $t \in (t_1, t_2]$.

Data oblivious sketching for logistic regression. In this paper we deal with unconstrained logistic regression in one pass over a turnstile data stream. As most known turnstile data stream algorithms are linear sketches (and there is some evidence that linear sketches are optimal for such algorithms in certain conditions (Li et al., 2014; Ai et al., 2016)), it is natural for achieving our goals to look for a distribution over random matrices that can be used to sketch the data matrix such that the (optimal) cost of logistic regression is preserved up to constant factors. Due to the aforementioned impossibility result, the reduced sketching dimension will depend polynomially on the mildness parameter $\mu(A)$, and thus we need $\mu(A)$ to be small, which is common under usual modeling assumptions in statistics (Munteanu et al., 2018). In this setting, logistic regression becomes similar to an ℓ_1-norm regression problem for the subset of misclassified inputs, and a uniform sample suffices to approximate the contribution of the other points.

Known linear *subspace embedding* techniques for ℓ_1 based on Cauchy (1-stable) random variables (Sohler & Woodruff, 2011) or exponential random variables (Woodruff & Zhang, 2013) have a dilation of $O(d \log d)$ or higher polynomials thereof, and nearly tight lower bounds for this distortion exist (Wang & Woodruff, 2019). While a contraction factor of $(1 - \varepsilon)$ seems possible over an entire linear subspace, a constant dilation bound for an arbitrary but fixed vector (e.g., the optimal solution) are the best we can hope for (Indyk, 2006; Clarkson & Woodruff, 2015; Li et al., 2021). A general sketching technique was introduced by (Clarkson & Woodruff, 2015) that achieves such a *lopsided* result for all regression loss functions that grow at least linearly and at most quadratically (the quadratic upper bound condition is necessary for a sketch with a sketching dimension subpolynomial in n to exist (Braverman & Ostrovsky, 2010)) and have properties of norms such as symmetry, are non-

decreasing in the absolute value of their argument, and have $f(0) = 0$, which is true for a class of robust M-estimators, though not for all. For example, the Tukey regression loss has zero growth beyond some threshold. The above sketching technique has been generalized to cope with this problem (Clarkson et al., 2019). However the latter work still relies on a symmetric and non-decreasing loss function f with $f(0) = 0$.

For the logistic regression loss $\ell(v) = \ln(1 + \exp(v))$, we note that it does not satisfy the above norm-like conditions since $\ell(0) = \ln(2)$, $\ell(x) \neq \ell(-x)$, and while it is linearly increasing on the positive domain, it is decreasing exponentially to zero on the negative domain. Indeed, the class of monotonic functions has linear lower bounds for the size of any coreset and more generally for any sketch (Tolochinsky & Feldman, 2018; Munteanu et al., 2018), where the unboundedness of the ratio $\ell(x)/\ell(-x)$ plays a crucial role.

1.2. Our contributions

In this paper we develop the first oblivious sketching techniques for a generalized linear model, specifically for logistic regression. Our `LogReg`-sketch is algorithmically similar to the M-estimator sketches of (Clarkson & Woodruff, 2015; Clarkson et al., 2019). However, there are several necessary changes and the theoretical analyses need non-trivial adaptations to address the special necessities of the logistic loss function. The sketching approach is based on a combination of subsampling at different levels and hashing the coordinates assigned to the same level uniformly into a small number of buckets (Indyk & Woodruff, 2005; Verbin & Zhang, 2012). Collisions are handled by summing all entries that are mapped to the same bucket, which corresponds to a variant of the so-called `CountMin`-sketch (Cormode & Muthukrishnan, 2005), where the sketch S_h on each level is presented only a fraction of all coordinates.

More precisely, we define an integer branching parameter b and a parameter $h_{\max} = O(\log_b n)$, and each row of our data matrix gets assigned to level $h \leq h_{\max}$ with probability proportional to b^{-h}. The row is then assigned one of the N buckets on level h uniformly at random and added to that bucket. The new matrix that we obtain consists of $h_{\max}$ blocks, where each block consists of N rows. The weight of a row is proportional to b^h. The formal definition of the sketch is in Section 3. This scheme is complemented by a row-sampling matrix T which takes a small uniform sample of the data, which will be dealt with in Section 4.

$$S = \begin{bmatrix} S_0 \\ S_1 \\ \vdots \\ S_{h_{\max}} \\ T \end{bmatrix}$$

The intuition behind this approach is that coordinates are grouped according to *weight classes* of similar loss which can be handled separately in the analysis. Weight classes with a small number of members will be approximated well on sketching levels with a large number of elements since roughly all members need to be subsampled to obtain a good estimate. Weight classes with many members will be approximated well on levels with a smaller number of subsamples, because if too many members survive the subsampling there will also be too many collisions under the uniform hashing, which would either lead to a large overestimate when those add up, or, due to asymmetry, would cancel each other and lead to large underestimations. The asymmetry problem is also one of the main reasons why we need a new analysis relying on the `CountMin`-sketch as a replacement for the `Count`-sketch previously used in (Clarkson & Woodruff, 2015; Clarkson et al., 2019). The reason is that `Count`-sketch assigns random signs to the coordinates before summing them up in a bucket. The error could thus not be bounded if the sign of an element is changed since the ratio $\ell(x)/\ell(-x)$ is unbounded for unconstrained logistic regression. Finally, since there could be too many small contributions near zero and logistic regression, unlike a normed loss function, assigns a non-zero but constant loss to them, their contribution can become significant. This is taken care of by the small uniform sample of size $\tilde{O}(\mu d)$.

Our main result is the following theorem, where $\texttt{nnz}(A)$ denotes the number of non-zero entries in A or in a data stream it corresponds to the number of updates,

$$f_w(Ax) = \sum_{i \in [n]} w_i \cdot \ln\left(1 + \exp(a_i x)\right)$$

denotes the weighted logistic loss function, and $f(Ax)$ is the unweighted case where w is the all 1s vector. It also assumes that the data is μ-complex for a small value μ meaning that $\mu(A) \leq \mu$ as in (Munteanu et al., 2018), see Section 2 for a formal definition:

Theorem 1. *Let $A \in \mathbb{R}^{n \times d}$ be a μ-complex matrix for bounded $\mu < \infty$. Then there is a distribution over sketching matrices $S \in \mathbb{R}^{r \times n}$ with $r = \text{poly}\left(\mu d \log(n)\right)$, and a corresponding weight vector $w \in \mathbb{R}^r$, for which $B = SA$ can be computed in $O(\texttt{nnz}(A))$ time over a turnstile data stream and for which if x' is the minimizer to $\min_x f_w(Bx)$, then with constant probability it holds that*

$$f(Ax') \leq O(\log n) \min_{x \in \mathbb{R}^d} f(Ax).$$

Further, there is a convex function $f_{w,c}$ such that for the minimizer x'' to $\min_x f_{w,c}(Bx)$ it holds that

$$f(Ax'') \leq O(1) \min_{x \in \mathbb{R}^d} f(Ax)$$

with constant probability.

The first item is a sketch-and-solve result in the sense that first, the data is sketched and then the sketch is put into a standard solver for weighted logistic regression. The output is guaranteed to be an $O(\log n)$ approximation. The second item requires a stronger modification which can be handled easily for instance with a subgradient based solver for convex functions. The individual loss for $f_{w,c}$ remains the original logistic loss as in f_w for each point. However for a fixed x occurring in the optimization, the loss and gradient are evaluated only on the K largest entries on each level of the sketch Bx (except for the uniform sample T), for a suitable $K < N$. This preserves the convexity of the problem, and guarantees a constant approximation. The details are given in Sections B.3, and 5.

Overview of the analysis. The rest of the paper is dedicated to proving Theorem 1. We first show that the logistic loss function can be split into two parts $f(Ax) \approx G^+(Ax) + f((Ax)^-)$, which can be handled separately while losing only an approximation factor of two; see Section 2.

The first part is $G^+(y) := \sum_{y_i \geq 0} y_i$, the sum of all positive entries which can be approximated by the aforementioned collection of sketches S_h. Here we show that with high probability no solution becomes much cheaper with respect to the objective function and that with constant probability, the cost of some good solution does not become too much larger, which can be bounded by a factor of at most $O(\log n)$ or $O(1)$ depending on which of our two algorithms we use. We prove this in Section 3. First, we bound the contraction. To do so we define *weight-classes* of similar loss. For weight classes with a small number of members, a leverage-score argument yields that there cannot be too many influential entries. For larger weight classes, we show that there exists a small subset of influential entries that on some subsampling level do not collide with any other influential entry when hashing into buckets, and thus represent their weight class well. This concludes the handling of the so-called *heavy-hitters*, see Section B.1. Those arguments hold with very high probability, and so we can union bound over a net and relate the remaining points to their closest point in the net. This yields the contraction bound for the entire solution space, see Section B.2. Although the high-level outline is similar to (Clarkson & Woodruff, 2015), several non-trivial adaptations are necessary to deal with the assymmetric G^+ function that is not a norm and has zero growth on the negative domain. The $O(\log n)$ dilation bound follows by a calculation of the expected value on each level and summing over $h_{\max} = O(\log n)$ levels. The $O(1)$ bound requires the aforementioned clipping of small contributions on each level. Each weight class q makes a main contribution on some level $h(q)$. The argument is now that it can have a significant impact only on levels in a small region of size $k = O(1)$ around $h(q) \pm k$. Further, with high probability for $h > h(q) + k$ there will be no element of the same weight class, so that the contribution to the expectation is zero, and for $h < h(q) - k$ the contribution can be bounded by $O(h_{\max}^{-1})$, so that for all three cases the expected contribution is at most $O(1)$ after summing over all levels.

The second part is $f^- := f((Ax)^-)$ which maps any misclassified point to $\ell(0) = \ln(2)$ and the remaining points to the usual logistic loss of a point, i.e., $\ell(a_i x) = \log(1 + \exp(a_i x))$. Here we prove that for μ-complex data sets the worst case contribution of any point can be bounded by roughly μ/n and thus a uniform sample of $\tilde{O}(\mu)$ can be used to approximate f^- well. This will be done via the well-known sensitivity framework (Langberg & Schulman, 2010) in Section 4. We put everything together to prove Theorem 1 in Section 5. In Section 6 our experimental results demonstrate that our sketching techniques are useful and competitive to uniform sampling, SGD, and an adaptive coreset construction. We show in some settings the oblivious sketch performs almost the same or better, but is never much worse. We stress that neither SGD nor the coreset allow the desired turnstile streaming capabilities. We finally conclude in Section 7.
Omitted proofs and details can be found in the supplementary material.

2. Preliminaries

2.1. Notation

In logistic regression we are usually given a data matrix $X \in \mathbb{R}^{n \times d}$ and a label vector $L \in \{-1, 1\}^n$. For notational brevity and since a data point always appears together with its label, we technically work with a data matrix $A \in \mathbb{R}^{n \times d}$ where each row a_i for $i \in [n]$ is defined as $a_i := -l_i x_i$. We set $g(v) = \ln(1 + \exp(v))$ for $v \in \mathbb{R}$. Our goal is to find $x \in \mathbb{R}^d$ that minimizes the logistic loss given by

$$f(Ax) = \sum_{i \in [n]} g(a_i x) = \sum_{i \in [n]} \ln\left(1 + \exp(a_i x)\right).$$

We parameterize our analysis by

$$\mu_A = \sup_{x \in \mathbb{R}^d \setminus \{0\}} \frac{\|(Ax)^+\|_1}{\|(Ax)^-\|_1}$$

where for $y \in \mathbb{R}^n$, the vector y^+ (resp. y^-) denotes the vector with all negative (resp. positive) entries replaced by 0. This definition of μ has been introduced before in (Munteanu et al., 2018) and is a useful parameter to bound the amount of data reduction possible for logistic regression. In the remainder we assume that A is μ-complex, meaning that $\mu_A \leq \mu$ for some $1 \leq \mu \leq \infty$. For any vector y we further define $G^+(y) := \sum_{y_i \geq 0} y_i$ to be the sum of all positive entries. Also we define $\bar{G}(y) := \|y\|_1$. Note that

by definition of μ the supremum considers for each $x \in \mathbb{R}^d$, also $-x$. Therefore, it holds for all $x \in \mathbb{R}^d$ that

$$\mu^{-1}\|(Ax)^-\|_1 \leq \|(Ax)^+\|_1 \leq \mu\|(Ax)^-\|_1.$$

In particular, the property $G^+(Ax) = \|(Ax)^+\|_1 \geq \frac{\|(Ax)^-\|_1}{\mu}$ will often be used.

2.2. Initial approach

Our first idea is that we can split f into two parts which can be approximated independently.

Lemma 2.1. *For all $x \in \mathbb{R}^d$ it holds that*

$$f(Ax) \;\geq\; \frac{1}{2}\left(f\left((Ax)^-\right) + G^+(Ax)\right)$$

and

$$f(Ax) \;\leq\; f\left((Ax)^-\right) + G^+(Ax).$$

Proof. Let $v \in \mathbb{R}_{\geq 0}$. Then it holds that $g(v) = g(0) + \int_0^v g'(y)dy = \ln(2) + \int_0^v g'(y)dy$. Note that $g'(y) = \frac{\exp(y)}{1+\exp(y)}$, and thus for any $y \in \mathbb{R}$ we have $g'(y) \leq 1$ and for any $y \in [0,\infty)$ we have $g'(y) \geq \frac{\exp(y)}{2\exp(y)} = \frac{1}{2}$. We conclude that

$$g(v) = g(0) + \int_0^v g'(y)dy \geq g(0) + \int_0^v \frac{1}{2}dy = g(0) + \frac{1}{2}v$$

and

$$g(v) = g(0) + \int_0^v g'(y)dy \leq g(0) + \int_0^v 1dy = g(0) + v.$$

Recall that in $(Ax)^-$ each coordinate $a_i x > 0$ is replaced by zero. Thus, if $a_i x > 0$ then $g(a_i x) = g(0) + \int_0^{a_i x} g'(y)dy = g((Ax)_i^-) + \int_0^{a_i x} g'(y)dy$. Hence, it holds that

$$\begin{aligned} f(Ax) &= \sum_{a_i x<0} g(a_i x) \;+\; \sum_{a_i x\geq 0} g(a_i x) \\ &\leq \sum_{a_i x<0} g(a_i x) \;+\; \sum_{a_i x\geq 0} g(0) + a_i x \\ &= \sum_{a_i x<0} g(a_i x) \;+\; \sum_{a_i x\geq 0} g(0) \;+\; \sum_{a_i x\geq 0} a_i x \\ &= f((Ax)^-) + G^+(Ax) \end{aligned}$$

and similarly

$$\begin{aligned} f(Ax) &= f((Ax)^-) + \sum_{a_i x>0} \int_0^{a_i x} g'(y)dy \\ &\geq f((Ax)^-) + \frac{1}{2}G^+(Ax) \\ &\geq \frac{1}{2}\left(f\left((Ax)^-\right) + G^+(Ax)\right). \end{aligned}$$

□

Next we show that for $\min_{x\in\mathbb{R}} f(Ax)$, there is a non-trivial lower bound that will later be used to show that $f((Ax)^-)$ can be approximated well enough.

Lemma 2.2. *For all $x \in \mathbb{R}^d$ it holds that*

$$f(Ax) \geq \frac{n}{2\mu}\left(1 + \ln(\mu)\right) = \Omega\left(\frac{n}{\mu}(1 + \ln(\mu))\right).$$

Proof. For any $w \geq 1$ it holds that $\ln(w) = \int_1^w \frac{1}{y}dy$. Thus for $v \leq 0$ we have $g(v) = \ln(1 + e^v) = \int_1^{1+e^v} \frac{1}{y}dy \geq \frac{e^v}{2}$ since for $1 \leq y \leq e^v + 1 \leq 2$ we have $\frac{1}{y} \in [\frac{1}{2}, 1]$. Let $z = Ax$. Using this fact and Lemma 2.1 we get $f(z) \geq \frac{1}{2}(\sum_i \exp(\min\{z_i, 0\}) + G^+(z))$. Since $\exp(v)$ is convex, Jensen's inequality implies $\sum_i \exp(\min\{z_i, 0\}) = n\sum_i \frac{1}{n}\exp(\min\{z_i, 0\}) \geq n\exp(\frac{1}{n}\sum_i \min\{z_i, 0\})$. Using this argument we get for $y = \frac{\|z^-\|_1}{n}$ that $\sum_i \exp(\min\{z_i, 0\}) \geq n\exp(-y)$. Recall that $G^+(z) \geq \frac{yn}{\mu}$ holds by definition of μ.

Using Lemma 2.1 we conclude that $f(z) \geq \frac{1}{2}(n\exp(-y) + \frac{yn}{\mu})$. The function $(n\exp(-y) + \frac{yn}{\mu})$ is minimized over y if its first derivative is zero, i.e., if

$$n\exp(-y) = \frac{n}{\mu}$$

which is equivalent to $y = \ln(\mu)$. Hence $f(z) \geq \frac{1}{2}\left(\frac{n}{\mu} + \frac{n\ln(\mu)}{\mu}\right)$. □

3. Approximating $G^+(Ax)$

Here we focus on approximating $G^+(Ax) = \sum_{a_i x>0} a_i x$. We develop a sketching method similar to the approach of (Clarkson & Woodruff, 2015). This gives us a new matrix $A' = SA$, referred to as the sketch, for which we will show that with high probability it holds for all $x \in \mathbb{R}^d$, that $G^+(A'x) \geq (1-\varepsilon)G^+(Ax)$ for $\varepsilon > 0$, and we also have $\mathbb{E}(G^+(A'x)) \leq CG^+(Ax)$ for some constant $C > 1$. We show the following result:

Theorem 2. *Given $\varepsilon > 0$, and $\delta > 0$ we set $r = O\left(d^5\left(\frac{\mu}{\varepsilon}\right)^7\delta^{-2}\ln^4\left(\frac{n\mu}{\delta\varepsilon}\right)\right)$. Then there is a random matrix $S \in \mathbb{R}^{r\times n}$ such that for $A' = SA$ and a convex function G_c^+, it holds that $G_c^+(A'x) \geq (1-\varepsilon)G^+(Ax)$ and $\mathbb{E}(G_c^+(A'x)) \leq CG^+(Ax)$ for some constant $C > 1$ and for all $x \in \mathbb{R}^d$. The failure probability of this event is at most δ.*

3.1. The sketching algorithm

The idea is to hash the rows of A uniformly into buckets. The rows that are assigned to the same bucket are added to obtain a row of A' which can also be written as $A' = SA$ for a suitable matrix S. To avoid that for a given $z = Ax$,

there are too many entries of z that cancel with each other, we assign a level to each bucket. The level of a bucket determines how many coordinates are assigned to it (in expectation). Buckets with fewer coordinates are given a larger weight. In this way, large entries of Ax are preserved in buckets with many coordinates, up to a small error, while the contribution of many small entries of Ax is preserved by buckets with few coordinates, but high weights.

More precisely, the sketching algorithm is defined as follows: we define N to be the number of buckets in each level and $h_{\max}$ to be the number of levels. Let b be a branching parameter that determines how the (expected) number of coordinates changes between different levels.

Then each coordinate $p \in [n]$ is hashed uniformly at random to bucket $g_p \in [N]$ at level $h_p \in [h_{\max}]$, where we set $h_p = h$ with probability $\frac{1}{\beta b^h}$, for $0 \leq h \leq h_{\max} = \log_b(\frac{n}{m})$ and some $b > 2$ and $\beta = \frac{b-b^{-h_{\max}}}{b-1}$. The weight of z_p is given by $w_p = b^{h_p}\beta$. The sketching matrix is given by $S \in \mathbb{R}^{h_{\max}N \times n}$, where $(S)_{jp} = b^{h_p}\beta$ if $j = g_p + h_p N$ and $(S)_{jp} = 0$ otherwise.

Assume we are given some error parameter $\varepsilon' \in (0, \frac{1}{3})$ and set $\varepsilon = \frac{\varepsilon'}{\mu'}$, where $\mu' = \mu + 1$. Then we have $G^+(z) \geq \frac{G(z^-)}{\mu} = \frac{G(z)-G(z^+)}{\mu}$, which is equivalent to $\frac{(\mu+1)G^+(z)}{\mu} = G^+(z) + \frac{G^+(z)}{\mu} \geq \frac{G(z)-G(z^+)}{\mu}$. Multiplying by μ gives us $G^+(z) \geq \frac{G(z)}{\mu+1} = \frac{G(z)}{\mu'}$. Let $\delta < 1$ be a failure probability. Let m be a parameter which determines whether a set of coordinates is considered large.

3.2. Outline of the analysis

Instead of explaining how the sketch is applied to A, we will explain how the sketch is applied to $z := Ax$ for a fixed x. Note that $(SA)x = S(Ax)$. We assume without loss of generality that $G(z) = \|z\|_1 = 1$. This can be done since for any $\lambda > 0$, we have $G(S\lambda z) = \lambda G(Sz)$ and $G^+(S\lambda z) = \lambda G^+(Sz)$.

We split the entries of z into weight classes and derive bounds for the contribution of each individual weight class. The goal is to show that for each $x \in \mathbb{R}^d$ the entries of z that can be large are the same, and for the remaining entries we can find a set of representatives which are in buckets of appropriate weights and are large in contrast to the remaining entries in their buckets. Therefore we let $Z = \{z_p \mid p \in [n]\}$ be the multiset of values appearing in z. We define weight classes as follows:
For $q \in \mathbb{N}$ we set $W_q^+ = \{z_p \in Z \mid 2^{-q} < z_p \leq 2^{-q+1}\}$ to be the positive weight class of q. Similarly we define $W_q = \{z_p \in Z \mid 2^{-q} < |z_p| \leq 2^{-q+1}\}$ to be the weight class of q. Since we are also interested in the number of elements in each weight class we define $h(q) := \lfloor \log_b(\frac{|W_q^+|}{\beta m}) \rfloor$ if $|W_q^+| \geq \beta m$ and $h(q) = 0$ otherwise. This way we have $\beta m b^{h(q)} \leq |W_q^+| \leq \beta m b^{h(q)+1}$ and thus $h(q)$ is the largest index such that the expected number of entries from W_q at level h is at least βm. Note that the contribution of weight class W_q is at most $2^{-q+1}n$. Thus we set $q_{\max} = \log_2(\frac{n}{\varepsilon})$ and will ignore weight classes with $q > q_{\max}$ as their contribution is smaller than ε.

Our first goal will be to show that there exists an event $\mathcal{E}$ with low failure probability (which will be defined later) such that if $\mathcal{E}$ holds then $G^+(SAx)$ gives us a good approximation to $G^+(Ax)$ with very high probability. More precisely:

Theorem 3. *If $\mathcal{E}$ holds then we have $G^+(SAx) \geq (1 - 60\varepsilon')G^+(Ax)$ for any fixed $x \in \mathbb{R}^d$ with failure probability at most $e^{-m\varepsilon^2/2}$.*

This will suffice to proceed with a net argument, i.e., we show that there exists a finite set $N \subset \mathbb{R}^d$ such that if we have $G^+(SAx) \geq (1 - \varepsilon')G^+(Ax)$ for all $x \in N$ then it holds that $G^+(SAx) \geq (1 - 4\varepsilon')G^+(Ax)$ for all $x \in \mathbb{R}^d$, and thus we obtain the desired contraction bound.

Theorem 4. *We have $G^+(SAx) \geq (1 - 240\varepsilon')G^+(Ax)$ for every $x \in \mathbb{R}^d$ with failure probability at most 2δ.*

Finally we show that in expectation $G^+(SAx)$ is upper bounded by $h_{\max}G^+(Ax) = O(\log(n))$. Further we show that there is a convex function G_c^+ and an event $\mathcal{E}'$ with low failure probability such that we have $G_c^+(SAx) \geq (1 - \varepsilon')G^+(Ax)$ with high probability, and in expectation $G_c^+(SAx)$ is upper bounded by $CG^+(Ax)$ for constant C.

Theorem 5. *There is a constant $C > 1$ such that if $\mathcal{E}'$ holds then $\mathbb{E}(G_c^+(Sz)) \leq CG^+(z)$.*

4. Approximating $f((Ax)^-)$

The following theorem shows that a uniform sample $R \subset \{a_i \mid i \in [n]\}$ gives us a good approximation to $f((Ax)^-)$. We also show that in expectation the contribution of R is not too large and thus with constant probability contributes at most $Cf(Ax)$ for some constant factor C, by Markov's inequality. Using the sensitivity framework and bounding its relevant parameters (see Section C) we get:

Theorem 6. *For a uniform sample $R \subset \{a_i \mid i \in [n]\}$ of size $k = O(\frac{\mu}{\varepsilon^2}(d\ln(\mu) + \ln(\frac{1}{\delta_1}))$ we have that with failure probability at most δ_1 that*

$$\sum_{a_i \in R} \frac{n}{k} g(a_i x) \geq f((Ax)^-) - 3\varepsilon f(Ax)$$

for all $x \in \mathbb{R}^d$. Further $\mathbb{E}(\sum_{a_i \in R} \frac{n}{k} g(a_i x)) = f(Ax)$.

5. Approximation for logistic regression in one pass over a stream of data

We set $\varepsilon = \frac{1}{8}$. To prove Theorem 1 we first show that we can tweak the sketch A' from Theorem 2 by adding weights and scaling A'. This way we get a new sketch where the weighted logistic loss and the sum of positive entries are the same up to an additive constant of $\ln(2)$. More precisely we set $A'' = Nh_{\max}A' = Nh_{\max}SA$ with equal weight $\frac{1}{Nh_{\max}}$ for all rows. We denote the weight vector by w'. This way we make sure that $f_{w'}(A''x)$ is very close to $G^+(Ax)$:

Lemma 5.1. *For all $x \in \mathbb{R}^d$ it holds that $G^+(A'x) \leq \frac{1}{Nh_{\max}} f(A''x) \leq G^+(A'x) + \ln(2)$.*

Proof. First note that for any $v \geq 0$ we have $g(v) = \ln(1 + \exp(v)) \geq \ln(\exp(v)) = v$. This implies that

$$\begin{aligned} f(A''x) &= \sum_{i=1}^{Nh_{\max}} g(a_i''x) \geq \sum_{a_i''x \geq 0} g(a_i''x) \\ &\geq \sum_{a_i''x \geq 0} a_i''x = \sum_{a_i'x \geq 0} Nh_{\max}a_i'x \\ &= Nh_{\max}G^+(A'x) \end{aligned}$$

Further, note that for any $v \in \mathbb{R}$ we have $g(v) \leq \ln(2) + \max\{v, 0\}$. This is true for $v \leq 0$ since g is monotonically increasing and $g(0) = \ln(2)$, and since the derivative of g is always bounded by 1 it also holds for $v > 1$, cf. Lemma 2.1. Consequently it holds that

$$\begin{aligned} f(A''x) &= \sum_{i=1}^{Nh_{\max}} g(a_i''x) \\ &\leq \sum_{i=1}^{Nh_{\max}} \ln(2) + \max\{0, a_i''x\} \\ &= \sum_{i=1}^{Nh_{\max}} \ln(2) \; + \sum_{a_i''x \geq 0} a_i''x \\ &= Nh_{\max}\ln(2) + \sum_{a_i'x \geq 0} Nh_{\max}a_i'x \\ &= Nh_{\max}(\ln(2) + G^+(Ax)). \end{aligned}$$

□

We are now ready to show our main result:

Proof of Theorem 1. Let (T, u) be a weighted uniform random sample from Theorem 6. We define $B = \begin{pmatrix} A'' \\ T \end{pmatrix}$ with weight vector $w = (w', u)$. The size of B is bounded by $\mathrm{poly}(\mu d \log n)$ since it is dominated by the sketch of Theorem 2. Note that B can handle any update in $O(1)$ time since we need to draw one random number for determining the level that a data point is assigned, another one for determining its bucket and a third one to decide whether to include it into the random sample or not. This sums up to $O(\mathtt{nnz}(A))$ time in total. We note that B can be computed over a turnstile data stream when we replace the random number generators by hash maps, so to compute the pseudorandom choices on demand, see (Alon et al., 1986; Dietzfelbinger, 1996).

Let x' be a minimizer to $\min_x f_w(Bx)$. By Theorem 2, Theorem 6, Lemma 5.1 and Lemma 2.1 we have

$$\begin{aligned} &f_w(Bx) \\ &\geq f_u(Tx) + G^+(A'x) \\ &\geq f((Ax)^-) - 3\varepsilon f(Ax) + (1-\varepsilon)G^+(Ax) \\ &\geq (1 - 4\varepsilon)f(Ax) \\ &= \frac{1}{2}f(Ax) \end{aligned}$$

with constant probability for all $x \in \mathbb{R}^d$ simultaneously. Let x^* be a solution minimizing $f(Ax)$. By Theorem 6 and Lemma B.8 it holds that

$$\begin{aligned} &\mathbb{E}\left(f_w(Tx^*) + G^+(A'x^*)\right) \\ &\leq \alpha\left(f\left((A'x^*)^-\right) + G^+(Ax^*)\right) \end{aligned}$$

for $\alpha = O(\log(n))$ Thus using Markov's inequality we have $f_w(Tx^*) + G^+(A'x^*) \leq 2\alpha\left(f((Ax^*)) + G^+(Ax^*)\right)$ with probability at least $1/2$. Since for any $\mu < \infty$ the optimal value of $f(Ax)$ is greater or equal to $\ln(2)$, i.e., there is at least one missclassification, the above inequality and Lemma 2.1 imply

$$\begin{aligned} f(Ax') &\leq 2f_w(Bx') \\ &\leq 2\left(f_u(Tx') + G^+(A'x') + \ln(2)\right) \\ &\leq 2\left(f_u(Tx^*) + G^+(A'x^*)\right) + 2\ln(2) \\ &\leq 2\left(2\alpha\left(f(Ax^*) + G^+(Ax^*)\right)\right) + 2\ln(2) \\ &\leq 10\alpha f(Ax^*) \end{aligned}$$

with probability at least $1/2$, which proves the first part of the theorem.

The last part of the theorem can be derived by slightly changing the logistic loss function similar to a Ky Fan norm. More precisely A'' can be split into blocks A_h for levels $h = 0, \ldots, h_{\max}$. Then we define $f_{w,c}(Bx) = f_u(Tx) + f_{w',c}(A''x)$ where

$$f_{w',c}(A''x) := \sum_h \sum_{i \in [K]} g((a''x)_{\pi(i,h)})$$

where $(a''x)_{\pi(i,h)}$ denotes the i'th largest entry of A_hx. The modified function thus omits for any fixed x all but

the largest K entries on each level. Lemma 5.1 can now be adjusted to show that $G_c^+(A'x) \leq f_{w',c}(A''x) \leq G_c^+(A'x) + \ln(2)$. Then we can use Theorem 5 to show that $f(Ax'') \leq 10C(Ax^*)$. The only other change in the proof is that G^+ gets replaced by G_c^+. □

Further we get the following corollary:

Corollary 5.2. *Let $A \in \mathbb{R}^{n\times d}$ be a μ-complex matrix for some bounded $1 \leq \mu < \infty$. There is an algorithm that solves logistic regression in $O(\texttt{nnz}(A) + \text{poly}(\mu d \log n))$ time up to a constant factor with constant probability.*

6. Experiments

Our results can be reproduced with our open Python implementation available at https://github.com/cxan96/oblivious-sketching-logreg. We compare our oblivious `LogReg`-sketch algorithm with uniform sampling (UNI), stochastic gradient descent (SGD), and the ℓ_2-leverage score (L2S) coreset from (Munteanu et al., 2018). SGD and L2S work only on data presented row-by-row. L2S requires two passes and is not data oblivious. We additionally note that the actual distribution is taking the *square root* of the ℓ_2-leverage scores as an approximation to ℓ_1 which is not covered by one-pass online leverage score algorithms given by (Cohen et al., 2020; Chhaya et al., 2020). We do not compare to the other coresets surveyed in the related work because they rely on regularization, and do not apply to the plain unconstrained logistic regression studied in this paper. SGD is allowed one entire pass over the data to be comparable to our sketch. Its plot thus displays a flat line showing the final result as a baseline. For all other data reduction methods, the optimization is done with standard optimizers from the scikit learn library[1]. The covertype and kddcup data sets are loaded automatically by our code from the scikit library, and webspam data is loaded from the LIBSVM data repository[2]. Additional details on the size and dimensions of the data sets are in the supplementary, Section E. The synthetic data is constructed so that it has n points in one place and two more points are placed in such a way that the leverage score vectors will be $(\frac{1}{n}, \ldots, \frac{1}{n}, \frac{1}{2}, \frac{1}{2})$ for ℓ_1, $(\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}}, \frac{1}{2}, \frac{1}{2})$ for ℓ_2 and it is crucial for any reasonable approximation to find those two points. The probability to see one of them is roughly $\frac{1}{n}$ for UNI and SGD, and $\frac{1}{\sqrt{n}}$ for L2S, but for ℓ_1-leverage scores it will be $1/2$ and thus the points will be heavy hitters and sketched well in separate buckets with constant probability (see Section B.1). The `LogReg`-sketch uses $h_{\max} + 1 = 3$ levels and one level of uniform sampling. By the Ky Fan argument all but the largest 25% entries are cut off at each level. The other algorithms were run using their standard parameters. We varied the target size of all reduction algorithms in thirty equal steps and calculated the approximation ratio $f(A\tilde{x})/f(Ax^*)$ where x^* is the solution returned on the full problem and $\tilde{x}$ is the solution returned on the reduced version. We repeated each experiment twenty times and displayed the median among all repetitions in Figure 1.

For the real data that seem easy to subsample uniformly, we show what happens if we introduce random Gaussian $N(0, 10^2)$ noise to 1% of data to simulate adversarial corruptions; displayed in Figure 1.

Finally we assess the sampling time as well as the total running time (including the subsequent optimization) vs. the accuracy of our sketch displayed in Figure 1.

Further plots can be found in the supplementary, Section E.

Discussion. The overall picture is that `LogReg`-sketch never performs much worse than its competitors, even for data that is particularly easy to handle for UNI and SGD (see covertype and webspam). On the kddcup data we see that `LogReg`-sketch improves slowly with increasing sketch sizes and performs slightly better than UNI. Here L2S clearly performs best. However, we emphasize again that L2S can choose its sampling distribution adaptively to the data and requires two passes in row-order. In contrast `LogReg`-sketch makes its random choices obliviously to the data and allows single-pass turnstile streaming. The higher flexibility justifies slightly weaker (but still competitive) approximations. On the synthetic data we see the theoretical claims confirmed. By construction UNI and SGD (not in the plot since its median approximation ratio exceeds 1000) have no chance to give a good approximation on a sublinear sample. L2S starts to converge at about $\Theta(\sqrt{n})$ samples. `LogReg`-sketch has roughly zero error even for very small constant sketch sizes.

When noise is added to corrupt a small number of data points, we see that UNI is not robust and its accuracy deteriorates. In comparison our Sketch (and L2S) are unaffected in this adversarial but natural setting due to their worst case guarantees.

The sketching time of our Sketch is slightly slower but closest to UNI, and the adaptive L2S takes much longer to construct the coreset. When the subsequent optimization is included, we see that the Sketch sometimes becomes even faster than UNI, which indicates that the Sketch produces a summary that is better preconditioned for the optimizer than UNI.

In summary `LogReg`-sketch provably produces good results that are competitive with the other methods and better than expected from the theoretical $O(1)$ approximation. It is weaker only when compared to the adaptive sampling algorithm that is not applicable in several challenging com-

[1] https://scikit-learn.org/

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

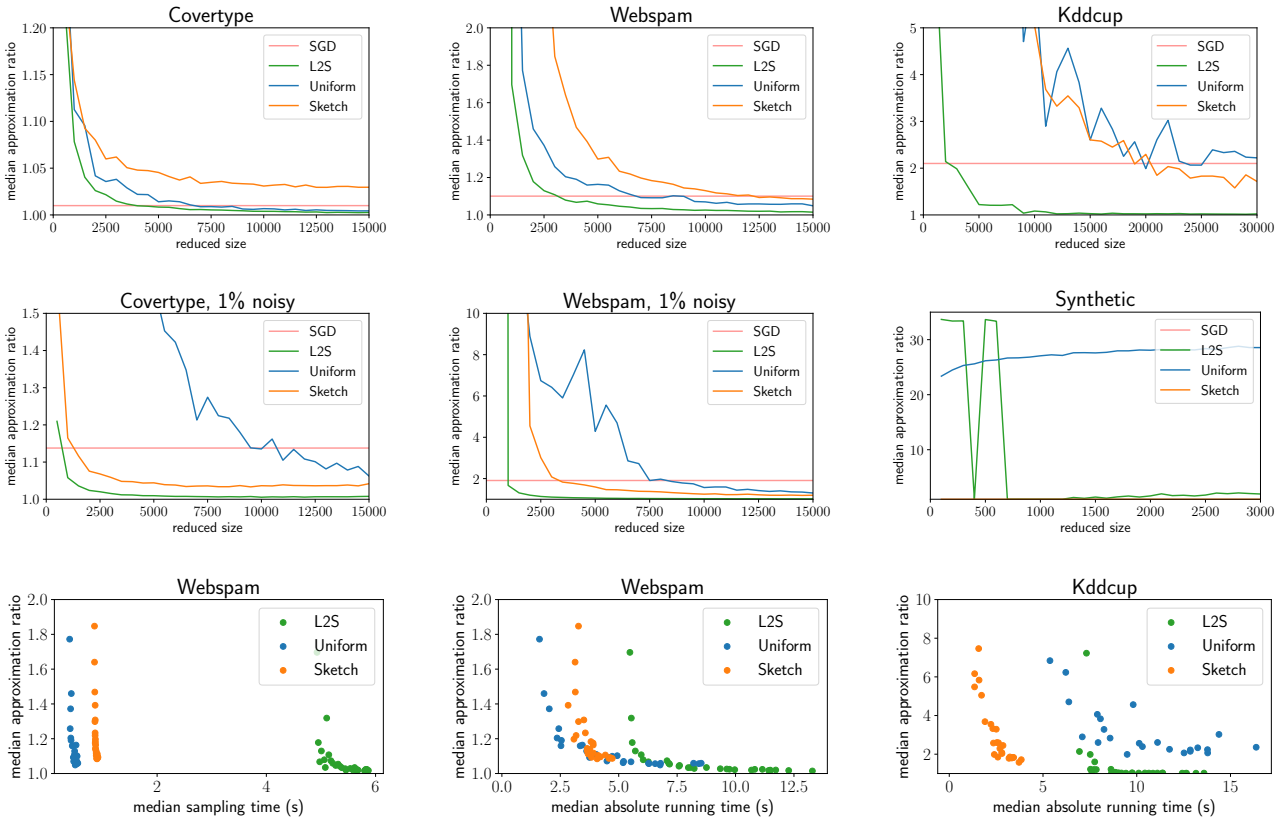

Figure 1. Comparison of the approximation ratios with and without added noise. Comparison of sketching resp. sampling times vs. accuracy. Comparison of total running times including optimization vs. accuracy.

putational settings motivated in the introduction. We also demonstrated that UNI and SGD have no error guarantees under corruptions and in the worst case where `LogReg`-sketch even outperforms the adaptive algorithm. In all cases `LogReg`-sketch performs almost the same or better compared to its competitors and comes with higher flexibility in the aforementioned computational scenarios.

7. Conclusion

We developed the first data oblivious sketch for a generalized linear model, specifically for logistic regression, which is an important model for classification (Shalev-Shwartz & Ben-David, 2014) and estimation of Bernoulli probabilities (McCullagh & Nelder, 1989). The sketching matrix can be drawn from a data-independent distribution over sparse random matrices which is simple to implement and can be applied to a data matrix A over a turnstile data stream in input-sparsity time. This is important and has advantages over existing coreset constructions when it comes to high-velocity streaming applications and when data is not presented in row-order but in an arbitrary unstructured way. The resulting sketch of polylogarithmic size can be put in any solver for weighted logistic regression and yields an $O(\log n)$-approximation. We also showed how the same sketch can be slightly adapted to give an $O(1)$-approximation. Our experiments demonstrate that those sketching techniques are useful and competitive to uniform sampling, SGD, and to state of the art coresets.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. Alexander Munteanu and Simon Omlor were supported by the German Science Foundation (DFG), Collaborative Research Center SFB 876, project C4 and by the Dortmund Data Science Center (DoDSc). D. Woodruff would like to thank NSF grant No. CCF-181584, Office of Naval Research (ONR) grant N00014-18-1-256, and a Simons Investigator Award. We thank Christian Peters and Alexander Neuhaus for their help with the experiments.

References

Ai, Y., Hu, W., Li, Y., and Woodruff, D. P. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pp. 20:1–20:22, 2016.

Alon, N., Babai, L., and Itai, A. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986.

Bentley, J. L. and Saxe, J. B. Decomposable searching problems I: Static-to-dynamic transformation. *J. Algorithms*, 1(4):301–358, 1980.

Braverman, V. and Ostrovsky, R. Zero-one frequency laws. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, (STOC)*, pp. 281–290, 2010.

Braverman, V., Feldman, D., and Lang, H. New frameworks for offline and streaming coreset constructions. *CoRR*, abs/1612.00889, 2016.

Braverman, V., Charikar, M., Kuszmaul, W., Woodruff, D. P., and Yang, L. F. The one-way communication complexity of dynamic time warping distance. In *35th International Symposium on Computational Geometry, (SoCG)*, pp. 16:1–16:15, 2019.

Chhaya, R., Choudhari, J., Dasgupta, A., and Shit, S. Streaming coresets for symmetric tensor factorization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1855–1865, 2020.

Clarkson, K. L. and Woodruff, D. P. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, (STOC)*, pp. 205–214, 2009.

Clarkson, K. L. and Woodruff, D. P. Sketching for M-estimators: A unified approach to robust regression. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 921–939, 2015.

Clarkson, K. L., Wang, R., and Woodruff, D. P. Dimensionality reduction for tukey regression. In *Proceedings of the 36th International Conference on Machine Learning, (ICML)*, pp. 1262–1271, 2019.

Cohen, M. B., Musco, C., and Pachocki, J. Online row sampling. *Theory Comput.*, 16:1–25, 2020.

Cormode, G. and Muthukrishnan, S. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.

Dasgupta, A., Drineas, P., Harb, B., Kumar, R., and Mahoney, M. W. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. Comput.*, 38(5):2060–2078, 2009.

Dietzfelbinger, M. Universal hashing and k-wise independent random variables via integer arithmetic without primes. In *Proc. of the 13th Annual Symposium on Theoretical Aspects of Computer Science, (STACS)*, pp. 569–580, 1996.

Feldman, D. Core-sets: An updated survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 10(1), 2020.

Feldman, D., Schmidt, M., and Sohler, C. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM J. Comput.*, 49(3): 601–657, 2020.

Gajjar, A. and Musco, C. Subspace embeddings under nonlinear transformations. *CoRR*, abs/2010.02264, 2020.

Geppert, L. N., Ickstadt, K., Munteanu, A., Quedenfeld, J., and Sohler, C. Random projections for bayesian regression. *Stat. Comput.*, 27(1):79–101, 2017.

Gessert, F., Wingerath, W., Friedrich, S., and Ritter, N. NoSQL database systems: a survey and decision guidance. *Computer Science - Research and Development*, 32 (3-4):353–365, 2017.

Huggins, J. H., Campbell, T., and Broderick, T. Coresets for scalable Bayesian logistic regression. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 4080–4088, 2016.

Indyk, P. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53 (3):307–323, 2006.

Indyk, P. and Woodruff, D. P. Optimal approximations of the frequency moments of data streams. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, (STOC)*, pp. 202–208, 2005.

Kannan, R., Vempala, S., and Woodruff, D. P. Principal component analysis and higher correlations for distributed data. In *Proceedings of the 27th Conference on Learning Theory (COLT)*, pp. 1040–1057, 2014.

Kearns, M. J. and Vazirani, U. V. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, 1994.

Langberg, M. and Schulman, L. J. Universal ε-approximators for integrals. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 598–607, 2010.

Li, Y., Nguyen, H. L., and Woodruff, D. P. Turnstile streaming algorithms might as well be linear sketches. In *Symposium on Theory of Computing, (STOC)*, pp. 174–183, 2014.

Li, Y., Woodruff, D. P., and Yasuda, T. Exponentially improved dimensionality reduction for ℓ_1: Subspace embeddings and independence testing. *CoRR*, abs/2104.12946, 2021.

Mahoney, M. W. Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.*, 3(2):123–224, 2011.

Maurer, A. A bound on the deviation probability for sums of non-negative random variables. *Journal of Inequalities in Pure & Applied Mathematics*, 4(1):1–6, 2003.

McCullagh, P. and Nelder, J. A. *Generalized Linear Models*. Chapman & Hall, London, 1989.

Meintrup, S., Munteanu, A., and Rohde, D. Random projections and sampling algorithms for clustering of high-dimensional polygonal curves. In *Advances in Neural Information Processing Systems 32, (NeurIPS)*, pp. 12807–12817, 2019.

Molina, A., Munteanu, A., and Kersting, K. Core dependency networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI)*, pp. 3820–3827. AAAI Press, 2018.

Munteanu, A. Sketches and coresets for large-scale statistical data analysis, 2019. *12th International Conference on Computational and Methodological Statistics, (CM-Statistics), 2019*.

Munteanu, A. and Schwiegelshohn, C. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *Künstliche Intell.*, 32 (1):37–53, 2018.

Munteanu, A., Schwiegelshohn, C., Sohler, C., and Woodruff, D. P. On coresets for logistic regression. In *Advances in Neural Information Processing Systems 31, (NeurIPS)*, pp. 6562–6571, 2018.

Muthukrishnan, S. Data streams: Algorithms and applications. *Found. Trends Theor. Comput. Sci.*, 1(2), 2005.

Nelson, J. Sketching and streaming algorithms, 2020. *Tutorial held at Advances in Neural Information Processing Systems 33 (NeurIPS), 2020*.

Phillips, J. M. Coresets and sketches. In *Handbook of Discrete and Computational Geometry*, pp. 1269–1288. Chapman and Hall/CRC, Boca Raton, 3rd edition, 2017.

Rohr, D. Data processing and online reconstruction. *arXiv preprint arXiv:1811.11485*, 2018.

Samadian, A., Pruhs, K., Moseley, B., Im, S., and Curtin, R. R. Unconditional coresets for regularized loss minimization. In *The 23rd International Conference on Artificial Intelligence and Statistics, (AISTATS)*, pp. 482–492, 2020.

Shalev-Shwartz, S. and Ben-David, S. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

Siddiqa, A., Karim, A., and Gani, A. Big Data storage technologies: A survey. *Frontiers of Information Technology & Electronic Engineering*, 18(8):1040–1070, 2017.

Sohler, C. and Woodruff, D. P. Subspace embeddings for the ℓ_1-norm with applications. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, (STOC)*, pp. 755–764, 2011.

Stolpe, M., Bhaduri, K., Das, K., and Morik, K. Anomaly detection in vertically partitioned data by distributed core vector machines. In *Machine Learning and Knowledge Discovery in Databases - European Conference, (ECML) (PKDD)*, pp. 321–336, 2013.

Tolochinsky, E. and Feldman, D. Coresets for monotonic functions with applications to deep learning. *CoRR*, abs/1802.07382, 2018.

Tukan, M., Maalouf, A., and Feldman, D. Coresets for near-convex functions. In *Advances in Neural Information Processing Systems 33, (NeurIPS)*, 2020.

van den Brand, J., Peng, B., Song, Z., and Weinstein, O. Training (overparametrized) neural networks in near-linear time. *CoRR*, abs/2006.11648, 2020.

Verbin, E. and Zhang, Q. Rademacher-sketch: A dimensionality-reducing embedding for sum-product norms, with an application to earth-mover distance. In *Automata, Languages, and Programming - 39th International Colloquium, (ICALP)*, pp. 834–845, 2012.

Wang, R. and Woodruff, D. P. Tight bounds for ℓ_p oblivious subspace embeddings. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pp. 1825–1843, 2019.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2): 1–157, 2014.

Woodruff, D. P. and Zhang, Q. Subspace embeddings and ℓ_p-regression using exponential random variables. In *The 26th Annual Conference on Learning Theory, (COLT)*, pp. 546–567, 2013.