Generic Coreset for Scalable Learning of Monotonic Kernels: Logistic Regression, Sigmoid and more

Elad Tolochinsky [1] **Ibrahim Jubran** [1] **Dan Feldman** [1]

Abstract

Coreset (or core-set) is a small weighted *subset* Q of an input set P with respect to a given *monotonic* function $f : \mathbb{R} \to \mathbb{R}$ that *provably* approximates its fitting loss $\sum_{p\in P} f(p \cdot x)$ to *any* given $x \in \mathbb{R}^d$. Using Q we can obtain an approximation of x^* that minimizes this loss, by running *existing* optimization algorithms on Q. In this work we provide: (i) A lower bound which proves that there are sets with no coresets smaller than $n = |P|$ for general monotonic loss functions. (ii) A proof that, with an additional common regularization term and under a natural assumption that holds e.g. for logistic regression and the sigmoid activation functions, a small coreset exists for *any* input P. (iii) A generic coreset construction algorithm that computes such a small coreset Q in $O(nd + n \log n)$ time, and (iv) Experimental results with open-source code which demonstrate that our coresets are effective and are much smaller in practice than predicted in theory.

1. Introduction

Traditional algorithms in computer science and machine learning are usually tailored to handle off-line finite datasets that are stored in memory. However, many modern systems do not use this computational model. For example, GPS data from millions of smartphones, high definition images, YouTube videos, Twitter tweets, or audio signals from smart homes arrive in a streaming fashion. The era of Internet of Things (IoT) provides us with wearable devices and mini-computers that collect data sets that are being gathered by ubiquitous information-sensing mobile devices and wireless sensor networks (Hellerstein, 2008; Segaran & Hammerbacher, 2009; Feldman et al., 2013b).

Challenges. Using such devices and networks pose a series of challenges:
(i) Limited memory. In such systems, the input is an infinite stream of batches that may grow in practice to petabytes of raw data, and cannot be stored in memory. Hence, only one-pass over the data and small memory are allowed.
(ii) Parallel and distributed computations. To leverage the power of multithreading and multiple processing units (as in GPUs), we are required to design variants of our algorithms which can run in parallel. Furthermore, if the dataset is distributed among many machines, e.g. on a “cloud", there is an additional problem of non-shared memory, which may be replaced by expensive and slow communication between the machines. **(iii) Larger data.** As those devices become more widespread, the amounts of data gathered and communicated between them becomes even larger. This might pose a challenge even for the algorithms previously regarded as efficient.

Weak or no theoretical guarantees. Due to the modern computation models above, learning trivial properties of the data may become non trivial, as stated in (Feldman et al., 2013b). These problems are especially common in machine learning applications, where the common optimization problems and models may be, already in the off-line settings NP-hard.

Alternative approach. Instead of designing, from scratch, a new algorithm to solve the problem at hand, an alternative approach is to provably summarize the data into a small representative subset, and to prove that applying *existing* algorithms, both heuristics and provable methods, on this small summarizations, will yield an output which approximates the result of running the same algorithms on the original (full) data. However, applying those algorithms on this small subset will be (i) much faster, and (ii) will help handle the complex data models above.

1.1. Coresets

Coresets, which are usually a small weighted subset of the input, suggest a natural solution or at least a very generic approach to address the above challenges. Coresets have some promising theoretical guarantees, while still leveraging the success of existing heuristics, as explained above.

[1] Robotics & Big Data Labs, Computer Science Department, University of Haifa, Israel. Correspondence to: Elad Tolochinsky <eladt26@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

Coresets and machine learning. We can use the notion of coresets, see formal definition below, for improving the performance of machine learning algorithms. Most machine learning algorithms essentially solve an optimization problem over some set of training data. By constructing a coreset for this training data, we can: (i) greatly reduce the time it takes to train a model, simply by training it or tuning its hyperparameters on the (small) coreset instead of the (big) input data, and (ii) allow support for streaming, parallel, and distributed data. This is by the useful the very well known merge-and-reduce property of coresets that allow them to handle big data; see formal details e.g. in (Agarwal et al., 2004) and (Feldman, 2020). Although the coreset provides guarantees for the approximation of the Mean Squared Error (MSE) of the training data, it can be shown that for some problems, a coreset can also provide guarantee for the approximations of the generalizations error. For example, when using Bayesian inference, it was shown in (Huggins et al., 2016) that a model which is based on coreset for the log likelihood function, has a marginal likelihood which is *guaranteed* to approximate the true marginal likelihood. The same can be shown for maximum likelihood estimation. The popular measure for the goodness of fit of an estimator is the the log-likelihood ratio: $\ln \Lambda(\hat{\theta}) = \mathcal{L}(\hat{\theta}) - \sup_{\theta \in \Theta} \mathcal{L}(\theta)$. The log-likelihood ratio of a model which is based on a coreset, uniformly approximates the log-likelihood ratio of the full model. Furthermore, coresets have been shown to practically improve the generalization error for machine learning algorithms (Huggins et al., 2016; Feldman et al., 2011; Munteanu et al., 2018)

Coresets for monotonic functions. In this paper we focus on coresets for monotonic continuous functions, that is: we assume that we are given a set P of n points in $\mathbb{R}^d$, and a non-decreasing monotonic functions $f : \mathbb{R} \to \mathbb{R}_{>0}$. For a given error parameter $\varepsilon \in (0, 1)$, we wish to compute an ε*-coreset* $Q \subseteq P$, with a weight function $u : Q \to [0, \infty)$ that *provably* approximates the fitting cost of P for every $x \in \mathbb{R}^d$, up to a multiplicative factor of $1 \pm \varepsilon$, i.e.,

$$(1-\varepsilon) \sum_{p \in P} f(p \cdot x) \leq \sum_{p \in Q} w(p) f(p \cdot x) \leq (1+\varepsilon) \sum_{p \in P} f(p \cdot x);$$

see Definition 2.2. Although it seems rather theoretic, many real world problems can be formulated using similar functions, including: (i) Sigmoid, (ii) Logistic regression, (iii) SVM, (iv) Linear classifiers, (v) ℓ_p regression, and (vi) Gaussian Mixture Models (GMMs); for further details and examples we refer the reader to the surveys (Agarwal et al., 2005; 2013; Phillips, 2016; Braverman et al., 2016; Bachem et al., 2017; Feldman, 2020; Maalouf et al., 2021).

1.2. Our contribution

(i) We provide an impossibility bound that proves that, for non-decreasing monotonic loss function, there are no small coresets in general. We do this by providing an example of an input set of points P, for which no coreset of size smaller than $|P|$ exists; see Section 3.

(ii) Following the bound above, we can either give up on the generic coreset paradigm, or add natural assumptions and modifications to the targeted functions f. In this paper we choose the second option; We add a regularization term to the loss function, which, in most cases, is added anyway to avoid overfitting (Schölkopf et al., 2002; Bishop, 1995). In fact, in some cases, this new term is crucial as some functions are minimized only for x approaching infinity if this term is omitted. For example, the regularization term we add to the sigmoid function is $\|x\|_2^2 / k$, where $k > 0$ defines the trade-off between minimizing the function and the complexity of the set of parameters. While minimizing such functions may still be NP-hard (Šíma, 2002), we prove that a small coreset Q exists for *any* input set P, for the sigmoid and logistic regression functions; see Section 5. However, the proof holds for a wider family of functions.

(iii) We provide a generic algorithm that computes the coreset Q above in $O(nd + n \log n)$ time. Unlike most existing works, our algorithm can construct a coreset for the sigmoid and logistic regression functions, as well as a wider set of functions; see Algorithm 1.

(iv) Open source code for our algorithms is given (Code, 2022), along with extensive experimental results on both synthetic and real-world public datasets; see Section 6.

1.3. Related Work

In (Har-Peled, 2006), Har-Peled shows how to construct a coreset of one dimensional points sets $(d = 1)$ for sums of single variable real valued functions. In the scope of machine learning most of the research involves clustering techniques (Feldman et al., 2013a; Feldman & Schulman, 2012; Jubran et al., 2020; Feldman et al., 2007; Cohen-Addad et al., 2021; Braverman et al., 2021) and regressions (Boutsidis et al., 2013; Dasgupta et al., 2009; Zheng & Phillips, 2017), including a recent coreset for decision trees (Jubran et al., 2021). Several coresets were constructed for unsupervised learning problems including coresets for Gaussian mixture models (Feldman et al., 2011), and SVM (Tsang et al., 2005; Har-Peled et al., 2007). Other works handle general families of supervised learning problems (Tukan et al., 2020; Maalouf et al., 2019).

The work by (Huggins et al., 2016) introduces lower bounds on the total sensitivity of the logistic regression problem that is used in this paper. It also introduces an upper bound for the total sensitivity and coreset size based on k-clustering coresets. However the bounds hold only for input set P from very specific distributions (roughly, when P is well separated into k clusters).

In (Munteanu et al., 2018), a lower bound of $\Omega\left(n/\log n\right)$ points, on the size of a coreset for a two dimensional logistic regression was introduced. To find a coreset, the authors have introduced a measure of the data μ, which depends on the log-ratio between the positive and negative labeled points, and have shown that for data sets in which μ is sufficiently small a coreset of size $O(poly(\log n))$ exist. Instead of imposing assumptions on the above input-related measure, in this work we add a regularization term to the loss function which, as we show, makes the coreset construction task feasible. There does not seem to be a direct relation between our work and the measure μ used in (Munteanu et al., 2018).

The main tool of this work uses the unified framework presented in (Feldman & Langberg, 2011), which was recently improved in (Braverman et al., 2016). We also use the reduction from $\mathcal{L}_\infty$ coresets that approximates $\max_{p\in P} f(p\cdot x)$ to our $\mathcal{L}_1$ coreset (sum of loss) which was introduced in (Varadarajan & Xiao, 2012).

1.4. Paper Organization

Section 2 describes preliminary results which we utilize in our coreset construction algorithm. In Section 3 we give examples of input sets which have no non-trivial coreset (i.e., smaller than the input size), for general monotonic functions. In Section 4 we introduce our main coreset construction algorithm. We then prove the correctness of this algorithm for the sigmoid and logistic regression activation functions. In Section 6 we provide our experimental results along with a discussion.

2. Preliminaries

In what follows we first describe the coreset construction framework of (Feldman & Langberg, 2011). The framework is based on a non-uniform sampling of the input, according to some importance distribution over the input points. This distribution assigns higher values to points of higher influence on the optimization problem at hand. Now, in order to keep the sample unbiased, the sampled points are reweighted reciprocal to their sampling probability. To quantify the influence of a single point on the optimization problem, Feldman and Langberg suggested in (Langberg & Schulman, 2010) a term called *sensitivity*, which we define later in this section. Using the sensitivity, a sampling-based coreset can be constructed, whose size depends on the total sensitivity over the input points, a complexity measure of the family of models, called the VC-dimension, and an error parameter $\varepsilon \in (0,1)$ that controls the trade-off between coreset size and approximation accuracy. Bounding the VC-dimension of the loss functions handled in this paper is straightforward; see formal details in Section D at the appendix. Hence, the majority of the paper is devoted to bound the sensitivity of each point.

We now formally define the sensitivity of every input point, with respect to a given problem at hand.

Definition 2.1 (Sensitivity (Feldman & Langberg, 2011; Langberg & Schulman, 2010))**.** Let (P, w, X, c) be a tuple called *query space*, where P is a finite set of elements, $w : P \to [0,\infty)$ is a weight function, X is a set called *queries* (models), and $c : P \times X \to [0,\infty)$ is a loss function. The *sensitivity* of a point $\boldsymbol{p} \in P$ with respect to (P, w, X, c) is defined as

$$s(\boldsymbol{p}) := s_{P,w,X,c}\left(\boldsymbol{p}\right) = \sup_{\boldsymbol{x}\in X} \frac{w\left(\boldsymbol{p}\right) c\left(\boldsymbol{p},\boldsymbol{x}\right)}{\sum_{\boldsymbol{p'}\in P} w\left(\boldsymbol{p'}\right) c\left(\boldsymbol{p'},\boldsymbol{x}\right)},$$

where the supremum is over every $\mathbf{x} \in X$ such that the denominator is positive . The *total sensitivity* of the query space is denoted by $t(P) := t(P, w, X, c) = \sum_{\boldsymbol{p}\in P} s(\boldsymbol{p})$.

One of the contributions of (Feldman & Langberg, 2011) is to establish a connection to the theory of range spaces and the well known VC-dimension. Informally, the (VC) dimension of a given problem is a measure of its combinatorial complexity (Anthony & Bartlett, 2009). For completeness, a formal definition is given at the appendix; see Section D.

Feldman and Langberg also show how to compute, without further assumptions, a small weighted set (Q, u), where $Q \subseteq P$, that will approximate the total cost $C\left(P, w, \boldsymbol{x}\right)$ of the input (P, w), for every query $\boldsymbol{x} \in X$, up to a multiplicative factor of $1 \pm \varepsilon$. Such a set, which we call a *coreset*, is defined as follows.

Definition 2.2 (ε-coreset)**.** Let (P, w, X, c) be a query space (see Definition 2.1), and $\varepsilon \in (0, 1)$ be an error parameter. An ε-*coreset* for (P, w, X, c) is a weighted set (Q, u) such that for every $\boldsymbol{x} \in X$,

$$\left|\sum_{\boldsymbol{p}\in P} w\left(\boldsymbol{p}\right) c\left(\boldsymbol{p},\boldsymbol{x}\right) - \sum_{\boldsymbol{q}\in Q} u\left(\boldsymbol{q}\right) c\left(\boldsymbol{q},\boldsymbol{x}\right)\right| \\ \leq \varepsilon \cdot \sum_{\boldsymbol{p}\in P} w(\boldsymbol{p}) c(\boldsymbol{p},\boldsymbol{x}).$$

In (Feldman & Langberg, 2011), a lower bound is given for the required coreset size, as a function of the total sensitivity $t(P)$. This bound was later made tighter in (Braverman et al., 2016). The following theorem describes the random sampling scheme for coreset construction using the sensitivity framework, and describes the required sample (coreset) size.

Theorem 2.3 (coreset construction (Braverman et al., 2016; Feldman & Langberg, 2011))**.** *Let* (P, w, X, c) *be a query* *space of VC-dimension d and total sensitivity t. Let $\varepsilon, \delta \in (0,1)$. Let Q be a random sample of $|Q| \geq \frac{10t}{\varepsilon^2}\left(d \log t + \log\left(\frac{1}{\delta}\right)\right)$ i.i.d points from P, such that every $\boldsymbol{p} \in P$ is sampled with probability $\frac{1}{t} \cdot s_{P,w,X,c}(\boldsymbol{p})$. Let $u(\boldsymbol{p}) = \frac{t \cdot w(\boldsymbol{p})}{s_{P,w,X,c}(\boldsymbol{p})|Q|}$ for every $\boldsymbol{p} \in Q$. Then, with probability at least $1-\delta$, (Q,u) is an ε-coreset of (P,w,X,c).*

3. Lower Bounds

In what follows, we consider query spaces $(P, w, \mathbb{R}^d, c)$, where $c(\boldsymbol{x}, \boldsymbol{p}) = f(\boldsymbol{x} \cdot \boldsymbol{p})$ for some non-decreasing monotonic function f. We prove that not all such query spaces admit a non-trivial coreset, by providing an example of an input sets P for which every coreset must be of size $|P|$.

No coreset. Consider a 2-dimensional circle $C \subseteq \mathbb{R}^3$ in 3-dimensional space, which is the intersection of the unit sphere and a non-affine plane (does not pass through the origin) that is parallel to the XY plane. For every point $\boldsymbol{p} \in P$, let $\pi_{\boldsymbol{p}}$ be a plane in $\mathbb{R}^3$ that passes through the origin which isolates $\boldsymbol{p}$ from the rest of the set, and let $\boldsymbol{x_p}$ be a vector orthogonal to $\pi_{\boldsymbol{p}}$, such that $\boldsymbol{p} \cdot \boldsymbol{x_p} > 0$; see Fig 1. Such a plane exists since the points are on a 2D circle that is not centered around the origin.

Now, for intuition, consider the logistic regression cost function: $c(\boldsymbol{x}, \boldsymbol{p}) = \log(1 + e^{\boldsymbol{p} \cdot \boldsymbol{x}})$. Let $\boldsymbol{p} \in P$ and let $\boldsymbol{x_p}$ be the query vector orthogonal to the plane $\pi_{\boldsymbol{p}}$ which separates $\boldsymbol{p}$ from the rest of the set. Since $\boldsymbol{p}$ is the only point on the positive side of $\boldsymbol{x_p}$, it holds that $\boldsymbol{p} \cdot \boldsymbol{x_p} > 0$ whereas for every other point $\boldsymbol{p'}$, $\boldsymbol{p'} \cdot \boldsymbol{x_p} < 0$. Moreover as $\|\boldsymbol{x_p}\|$ grows, $\boldsymbol{p} \cdot \boldsymbol{x_p}$ goes to ∞ and $\boldsymbol{p'} \cdot \boldsymbol{x_p}$ grows to $-\infty$. Thus the cost $c(\boldsymbol{x_p}, \boldsymbol{p}) = \log(1 + e^{\boldsymbol{p} \cdot \boldsymbol{x_p}})$ of $\boldsymbol{p}$, goes to ∞ and the cost of every other point goes to 0. Therefore, $\boldsymbol{p}$ has a sensitivity of 1. In this case, intuitively, every coreset must include $\boldsymbol{p}$ or else it cannot provide a good approximation to the cost of the original (full) set. Since this argument holds for every $\boldsymbol{p} \in P$, any coreset for P must include all points in P. Thus, no non-trivial coreset exists in this case. Putting it differently, the above discussion shows that if the sensitivity of every point in P is 1 then the size of every coreset is $\Omega(n)$; see Lemma A.1 in the appendix for a formal statement.

Note that the above holds true not only for logistic regression but for any function f that satisfies $\lim_{x \to \infty} \frac{f(-x)}{f(x)} = 0$. This is formally stated in the following theorem. A formal proof is given in Section A of the appendix.

Theorem 3.1. *Let $f : \mathbb{R} \to (0, \infty)$ be a non-decreasing monotonic function that satisfies $\lim_{x \to \infty} \frac{f(-x)}{f(x)} = 0$, and let $c(\boldsymbol{x}, \boldsymbol{p}) = f(\boldsymbol{x} \cdot \boldsymbol{p})$ for every $\boldsymbol{x}, \boldsymbol{p} \in \mathbb{R}^d$. Let $\varepsilon \in (0,1)$, $n \geq 1$ be an integer, and $w : \mathbb{R}^d \to (0, \infty)$. There is a set $P \subset \mathbb{R}^d$ of $|P| = n$ points such that if (Q,u) is an ε-coreset of $\left(P, w, \mathbb{R}^d, c\right)$ then $Q = P$.*

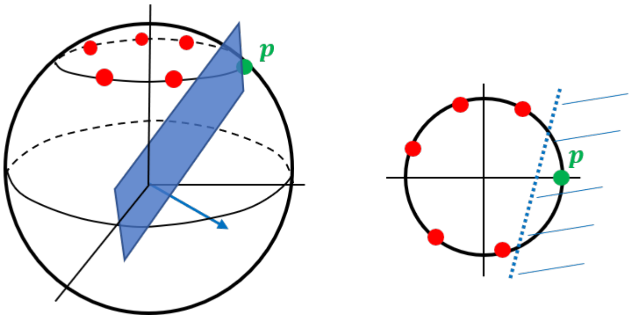


Figure 1. (Left): A set of points P in $\mathbb{R}^3$ (red and green points), a plane $\pi_{\boldsymbol{p}}$ separating p from $P \setminus \{p\}$ and the vector $\boldsymbol{x_p}$ orthogonal to $\pi_{\boldsymbol{p}}$. (Right): A top-down view of the data on the left. The dotted line ℓ is the intersection of $\pi_{\boldsymbol{p}}$ and the plane containing P. All the points to the right (left) of ℓ are projected onto the positive (negative) side of $\boldsymbol{x_p}$.

A different lower bound was also suggested in (Munteanu et al., 2018). However, while our formulation of the problem at hand is not directly comparable to the formulation in (Munteanu et al., 2018), their bound is $\Omega(n/\log n)$, which holds only for logistic regression. It also handles only a specific one-pass streaming data model. We suggest a different bound of $\Omega(n)$ which holds for a wider range of functions and computational models.

Adding assumptions. The above counter-example and formal claim motivate the necessity of adding assumptions on the loss function, as described in the following section. Mainly, a regularization term needs to be added. This term is usually added anyway, both in theory and in practice, to reduce the complexity of the model and avoid overfitting.

4. Coresets For Monotonic Bounded Functions

From the previous section, we conclude that an additional constraint must be imposed on the problem at hand in order to construct a small coreset. To better understand the required constraint, recall the reason for the lower bound from the example at Section 3; the (problematic) points with sensitivity 1 were the points which had very large values of $\boldsymbol{x} \cdot \boldsymbol{p}$. This can happen when $\|\boldsymbol{x}\|$ is very large or when $\|\boldsymbol{p}\|$ is large. For the moment, assume that $\|\boldsymbol{p}\|$ is small (we will later see how $\|\boldsymbol{p}\|$ affects the size of the coreset). The standard technique for preventing the parameters from growing too large is to add a regularization term, which is widely used in many real world problems (Schölkopf et al., 2002; Kukačka et al., 2017). As it happens to be, adding a regularization term also advances us towards our goal of constructing a coreset, as was also noted e.g., in (Samadian et al., 2020; Tukan et al., 2021). To see this, consider a regularized variant of the loss function: $c(\boldsymbol{x}, \boldsymbol{p}) = f(\boldsymbol{x} \cdot \boldsymbol{p}) + \frac{\|\boldsymbol{x}\|}{k}$. Since f is bounded, when $\|\boldsymbol{x}\|$ grows to infinity the value of the regularization dominates the loss. Thus, in this case, all points have approximately the same loss, and are all equally unimportant. In other words, the sensitivity of those points can not be 1.

Comparison to prior works. Coresets for the problem at hand are not new. However, our coreset has a provable upper bound on its size for every given dataset P. This bound depends only on the desired error ε. However, the size of the logistic regression coreset, e.g., as in (Mai et al., 2021) depends on a data-dependent parameter $\mu(P)$ that may be arbitrarily large for some datasets.

The common case for the value of the regularization parameter k is $k = n^{1-\kappa}$ for $\kappa \in (0, 1)$; see e.g., in (Curtin et al., 2019; Mai et al., 2021). In practice, we observed that the values of k have only a small effect on the coresets approximation accuracy; see Section 6.

4.1. $\mathcal{L}_\infty$ coresets

We now address the common case, in which for some $\boldsymbol{x} \in X$ and every two points, $\boldsymbol{p}_1$, $\boldsymbol{p}_2 \in P$ the values of $\boldsymbol{p}_1 \cdot \boldsymbol{x}$ and $\boldsymbol{p}_2 \cdot \boldsymbol{x}$ do not greatly differ. To do so, we will reduce our problem to the problem of constructing an $\mathcal{L}_\infty$ coreset, which is defined as follows.

Definition 4.1. ($\mathcal{L}_\infty$ coreset(Varadarajan & Xiao, 2012)) Let (P,w,X,c) be a query space and $\varepsilon > 0$. An $\varepsilon - \mathcal{L}_\infty$ coreset is a subset $Q \subseteq P$ such that $\max_{\boldsymbol{p} \in P} c(\boldsymbol{p}, \boldsymbol{x}) \leq (1 + \varepsilon) \max_{\boldsymbol{q} \in Q} c(\boldsymbol{q}, \boldsymbol{x})$ for every $\boldsymbol{x} \in X$.

We will now focus on constructing an $\mathcal{L}_\infty$ coreset. We will then show how to leverage this $\mathcal{L}_\infty$ coreset to obtain a coreset as defined in Definition 2.2.

Consider a monotonic non-decreasing function $f : \mathbb{R} \to (0, M]$, a query $\boldsymbol{x} \in X$ and a point $\boldsymbol{p} \in P$ such that $\boldsymbol{p} \cdot \boldsymbol{x} > 0$. Since f is a monotonic function, $f(0) \leq f(\boldsymbol{p} \cdot \boldsymbol{x})$. Hence,

$$\max_{\boldsymbol{p}' \in P} f(\boldsymbol{p}' \cdot \boldsymbol{x}) \leq M = \frac{M}{f(0)} f(0) \leq \frac{M}{f(0)} f(\boldsymbol{p} \cdot \boldsymbol{x}),$$

Therefore, for a query $\boldsymbol{x}$, if a point $\boldsymbol{p}$ falls on the positive side of the line defined by $\boldsymbol{x}$ we can say this point is an $\mathcal{L}_\infty$ coreset. But what if the point falls on the negative side of the line? Since f is monotonic, we know that if $\boldsymbol{p} \cdot \boldsymbol{x} < 0$ then, $f(\boldsymbol{p} \cdot \boldsymbol{x}) < f(-\boldsymbol{p} \cdot \boldsymbol{x})$, but if f is sufficiently "well behaved" then as long as the distance between $-\boldsymbol{p} \cdot \boldsymbol{x}$ and $\boldsymbol{p} \cdot \boldsymbol{x}$ is not too large, then the distance between $f(-\boldsymbol{p} \cdot \boldsymbol{x})$ and $f(\boldsymbol{p} \cdot \boldsymbol{x})$ is also bounded. Specifically, we can assume there is a constant $b > 0$ such that

$$f(-\boldsymbol{p} \cdot \boldsymbol{x}) < b \cdot f(\boldsymbol{p} \cdot \boldsymbol{x})$$

which implies that even if $\boldsymbol{p}$ falls on the negative side of the line, then $\boldsymbol{p}$ is an $\mathcal{L}_\infty$ coreset.

Assumptions and conclusions made so far. Before we conclude the results of this section, we must conduct the assumptions and conclusions we have made so far. We have assumed that the distance between $-\boldsymbol{p} \cdot \boldsymbol{x}$ and $\boldsymbol{p} \cdot \boldsymbol{x}$ is not too large. We can bound the distance as follows:

$$| - \boldsymbol{p} \cdot \boldsymbol{x} - (\boldsymbol{p} \cdot \boldsymbol{x})| = |2\boldsymbol{p} \cdot \boldsymbol{x}| \leq 2 \|\boldsymbol{x}\| \|\boldsymbol{p}\| .$$

From the discussion in the beginning of the section, adding regularization will guarantee that $\|\boldsymbol{x}\|$ can not grow arbitrarily large. As for the $\|\boldsymbol{p}\|$ term, we expect the coreset to be somehow affected by this term in order to ensure the above property. Indeed, this is one of the main terms which affect the sensitivity of the input points. Hence, the final coreset will be more likely to choose points with larger norm.

We conclude that every point $\boldsymbol{p} \in P$ is an $\varepsilon - \mathcal{L}_\infty$ coreset for sufficiently large ε that depends on properties of the function (M, $f(0)$ and b) and on $\|\boldsymbol{p}\|$. This is formally stated in the following lemma.

Lemma 4.2 ($\mathcal{L}_\infty$ coresets)**.** *Let $P \subset \mathbb{R}^d$ be a finite set, $M, k > 0$ be constants, $f : \mathbb{R} \to (0, M]$ be non-decreasing function and $g : [0, \infty) \to [0, \infty)$ be a function. For every $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{p} \in P$ define $c_k(\boldsymbol{p}, \boldsymbol{x}) = f(\boldsymbol{p} \cdot \boldsymbol{x}) + \frac{g(\|\boldsymbol{x}\|)}{k}$. Put $\boldsymbol{p} \in P$ and suppose there is $b_{\boldsymbol{p}} > 0$ such that for every $z > 0$, $f(\|\boldsymbol{p}\| z) + \frac{g(z)}{k} \leq b_{\boldsymbol{p}} \left(f(-\|\boldsymbol{p}\| z) + \frac{g(z)}{k} \right)$. Then $\{\boldsymbol{p}\}$ is an $\varepsilon - \mathcal{L}_\infty$ coreset with $\varepsilon = \frac{M}{f(0)} (b_{\boldsymbol{p}} + 1) - 1$.*

Algorithm 1 MONOTONIC-CORESET(P, k, m)

```
Input: A set $P = \{p_1, \cdots, p_n\}$ of points in $\mathbb{R}^d$,
       a real valued regularization term $k > 0$, and
       an integer $m \geq 1$.
Output: A pair $(Q, u)$ where $|Q| = m$ and $u : Q \to [0, \infty)$; see Theorems 5.1-5.2.
Sort the points in $P = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_n\}$ by their length, i.e., $\|\boldsymbol{p}_1\| \leq \cdots \leq \|\boldsymbol{p}_n\|$ .
$s(\boldsymbol{p}_i) := \dfrac{c \cdot \sqrt{k} \|\boldsymbol{p}_i\| + 2}{i}$ for every $i \in [n]$ {$c$ is a sufficiently large constant.}
Set $t \leftarrow \sum_{i=1}^{n} s(\boldsymbol{p}_i)$
Pick an i.i.d random sample $Q \subseteq P$ of $|Q| \geq \min\{m, n\}$ from $P$, where every $\boldsymbol{p} \in P$ is chosen with probability $s(\boldsymbol{p})/t$.
$u(\boldsymbol{p}) := \dfrac{1}{|Q| \operatorname{Prob}(\boldsymbol{p})}$ for every $\boldsymbol{p} \in Q$
return $(Q, u)$
```

4.2. From $\mathcal{L}_\infty$ coresets to coresets

We now describe how to leverage an $\mathcal{L}_\infty$ coreset to bound the sensitivity of every input point.

Intuition behind Algorithm 1. Let Q be an $\varepsilon - \mathcal{L}_\infty$ coreset of P. Intuitively, since the points in Q provide a $(1 + \varepsilon)$-approximation to the maximal cost, we would require a random sampling scheme to choose these points with relatively high probability (compared to points in $P \setminus Q$). Let Q_2 be an $\varepsilon - \mathcal{L}_\infty$ coreset of $P \setminus Q$. Using the same reasoning, we would require the probability of sampling points in Q_2 to be greater then the probability of sampling a point in $P \setminus Q \setminus Q_2$, but less than the probability of sampling a point in Q. Using this logic, we can continue to construct $\mathcal{L}_\infty$ coresets and remove them from the set of remaining points. The probability of every point $\boldsymbol{p} \in P$ should intuitively be proportional to $\frac{1}{i}$, where i is the index of the $\mathcal{L}_\infty$ coreset which contains $\boldsymbol{p}$. Phrasing this differently: for every $\boldsymbol{p}$, the sensitivity of $\boldsymbol{p}$ is proportional to $\frac{1}{i}$. In (Varadarajan & Xiao, 2012) it was proven that by repeatedly constructing $\mathcal{L}_\infty$ coresets as described above, one can bound the total sensitivity and construct a coreset. Fig. 2 illustrates the above reduction.

The following lemma gives the formal statement for the algorithm described above. The lemma is based on Lemma 3.1 in (Varadarajan & Xiao, 2012).

Lemma 4.3. *Let $c : \mathbb{R}^d \times \mathbb{R}^d \to (0, \infty)$. Suppose that for some $\varepsilon \in (0,1)$ there is a non-decreasing function $\Delta_\varepsilon(n)$ so that for any $P' \subseteq \mathbb{R}^d$ of size n there is an $\varepsilon - \mathcal{L}_\infty$ coreset of size at most $\Delta_\varepsilon(n)$ for $(P', \mathbf{1}, \mathbb{R}^d, c)$. Then, for any $P \subseteq \mathbb{R}^d$ of size n we can compute an upper bound $s(p)$ on the sensitivity $s_{P,\mathbf{1},\mathbb{R}^d,c}(p)$ for each $p \in P$, so that $\sum_{p \in P} s_{P,\mathbf{1},\mathbb{R}^d,c}(p) \leq (1+\varepsilon)\Delta_\varepsilon(n) \ln n$.*

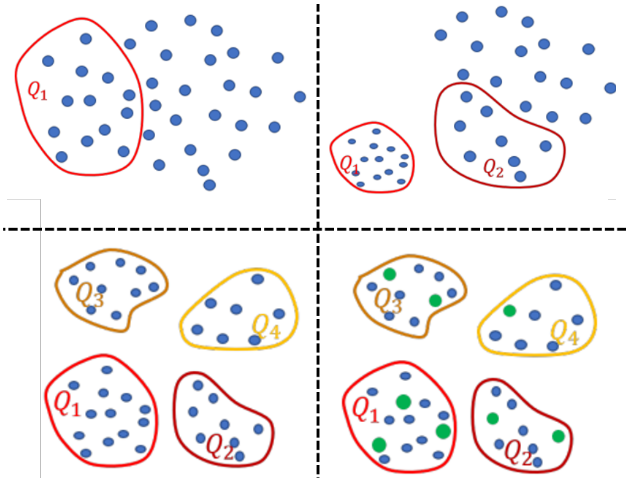


Figure 2. **Coreset construction illustration. Upper left:** Construct an $\mathcal{L}_\infty$ coreset Q_1 from P. **Upper right:** Remove Q_1 from P and construct an $\mathcal{L}_\infty$ coreset Q_2 for $P \setminus Q_1$. **Lower left:** Continue to do so until the sets $Q_1, Q_2, \cdots$ cover the entire set P. **Lower right:** Sample a subset of P, such that every point $\boldsymbol{p} \in P$ is sampled with probability proportional to $\frac{1}{i}$, where $\boldsymbol{p} \in Q_i$. The resulting set, after a re-weighting reciprocal to its sampling probability, is a coreset. See more details in the intuition behind Algorithm 1 in Section 4.2.

A minor pitfall. The algorithm described above, assumes that all of the $\mathcal{L}_\infty$ coresets have the same approximation constant ε. However, this assumption does not hold in our case since the approximation constants of the $\mathcal{L}_\infty$ coresets we have constructed in the previous section depend on $\|\boldsymbol{p}\|$. Fortunately, we can still use the same general idea as before: in every iteration of the algorithm we have multiple choices to construct an $\mathcal{L}_\infty$ coreset, we must choose the correct order of construction so the total sensitivity will be the smallest. To understand this optimal order, we need to understand how the sensitivity of a point $s(\boldsymbol{p})$ depends on the approximation constant ε. As was shown in (Varadarajan & Xiao, 2012), $s(\boldsymbol{p})$ linearly depends on $1/\varepsilon$, or in our case $\|\boldsymbol{p}\|$, and on $1/i$ where is i is the index of the point in some ordering. Thus for every point $\boldsymbol{p}$, $s(\boldsymbol{p})$ is proportional to $\|\boldsymbol{p}\| / i$. To minimize the total sensitivity we will prefer first to choose the points with smaller norms, so that the sensitivity of the points with the larger norms will be divided by a greater constant i. Algorithm 1 gives a suggested implementation for the algorithm from the discussion above and the following theorem formally states the results. A full proof is given in the appendix.

Theorem 4.4. *Let $M, k > 0$ be constants, $P \subseteq \mathbb{R}^d$ be a set of points, $f : \mathbb{R} \to (0, M]$ be a monotonic non-decreasing function, and $c_k(\mathbf{p'}, \boldsymbol{x}) = f(\mathbf{p'} \cdot \boldsymbol{x}) + \frac{g(\|x\|)}{k}$ for every $\boldsymbol{x} \in \mathbb{R}^d$ and $\mathbf{p'} \in P$. Suppose there is $b : P \to (0, \infty)$ such that for every $\mathbf{p} \in P$ and every $z > 0$ we have $f(\|\boldsymbol{p}\| z) + \frac{g(z)}{k} \leq b(\boldsymbol{p}) \left(f(-\|\boldsymbol{p}\| z) + \frac{g(z)}{k} \right)$. Let $b_{\max} \in \arg\max_{\boldsymbol{p} \in P} b(\boldsymbol{p})$, $t = (1 + \frac{M}{f(0)} b_{\max}) \ln n$, and $\varepsilon, \delta \in (0,1)$. Lastly, let d_{VC} be the VC-dimension of $(P, \mathbf{1}, \mathbb{R}^d, c_k)$. Then, there is a weighted set (Q, u), where $Q \subseteq P$ and $|Q| \in O\left(\frac{t}{\varepsilon^2}\left(d_{VC} \log t + \log \frac{1}{\delta}\right)\right)$, such that with probability at least $1 - \delta$, (Q, u) is an ε-coreset for the query space $(P, \mathbf{1}, \mathbb{R}^d, c_k)$.*

Discussion behind Theorem 4.4. The above theorem suggests a sufficient condition for the existence of a coreset, in the case of a monotonic non-decreasing function f, to which a regularization term is added. The proof of this theorem is constructive; it combines the above condition with Lemma 4.3 in order to bound the sensitivity of every input point $\boldsymbol{p} \in P$ and also gives an upper bound to the total sensitivity; see Section B.2 of the appendix. As an example, the following section constructs a coreset for the sigmoid and logistic regression activation functions by proving that the above condition is indeed met. However, the above theorem is not limited to those activation functions, and can be utilized for many other functions. Given this sensitivity upper bound, the coreset construction algorithm is straightforward: it simply samples the input set P based on the sensitivity distribution, and assigns appropriate weights to the sampled points. The only thing left to determine is the sample size required in order to achieve some predefined approximation error ε. A suggested implementation for the sigmoid and logistic regression functions is given in Algorithm 1.

5. Example Applications - Coresets for Sigmoid and Logistic Regression

In this section, we leverage the framework derived in the previous section in order to construct, as an example, a coreset for the sigmoid and logistic regression activations; see Theorems 5.1 and 5.2 respectively. The full proofs are placed in Section C.2 of the appendix.

Overview of Theorems 5.1-5.2. The following theorems construct a coreset for sums of sigmoid functions and for the logistic regression log-likelihood, for normalized input sets. Here, $f(p \cdot x)$ is the loss (e.g., a sigmoid function) of a specific input point p to a candidate classifier that is parametrized by x. The coreset approximates the sum $\sum_p f(p \cdot x)$ of this function (i.e., sum of sigmoids) over different input points.

To do so, we: (i) prove that the sufficient condition from Lemma 4.2 and Theorem 4.4 in the previous section is met for both the sigmoid and the logistic regression functions; see Lemma C.5 and Lemma C.6 in the appendix respectively. (ii) Based on the sufficient condition, we give an upper bound for the sensitivity of every input point as well as bound the total sensitivity; see Lemma C.7 and Lemma C.9 in the appendix respectively. (ii) Lastly, we combine the above with the coreset construction framework from Theorem 2.3 to obtain a provable sampling algorithm for coreset construction, as formally stated in Theorems 5.1-5.2. An important ingredient in this construction was an upper bound for the VC-dimension of the relevant query spaces. An upper bound of $O(d^2)$ for both functions is given in Section D of the appendix.

Theorem 5.1. *Let P be a set of n points in the unit ball of $\mathbb{R}^d$, $\varepsilon, \delta \in (0,1)$, $k > 0$ be a sufficiently large constant, and let $t = (1+k)\log n$. For every $p, x \in \mathbb{R}^d$, let $c_{\text{sigmoid},k}(\boldsymbol{p}, \boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{p}\cdot\boldsymbol{x}}} + \frac{\|\boldsymbol{x}\|^2}{k}$. Finally, let (Q,u) be the output of a call to* MONOTONIC-CORESET(P,k,m), *where $m \in \Omega\left(\frac{t}{\varepsilon^2}\left(d^2 \ln t + \ln \frac{1}{\delta}\right)\right)$; see Algorithm 1. Then, with probability at least $1-\delta$, (Q,u) is an ε-coreset for $(P, \mathbf{1}, \mathbb{R}^d, c_{\text{sigmoid},k})$. Moreover, $|Q| \in O(m)$, and (Q,u) can be computed in $O(nd + n\log n)$ time.*

Theorem 5.2. *Let P be a set of n points in the unit ball of $\mathbb{R}^d$, $\varepsilon, \delta \in (0,1)$, $R, k > 0$ where k is a sufficiently large constant, and $t = R\log n(1+Rk)$. For every $\boldsymbol{p} \in \mathbb{R}^d$,$\boldsymbol{x} \in B(\mathbf{0}, R)$ let $c_{\text{logistic},k}(\boldsymbol{p},\boldsymbol{x}) = \log\left(1 + e^{\boldsymbol{p}\cdot\boldsymbol{x}}\right) + \frac{\|\boldsymbol{x}\|^2}{k}$. Finally, let (Q,u) be the output of a call to* MONOTONIC-CORESET(P,k,m) *where $m \in \Omega\left(\frac{t}{\varepsilon^2}\left(d^2 \ln t + \ln \frac{1}{\delta}\right)\right)$; see Algorithm 1. Then, with probability at least $1-\delta$, (Q,u) is an ε-coreset for $(P, \mathbf{1}, \mathbb{R}^d, c_{\text{logistic},k})$. Moreover, $|Q| \in O(m)$ and (Q,u) can be computed in $O(nd + n\log n)$ time.*

Supporting additional activation functions. The above theorems give two example activation functions that our framework supports. However, the framework is not limited to those functions only. To support other functions, one must prove the sufficient condition to obtain the sensitivity upper bound, which can be then simply plugged into Algorithm 1 to obtain the desired coreset. In Section C.3 we give the formal statements for some of the additional applications of our framework, namely the support vector machine (SVM) problem.

6. Experiments

We implemented Algorithm 1 in Pythonm, and, in this section, we evaluate its empirical results both on synthetic and real-world datasets. We utilizes the sorting function from the Numpy library to implement Line 4.1 of Algorithm 1. Rather than competing with existing solvers, our coreset is simply a pre-processing step which reduces the input size. To this end, we apply existing solvers as a black box on our small coreset. The results show that a coreset of size only 1% of the original data can represent the full data with an error ε smaller than 0.001. Open-source code can be found in (Code, 2022).

Competing methods. Our main competing method is a random sampling scheme. As implied by the theoretical analysis, "important" points, i.e., with high sensitivity, are sampled with high probability in our coreset construction algorithm. However, such points are sampled with probability roughly $1/n$ using the naive uniform sampling. Hence, we expect the coreset would yield results much better than a uniform sampling scheme. With that said, we chose real-world databases with relatively uniform data, in order to demonstrate the effectiveness of our coreset even in such cases. Even in this case, the improvement over uniform sampling is consistent and usually significant.

Other existing methods are not directly comparable, since they do not optimize our same regularized objective function, and have different assumptions. This is directly stated e.g. in (Munteanu et al., 2018; Tukan et al., 2020) and (Mai et al., 2021). Moreover, we could not find an open and stable implementation for more of those methods.

Datasets used. We used the following datasets:
(i) Synthetic dataset. This data contains a set of $n = 20{,}010$ points in $\mathbb{R}^2$. $20{,}000$ of the points were generated by sampling a two dimensional normal distribution with mean $\mu_1 = (10{,}000, 10{,}000)$ and covariance matrix $\Sigma_1 = \begin{pmatrix} 0.0025 & 0 \\ 0 & 0.0025 \end{pmatrix}$ and 10 points were generated by sampling a two dimensional normal distribution with mean $\mu_2 = (-9998, -9998)$ and covariance matrix $\Sigma_2 = \begin{pmatrix} 0.0025 & 0 \\ 0 & 0.0025 \end{pmatrix}$. Our goal was to have one large clus-

ter near $10,000$ and another small cluster near $-10,000$, both with variance that was chosen uniformly from $[0, 0.1]$. No "cherry picking" here.

(ii) Bank marketing dataset (Moro et al., 2014). It contains $n = 20,000$ numerical valued records in $d = 10$ dimensional space with. The data was generated for direct marketing campaigns of a Portuguese banking institution. Each record represents a marketing call to a client, that aims to convince him/her to buy a product (bank term deposit). A binary label (yes or no) was added to each record. We used the numerical values of the records to predict if a subscription was made.

(iii) Wine Quality dataset (Cortez et al., 2009; Wang & Zhang, 2013; Elidan, 2010; Kajino et al., 2012). It contains $n = 6497$ numerical valued records in $d = 12$ dimensional space.

Experiments. We conducted the following experiments:

(i) Sigmoid Activation. For a given size m we computed a coreset of size m using Algorithm 1. We used the datasets above to produce coresets of size $5\ln(n) \leq m \leq 20\ln(n)$, where n is the size of the full data. We then normalized the data and found the optimal solution to the problem with values of $k = 100, 500, 1000, 5000$ using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) iterative method for solving unconstrained nonlinear optimization problems. We repeated the experiment with a uniform sample of size m. For each optimal solution that we have found, we computed the sum of sigmoids and denoted these "approximated solutions" by C_1 and C_2 for our algorithm and uniform sampling respectively. The "ground truth" C^k was computed using BFGS on the entire dataset. The empirical error is then defined to be $E_t = \left|\frac{C_t}{C^k} - 1\right|$ for $t = 1, 2$. For every size m we computed E_1 and E_2 100 times and calculated the mean of the results.

(ii) Logistic Regression. Similarly, we produced coresets and uniform samples of size $5\ln(n) \leq m \leq 40\ln(n)$ and maximized the regularized log-likelihood using the BFGS algorithm. For every sample size we calculated the negative test log-likelihood. Every experiment was repeated 20 times and the results were averaged. All the results are presented in Fig. 3.

Discussion. As seen in Fig. 3, our coreset outperforms the random sampling scheme as of accuracy, and is more stable (which can be seen in the standard deviation). For small sample sizes, the coreset provides a very small approximation error in practice, unlike the pessimistic theory which suggests bigger error. In this case, the coreset produces errors much smaller than the uniform sampling scheme. As the sample sizes grow, both the coreset and the random sample simply contain a big portion of the original full data, and hence their output errors decrease and also becomes more similar, as predicted. Furthermore, the coreset construction takes a neglectable amount of time from the total running time of computing a coreset and running the BFGS algorithm on the coreset. In fact, the speed-up in practice can be predicted as follows: (i) The most time consuming step in our algorithm is the sorting step, which is near linear and very efficient in practice, and (ii) The BFGS requires quadratic running time. Hence, for data of size $n = 20,000$, computing a coreset of size $\sim 1\%$ of the input and running BFGS on it would be up to two orders of magnitude faster than running BFGS on top of the full data.

Moreover, while in theory our results hold only for sufficiently large values of k, in practice we tested multiple k values and witnessed a neglectable effect on the results. This is common in coresets paper where the worst-case theoretical bounds are too pessimistic and ignore structure in the data.

7. Conclusions and Future Work

We provided a new coreset construction algorithm which computes a coreset for sums of sigmoid functions, which is NP-hard to minimize, and logistic regression, where a coreset in (Huggins et al., 2016) were suggested but with no support for regularization term, and no provable worst case bounds on the size of the coreset. Our construction algorithm is easily applicable to other functions as well. The coreset is of size near-logarithmic in the input size and can be computed in near-linear time. In most coreset paper, and in this paper in particular, coresets of size m can provably handle streaming and distributed data using $O(m)$ memory, and insertion/deletion of points in $O(m)$ time. This is using the standard and widely used techniques (Braverman et al., 2016; Lu et al., 2020).

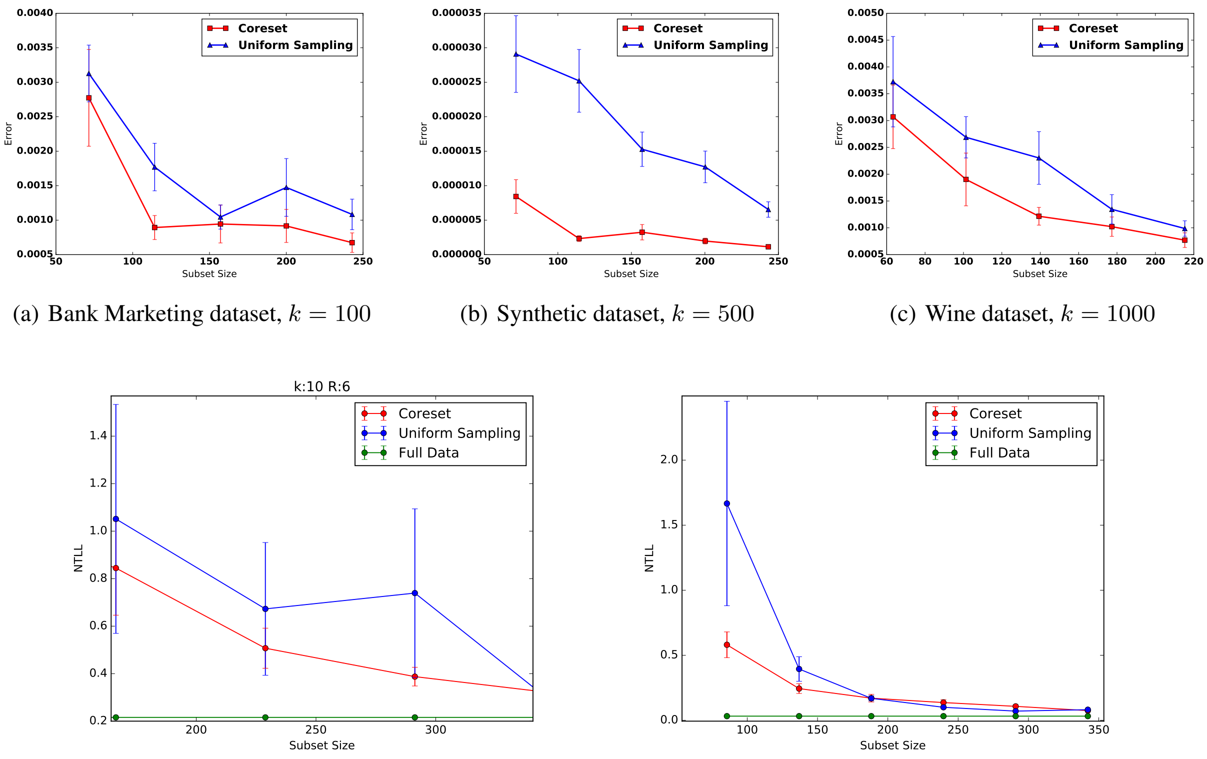
Experimental results demonstrate that our coreset outperforms a standard sampling method, both in accuracy and stability. The experiments prove that empirically, our coreset is very effective; A coreset of size less than 1% of the input suffices to produce a small error of $\varepsilon = 0.001$. Future work includes relaxing the assumptions required on the handled functions, which is the main downside of our work, and hopefully generalizing our results for additional widely common functions.

References

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. Approximating extent measures of points. *Journal of the ACM (JACM)*, 51(4):606–635, 2004.

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

Agarwal, P. K., Cormode, G., Huang, Z., Phillips, J. M., Wei,



(a) Bank Marketing dataset, $k = 100$

(b) Synthetic dataset, $k = 500$

(c) Wine dataset, $k = 1000$

(d) Bank Marketing dataset, $k = 10, R = 6$

(e) Wine dataset, $k = 500, R = 4$

Figure 3. Experimental results. **Fig. 3(a)-3(c):** The error of maximizing sum of sigmoids using coreset and uniform sampling. **Fig. 3(d)-3(e):** Negative test log-likelihood. Lower is better in all figures.

Z., and Yi, K. Mergeable summaries. *ACM Transactions on Database Systems (TODS)*, 38(4):1–28, 2013.

Anthony, M. and Bartlett, P. L. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Bachem, O., Lucic, M., and Krause, A. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*, 2017.

Bishop, C. M. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

Boutsidis, C., Drineas, P., and Magdon-Ismail, M. Near-optimal coresets for least-squares regression. *IEEE transactions on information theory*, 59(10):6880–6892, 2013.

Braverman, V., Feldman, D., and Lang, H. New frameworks for offline and streaming coreset constructions. *arXiv preprint arXiv:1612.00889*, 2016.

Braverman, V., Jiang, S., Krauthgamer, R., and Wu, X. Coresets for clustering with missing values. *Advances in Neural Information Processing Systems*, 34:17360–17372, 2021.

Code. Open source code for all the algorithms presented in this paper, 2022. Link for open-source code.

Cohen-Addad, V., Saulpic, D., and Schwiegelshohn, C. A new coreset framework for clustering. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 169–182, 2021.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4): 547–553, 2009.

Curtin, R. R., Im, S., Moseley, B., Pruhs, K., and Samadian, A. On coresets for regularized loss minimization. *arXiv preprint arXiv:1905.10845*, 2019.

Dasgupta, A., Drineas, P., Harb, B., Kumar, R., and Mahoney, M. W. Sampling algorithms and coresets for \ell_p regression. *SIAM Journal on Computing*, 38(5):2060–2078, 2009.

Elidan, G. Copula bayesian networks. In *Advances in neural information processing systems*, pp. 559–567, 2010.

Feldman, D. Core-sets: Updated survey. *Sampling Techniques for Supervised or Unsupervised Tasks*, pp. 23–44, 2020.

Feldman, D. and Langberg, M. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pp. 569–578. ACM, 2011.

Feldman, D. and Schulman, L. J. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1343–1354. Society for Industrial and Applied Mathematics, 2012.

Feldman, D., Monemizadeh, M., and Sohler, C. A ptas for k-means clustering based on weak coresets. In *Proceedings of the twenty-third annual symposium on Computational geometry*, pp. 11–18. ACM, 2007.

Feldman, D., Faulkner, M., and Krause, A. Scalable training of mixture models via coresets. In *Advances in neural information processing systems*, pp. 2142–2150, 2011.

Feldman, D., Schmidt, M., and Sohler, C. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1434–1453. Society for Industrial and Applied Mathematics, 2013a.

Feldman, D., Schmidt, M., and Sohler, C. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1434–1453. SIAM, 2013b.

Har-Peled, S. Coresets for discrete integration and clustering. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pp. 33–44. Springer, 2006.

Har-Peled, S., Roth, D., and Zimak, D. Maximum margin coresets for active and noise tolerant learning. In *IJCAI*, pp. 836–841, 2007.

Hellerstein, J. Parallel programming in the age of big data. Gigaom Blog, 2008.

Huggins, J., Campbell, T., and Broderick, T. Coresets for scalable bayesian logistic regression. In *Advances In Neural Information Processing Systems*, pp. 4080–4088, 2016.

Jubran, I., Tukan, M., Maalouf, A., and Feldman, D. Sets clustering. In *International Conference on Machine Learning*, pp. 4994–5005. PMLR, 2020.

Jubran, I., Sanches Shayda, E. E., Newman, I., and Feldman, D. Coresets for decision trees of signals. *Advances in Neural Information Processing Systems*, 34, 2021.

Kajino, H., Tsuboi, Y., and Kashima, H. A convex formulation for learning from crowds. *Transactions of the Japanese Society for Artificial Intelligence*, 27(3):133–142, 2012.

Kukačka, J., Golkov, V., and Cremers, D. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

Langberg, M. and Schulman, L. J. Universal ε approximators for integrals. *To appear in proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

Lu, H., Li, M.-J., He, T., Wang, S., Narayanan, V., and Chan, K. S. Robust coreset construction for distributed machine learning. *IEEE Journal on Selected Areas in Communications*, 38(10):2400–2417, 2020.

Lucic, M., Faulkner, M., Krause, A., and Feldman, D. Training gaussian mixture models at scale via coresets. *The Journal of Machine Learning Research*, 18(1):5885–5909, 2017.

Maalouf, A., Jubran, I., and Feldman, D. Fast and accurate least-mean-squares solvers. In *Advances in Neural Information Processing Systems*, pp. 8305–8316, 2019.

Maalouf, A., Jubran, I., and Feldman, D. Introduction to coresets: Approximated mean. *arXiv preprint arXiv:2111.03046*, 2021.

Mai, T., Rao, A. B., and Musco, C. Coresets for classification–simplified and strengthened. *arXiv preprint arXiv:2106.04254*, 2021.

Moro, S., Cortez, P., and Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Munteanu, A., Schwiegelshohn, C., Sohler, C., and Woodruff, D. On coresets for logistic regression. In *Advances in Neural Information Processing Systems*, pp. 6561–6570, 2018.

Phillips, J. M. Coresets and sketches. *arXiv preprint arXiv:1601.00617*, 2016.

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

Schölkopf, B., Smola, A. J., Bach, F., et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

Segaran, T. and Hammerbacher, J. *Beautiful Data: The Stories Behind Elegant Data Solutions*. O'Reilly Media, 2009.

Šíma, J. Training a single sigmoidal neuron is hard. *Neural computation*, 14(11):2709–2728, 2002.

Tsang, I. W., Kwok, J. T., and Cheung, P.-M. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6(Apr):363–392, 2005.

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

Tukan, M., Baykal, C., Feldman, D., and Rus, D. On coresets for support vector machines. *Theoretical Computer Science*, 890:171–191, 2021.

Vapnik, V. N. and Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Prob. Appl.*, 16:264–280, 1971.

Varadarajan, K. and Xiao, X. A near-linear algorithm for projective clustering integer points. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1329–1342. SIAM, 2012.

Wang, S. and Zhang, Z. Improving cur matrix decomposition and the nyström approximation via adaptive sampling. *The Journal of Machine Learning Research*, 14(1): 2729–2769, 2013.

Zheng, Y. and Phillips, J. M. Coresets for kernel regression. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 645–654, 2017.

A. No Coreset for General Monotonic Functions

In this section, we provide the full proofs behind the impossibility bound claims presented in Section 3.

The following lemma proves that if the sensitivity of every input point is 1 in a given query space, then there is no non-trivial coreset for the query space.

Lemma A.1 (Lower bound via Total sensitivity)**.** *Let (P, w, X, c) be a query space, and $\varepsilon \in (0, 1)$. If every $\boldsymbol{p} \in P$ has sensitivity $s_{P,w,X,c}(\boldsymbol{p}) = 1$, then for every ε-coreset (Q, u) we have $Q = P$.*

Proof. Let (Q, u) be a weighted set, where $Q \subset P$. It suffices to prove that (Q, u) is not an ε-coreset for P. Denote

$$u_{\max} \in \arg\max_{\boldsymbol{p}\in Q} u(\boldsymbol{p}), \text{ and } w_{\min} \in \arg\min_{\boldsymbol{p}\in P} w(\boldsymbol{p}).$$

Let $p \in P \setminus Q$. By the assumption $s_{P,w,X,c}(\boldsymbol{p}) \geq 1$, there is $\boldsymbol{x_p} \in X$ such that

$$\frac{w(\boldsymbol{p})\, c(\boldsymbol{p}, \boldsymbol{x_p})}{C(P, w, \boldsymbol{x_p})} = 1 > \frac{u_{\max}}{u_{\max}} - \frac{w_{\min}(1-\varepsilon)}{u_{\max}}.$$

Multiplication by $C(P, w, \boldsymbol{x}_p)$ yields

$$\begin{aligned} & w(\boldsymbol{p})\, c(\boldsymbol{p}, \boldsymbol{x_p}) > \\ & \frac{u_{\max} - w_{\min}(1-\varepsilon)}{u_{\max}} \cdot C(P, w, \boldsymbol{x_p}). \end{aligned} \tag{1}$$

We have that

$$\begin{aligned} & C(Q, u, \boldsymbol{x_p}) = \sum_{\boldsymbol{q}\in Q} u(\boldsymbol{q})\, c(\boldsymbol{q}, \boldsymbol{x_p}) \\ & = \sum_{\boldsymbol{q}\in Q} \frac{u(\boldsymbol{q})}{w(\boldsymbol{q})} w(\boldsymbol{q})\, c(\boldsymbol{p}, \boldsymbol{x_p}) \leq \frac{u_{\max}}{w_{\min}} \sum_{\boldsymbol{q}\in Q} w(\boldsymbol{q})\, c(\boldsymbol{q}, \boldsymbol{x_p}) \\ & \leq \frac{u_{\max}}{w_{\min}} \sum_{\boldsymbol{p'}\in P\setminus\{\boldsymbol{p}\}} w(\boldsymbol{p})\, c(\boldsymbol{p'}, \boldsymbol{x_p}) \qquad (2) \\ & = \frac{u_{\max}}{w_{\min}} \left(C(P, w, \boldsymbol{x_p}) - w(\boldsymbol{p})\, c(\boldsymbol{p}, \boldsymbol{x_p})\right) \\ & < \frac{u_{\max}}{w_{\min}} C(P, w, \boldsymbol{x_p}) \left(1 - \frac{u_{\max} - w_{\min}(1-\varepsilon)}{u_{\max}}\right) \qquad (3) \\ & = (1-\varepsilon)\, C(P, w, \boldsymbol{x_p}), \end{aligned}$$

where (2) is by the assumption $\boldsymbol{p} \in P \setminus Q$, and (3) is by (1). Hence Q cannot be used to approximate $C(P, w, \boldsymbol{x_p})$ and thus is not an ε-coreset for P. □

To prove there are query spaces (P, w, X, c) which admit no non-trivial coreset, we are left to formally prove there exists a set of points for which the sensitivity of every point is 1. Together with the lemma above, this will complete the proof.

Similarly to the idea behind the counter example in Section 3, the idea behind finding a set for which every point has sensitivity 1 is to find a set of points in which every point is linearly separable from the rest of the set. Such a set was shown to exist in (Huggins et al., 2016).

Lemma A.2 ((Huggins et al., 2016))**.** *There is a finite set of points $P \subseteq \mathbb{R}^d$ such that for every $\boldsymbol{p} \in P$ and $R > 0$ there is $\boldsymbol{y_p} \in \mathbb{R}^d$ of length $\|\boldsymbol{y_p}\| \leq R$ such that $\boldsymbol{y_p} \cdot \boldsymbol{p} = -R$, and for every $\boldsymbol{q} \in P \setminus \{\boldsymbol{p}\}$ we have $\boldsymbol{y_p} \cdot \boldsymbol{q} \geq R$.*

The following theorem stems from the combination of the above claims. Consider the query space (P, w, X, c), where P is the set of points from the lemma above, and $c(\boldsymbol{x}, \boldsymbol{p}) = f(\boldsymbol{x} \cdot \boldsymbol{p})$ for every $\boldsymbol{x}, \boldsymbol{p} \in \mathbb{R}^d$, where $f : \mathbb{R} \to (0, \infty)$ is a non-decreasing monotonic function. The theorem proves that, with respect to the query space (P, w, X, c), the sensitivity of every point in P is 1. We generalize a result from (Huggins et al., 2016) by considering weighted data and by letting the cost be any function upholding the conditions of Theorem A.3.

Theorem A.3 (Theorem 3.1)**.** *Let $f:\mathbb{R}\to(0,\infty)$ be a non-decreasing monotonic function that satisfies $\lim_{x\to\infty}\frac{f(-x)}{f(x)}=0$, and let $c\left(\boldsymbol{x},\boldsymbol{p}\right)=f\left(\boldsymbol{x}\cdot\boldsymbol{p}\right)$ for every $\boldsymbol{x},\boldsymbol{p}\in\mathbb{R}^d$. Let $\varepsilon\in(0,1)$, $n\geq 1$ be an integer, and $w:\mathbb{R}^d\to(0,\infty)$. There is a set $P\subset\mathbb{R}^d$ of $|P|=n$ points such that if (Q,u) is an ε-coreset of $\left(P,w,\mathbb{R}^d,c\right)$ then $Q=P$.*

Proof. Let $P\subseteq\mathbb{R}^d$ be the set that is defined in Lemma A.2, and let $\boldsymbol{p}\in P$, and $R>0$. By Lemma A.2, there is $\boldsymbol{y_p}\in\mathbb{R}^d$ such that $\boldsymbol{y_p}\cdot\boldsymbol{p}=-R$, and for every $\boldsymbol{q}\in P\setminus\{\boldsymbol{p}\}$ we have $\boldsymbol{-y_p}\cdot\boldsymbol{q}\leq -R$. By this pair of properties,

$$f\left(-\boldsymbol{y_p}\cdot\boldsymbol{p}\right)=f\left(R\right)\text{ and }f\left(\boldsymbol{-y_p}\cdot\boldsymbol{q}\right)\leq f\left(-R\right),$$

where in the last inequality we use the assumption that f is non-decreasing. By letting $\boldsymbol{x_p}=\boldsymbol{-y_p}$, we have

$$\frac{w(q)f\left(\boldsymbol{x_p}\cdot\boldsymbol{q}\right)}{w(p)f\left(\boldsymbol{x_p}\cdot\boldsymbol{p}\right)}=\frac{w(q)f\left(-\boldsymbol{y_p}\cdot\boldsymbol{q}\right)}{w(p)f\left(-\boldsymbol{y_p}\cdot\boldsymbol{p}\right)}\leq\frac{w(q)f\left(-R\right)}{w(p)f\left(R\right)}.$$

Therefore, by letting $w_{\max}\in\arg\max_{\boldsymbol{p}\in P}w\left(\boldsymbol{p}\right)$,

$$\begin{aligned}s_{P,w,\mathbb{R}^d,c}\left(\boldsymbol{p}\right)&\geq\frac{w\left(\boldsymbol{p}\right)f\left(\boldsymbol{x_p}u\cdot\boldsymbol{p}\right)}{\sum_{\boldsymbol{q}\in P}w\left(\boldsymbol{q}\right)f\left(\boldsymbol{x_p}\cdot\boldsymbol{q}\right)}\\&=\frac{w\left(\boldsymbol{p}\right)f\left(\boldsymbol{x_p}\cdot\boldsymbol{p}\right)}{w\left(\boldsymbol{p}\right)f\left(\boldsymbol{p}\cdot\boldsymbol{x_p}\right)+\sum_{\boldsymbol{q}\in P\setminus\{\boldsymbol{p}\}}w\left(\boldsymbol{q}\right)f\left(\boldsymbol{x_p}\cdot\boldsymbol{q}\right)}\\&=\frac{1}{1+\sum_{\boldsymbol{q}\in P\setminus\{\boldsymbol{p}\}}\frac{w(\boldsymbol{q})f(\boldsymbol{x_p}\cdot\boldsymbol{q})}{w(\boldsymbol{p})f(\boldsymbol{x_p}\cdot\boldsymbol{p})}}\geq\frac{1}{1+\sum_{q\in P\setminus\{\boldsymbol{p}\}}\frac{w(\boldsymbol{q})f(-R)}{w(\boldsymbol{p})f(R)}}\\&\geq\frac{1}{1+(n-1)\frac{w_{\max}f(-R)}{w(\boldsymbol{p})f(R)}}.\end{aligned}$$

We also have

$$\lim_{R\to\infty}\frac{w_{\max}f\left(-R\right)}{w\left(\boldsymbol{p}\right)f\left(R\right)}=\frac{w_{\max}}{w(\boldsymbol{p})}\lim_{R\to\infty}\frac{f\left(-R\right)}{f\left(R\right)}=0,$$

where the last derivation holds by the assumption on f. Thus we obtain

$$s_{P,w,\mathbb{R}^d,c}\left(\boldsymbol{p}\right)=\sup_{R>0}\frac{1}{1+(n-1)\frac{w_{\max}f(-R)}{w(\boldsymbol{p})f(R)}}=1.$$

Theorem A.3 then follows from the last equality and Lemma A.1. □

B. $\mathcal{L}_\infty$-Coresets

Lemma B.1 (Lemma 4.2)**.** *Let $P\subset\mathbb{R}^d$ be a finite set, $M,k>0$ be constants, $f:\mathbb{R}\to(0,M]$ be non-decreasing function and $g:[0,\infty)\to[0,\infty)$ be a function. For every $\boldsymbol{x}\in\mathbb{R}^d$ and $\boldsymbol{p}\in P$ define $c_k\left(\boldsymbol{p},\boldsymbol{x}\right)=f\left(\boldsymbol{p}\cdot\boldsymbol{x}\right)+\frac{g(\|\boldsymbol{x}\|)}{k}$. Put $\boldsymbol{p}\in P$. Suppose there is $b_{\boldsymbol{p}}>0$ such that for every $z>0$*

$$f\left(\|\boldsymbol{p}\|z\right)+\frac{g\left(z\right)}{k}\leq b_{\boldsymbol{p}}\left(f\left(-\|\boldsymbol{p}\|z\right)+\frac{g\left(z\right)}{k}\right).\tag{4}$$

Then $\{\boldsymbol{p}\}$ is an $\varepsilon-\mathcal{L}_\infty$ coreset with $\varepsilon=\frac{M}{f(0)}\left(b_{\boldsymbol{p}}+1\right)-1$, i.e., for every $\boldsymbol{x}\in\mathbb{R}^d$

$$\max_{\boldsymbol{p'}\in P}c_k\left(\boldsymbol{p'},\boldsymbol{x}\right)\leq\frac{M}{f\left(0\right)}\left(b_{\boldsymbol{p}}+1\right)c_k\left(\boldsymbol{p},\boldsymbol{x}\right).$$

Proof. Let $\boldsymbol{x}\in\mathbb{R}^d$ and $q\in P$ such that $\boldsymbol{x}\cdot\boldsymbol{q}>0$. We have, by the monotonic properties of f,

$$f\left(0\right)\leq f\left(\boldsymbol{x}\cdot\boldsymbol{q}\right).\tag{5}$$

Hence,

$$\max_{\boldsymbol{p}'\in P} f\left(\boldsymbol{x}\cdot\boldsymbol{p}'\right) \le M = \frac{M}{f\left(0\right)} f\left(0\right) \le \frac{M}{f\left(0\right)} f\left(\boldsymbol{x}\cdot\boldsymbol{q}\right), \tag{6}$$

where the first inequality is since f is bounded by M, and the last inequality is by (5). By adding $\frac{g(\|\boldsymbol{x}\|)}{k}$ to both sides of (6) and since $1 \le \frac{M}{f(0)}$ we obtain,

$$\begin{aligned} &\max_{p'\in P} c_k(p', x) = \max_{\boldsymbol{p}'\in P} f\left(\boldsymbol{x}\cdot\boldsymbol{p}'\right) + \frac{g\left(\|\boldsymbol{x}\|\right)}{k} \\ &\le \frac{M}{f\left(0\right)} f\left(\boldsymbol{x}\cdot\boldsymbol{q}\right) + \frac{g\left(\|\boldsymbol{x}\|\right)}{k} \\ &\le \frac{M}{f\left(0\right)} \left(f\left(\boldsymbol{x}\cdot\boldsymbol{q}\right) + \frac{g\left(\|\boldsymbol{x}\|\right)}{k} \right). \end{aligned} \tag{7}$$

The rest of the proof follows by case analysis on the sign of $\boldsymbol{x}\cdot\boldsymbol{p}$, i.e. (i) $\boldsymbol{x}\cdot\boldsymbol{p} \ge 0$ and (ii) $\boldsymbol{x}\cdot\boldsymbol{p} < 0$.

Case (i): $\boldsymbol{x}\cdot\boldsymbol{p} \ge 0$. Substituting $q = p$ in (7) yields

$$\begin{aligned} &\max_{p'\in P} c_k(p', x) \le \frac{M}{f\left(0\right)} \left(f\left(\boldsymbol{x}\cdot\boldsymbol{p}\right) + \frac{g\left(\|\boldsymbol{x}\|\right)}{k} \right) \\ &= \frac{M}{f\left(0\right)} c_k(p, x) \le \frac{M}{f\left(0\right)} (b_p + 1) c_k(p, x), \end{aligned} \tag{8}$$

where the last inequality follows by the assumption $b_p > 0$. **Case (ii)**: $\boldsymbol{x}\cdot\boldsymbol{p} < 0$. In this case $\boldsymbol{x}\cdot(-\boldsymbol{p}) > 0$. Substituting $q = -p$ in (7) yields

$$\begin{align} &\max_{p'\in P} c_k(p', x) \le \frac{M}{f\left(0\right)} \left(f\left(\boldsymbol{x}\cdot(-\boldsymbol{p})\right) + \frac{g\left(\|\boldsymbol{x}\|\right)}{k} \right) \tag{9} \\ &\le \frac{M}{f\left(0\right)} \left(f\left(\|\boldsymbol{x}\| \, \|\boldsymbol{p}\|\right) + \frac{g\left(\|\boldsymbol{x}\|\right)}{k} \right) \tag{10} \\ &\le \frac{M}{f\left(0\right)} b_{\boldsymbol{p}} \left(f\left(-\|\boldsymbol{x}\| \, \|\boldsymbol{p}\|\right) + \frac{g\left(\|\boldsymbol{x}\|\right)}{k} \right) \tag{11} \\ &\le \frac{M}{f\left(0\right)} b_{\boldsymbol{p}} \left(f\left(\boldsymbol{x}\cdot\boldsymbol{p}\right) + \frac{g\left(\|\boldsymbol{x}\|\right)}{k} \right), \tag{12} \\ &= \frac{M}{f(0)} b_p c_k(p, x) \le \frac{M}{f(0)} (b_p + 1) c_k(p, x), \tag{13} \end{align}$$

where (10) and (12) are by the Cauchy-Schwartz inequality and the monotonicity of f, and (11) follows by substituting $z = \|x\|$ in the main assumption of the lemma. □

B.1. From $\varepsilon - \mathcal{L}_\infty$ coresets to ε-coresets

In what follows is the full proof for Lemma 4.3. We prove that the algorithm described in Section 4.2, which constructs a series of $\mathcal{L}_\infty$ coresets, can indeed give an upper bound on the sensitivity of every input element as well as a near logarithmic upper bound on the total sensitivity.

Lemma B.2 (Lemma 4.3)**.** *Let $c : \mathbb{R}^d \times \mathbb{R}^d \to (0, \infty)$. Suppose that for some $\varepsilon \in (0, 1)$ there is a non-decreasing function $\Delta_\varepsilon(n)$ so that for any $P' \subseteq \mathbb{R}^d$ of size n there is an $\varepsilon - \mathcal{L}_\infty$ coreset of size at most $\Delta_\varepsilon(n)$ for $(P', \mathbf{1}, \mathbb{R}^d, c)$. Then, for any $P \subseteq \mathbb{R}^d$ of size n we can compute an upper bound $s(p)$ on the sensitivity $s_{P,\mathbf{1},\mathbb{R}^d,c}(p)$ for each $p \in P$, so that $\sum_{p\in P} s_{P,\mathbf{1},\mathbb{R}^d,c}(p) \le (1+\varepsilon)\Delta_\varepsilon(n)\ln n$.*

Proof. The proof is constructive. We build a sequence of subsets $P_1 \supseteq P_2 \supseteq \cdots \supseteq P_m$, where $P = P_1$, $m \le n$, and $|P_m| \le \Delta_\varepsilon(n)$. We construct the sequence as follows. If $|P_i| \le \Delta_\varepsilon(n)$ the sequence stops. Otherwise, we compute an $\mathcal{L}_\infty$ ε-coreset C_i for $(P_i, \mathbf{1}, \mathbb{R}^d, c)$ of size $|C_i| \le \Delta_\varepsilon(n)$. We now define $P_{i+1} = P_i \setminus C_i$.

Put $i \in [m]$. We now upper bound the sensitivity $s_{P,\mathbf{1},\mathbb{R}^d,c}(q)$ for every $q \in C_i$ by $\frac{1+\varepsilon}{i}$ and upper bound the total sensitivity $\sum_{p \in P} s(p) \leq (1+\varepsilon)\Delta_\varepsilon(n)\ln n$.

Put $x \in \mathbb{R}$ and $q' \in C_i$, and consider $1 \leq j \leq i$. Let $q_j \in C_j$ be the points in the $\varepsilon - \mathcal{L}_\infty$ coreset C_j such that $c(q_j, x) = \max_{q \in C_j} c(q, x)$. We now have that

$$c(q', x) \leq \max_{p \in P_j} c(p, x) \leq (1+\varepsilon) \max_{q \in C_j} c(q, x) = (1+\varepsilon) c(q_j, x) \tag{14}$$

where the first derivations holds since, by construction, $q' \in P_j$. The second derivation is by the definition of an $\varepsilon - \mathcal{L}_\infty$ coreset. We thus obtain that

$$\frac{c(q', x)}{\sum_{p \in P} c(p, x)} \leq \frac{c(q', x)}{\sum_{\ell=1}^{i} c(q_\ell, x)} \leq \frac{1+\varepsilon}{i}, \tag{15}$$

where the second derivation holds since $\{q_j \mid 1 \leq j \leq i\} \subseteq P$, and the last derivation is by (14). Since (15) holds for any $x \in \mathbb{R}^d$, we obtain that the sensitivity of q' is upper bounded by

$$s_{P,\mathbf{1},\mathbb{R}^d,c}(q') = \sup_{\boldsymbol{x} \in \mathbb{R}^d} \frac{c(q', x)}{\sum_{p \in P} c(p, x)} \leq \frac{1+\varepsilon}{i}.$$

Hence, for every $q' \in C_i$ we have that $s_{P,\mathbf{1},\mathbb{R}^d,c}(q') \leq \frac{1+\varepsilon}{i}$. Now, the total sensitivity can be bounded by

$$\sum_{p \in P} s_{P,\mathbf{1},\mathbb{R}^d,c}(p) = \sum_{i=1}^{m} \frac{(1+\varepsilon)|C_i|}{i} \leq \Delta_\varepsilon(n) \sum_{i=1}^{m} \frac{1+\varepsilon}{i} \leq (1+\varepsilon)\Delta_\varepsilon(n) \ln n.$$

□

B.2. Coreset sufficient condition

In what follows we give the full proof for Theorem 4.4. The proof is constructive in the sense that it gives an upper bound for the sensitivity of every input point and upper bounds the total sensitivity by a term which is near logarithmic in the input size.

Theorem B.3 (Theorem 4.4)**.** *Let $M, k > 0$ be constants, $P \subseteq \mathbb{R}^d$ be a set of points, $f : \mathbb{R} \to (0, M]$ be a monotonic non-decreasing function, and $c_k(\boldsymbol{p'}, \boldsymbol{x}) = f(\boldsymbol{p'} \cdot \boldsymbol{x}) + \frac{g(\|x\|)}{k}$ for every $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{p'} \in P$. Suppose there is $b : P \to (0, \infty)$ such that for every $\boldsymbol{p} \in P$ and every $z > 0$*

$$f(\|\boldsymbol{p}\| z) + \frac{g(z)}{k} \leq b(\boldsymbol{p}) \left(f(-\|\boldsymbol{p}\| z) + \frac{g(z)}{k} \right). \tag{16}$$

Let $b_{\max} \in \arg\max_{\boldsymbol{p} \in P} b(\boldsymbol{p})$, $t = (1 + \frac{M}{f(0)} b_{\max}) \ln n$, and $\varepsilon, \delta \in (0, 1)$. Lastly, let d_{VC} be the VC-dimension of $(P, \mathbf{1}, \mathbb{R}^d, c_k)$. Then, there is a weighted set (Q, u), where $Q \subseteq P$ and

$$|Q| \in O\left(\frac{t}{\varepsilon^2} \left(d_{VC} \log t + \log \frac{1}{\delta} \right) \right),$$

such that with probability at least $1 - \delta$, (Q, u) is an ε-coreset for the query space $(P, \mathbf{1}, \mathbb{R}^d, c_k)$.

Proof. For $\boldsymbol{p} \in P$ we have that

$$\max_{\boldsymbol{p'} \in P} c_k(\boldsymbol{p'}, \boldsymbol{x}) \leq \frac{M}{f(0)} (b_{\boldsymbol{p}} + 1) c_k(\boldsymbol{p}, \boldsymbol{x}) \leq \tag{17}$$

$$\frac{M}{f(0)} (b_{\max} + 1) c_k(\boldsymbol{p}, \boldsymbol{x}). \tag{18}$$

Where (17) is by substituting in Lemma 4.2 and (18) holds since for every $\boldsymbol{p} \in P$, $b(\boldsymbol{p}) \leq b_{\max}$. Let $\varepsilon(\boldsymbol{p}) := \left[\left(\frac{M}{f(0)} (b(\boldsymbol{p}) + 1) \right) - 1 \right]$ for every $\{p\} \in P$ and let $\varepsilon = \left(\frac{M}{f(0)} (b_{\max} + 1) \right) - 1$. Thus, for every $\boldsymbol{p} \in P$, we have that $\{\boldsymbol{p}\}$ is an $\varepsilon(\boldsymbol{p})$-$\mathcal{L}_\infty$ coreset which is a ε-$\mathcal{L}_\infty$ coreset.

In Lemma 4.3, a sequence of distinct $\mathcal{L}_\infty$ coresets that cover the entire set P $C_1 \cup \dots \cup C_m \subseteq P$ are constructed. For every $\boldsymbol{p} \in P$, let $i(\boldsymbol{p})$ be the index of the $\mathcal{L}_\infty$ coreset $C_{i(\boldsymbol{p})}$ such that $\boldsymbol{p} \in C_{i(\boldsymbol{p})}$. Plugging ε and $\Delta_\varepsilon(n) = 1$ in Lemma 4.3 and its proof yields that we can upper bound the sensitivity of every $\boldsymbol{p} \in P$ by

$$s_{P,\mathbf{1},\mathbb{R}^d,c_k}(\boldsymbol{p}) \leq \frac{1+\varepsilon(\boldsymbol{p})}{i(\boldsymbol{p})} \leq \frac{1+\varepsilon}{i(\boldsymbol{p})} = \frac{\left(\frac{M}{f(0)}\left(b_{\max}+1\right)\right)}{i(\boldsymbol{p})},$$

where $i(\boldsymbol{p})$ is the index of $\boldsymbol{p}$ when sorting the points in P by their norm. Furthermore, the total sensitivity is bounded by

$$t(P,\mathbf{1},\mathbb{R}^d,c_k) = \sum_{\boldsymbol{p}\in P} \frac{\left(\frac{M}{f(0)}\left(b_{\max}+1\right)\right)}{i(\boldsymbol{p})} \in O\left((1+\frac{M}{f(0)}b_{\max})\ln n\right).$$

Observe that sensitivity of $\boldsymbol{p} \in P$ depends on $\varepsilon(\boldsymbol{p}) = \left[\left(\frac{M}{f(0)}\left(b(\boldsymbol{p})+1\right)\right) - 1\right]$ divided by the index $i(\boldsymbol{p})$. Hence, empirically, to obtain smaller total sensitivity, we would prefer to reorder P such that points $\boldsymbol{p}$ with larger value of $b(\boldsymbol{p})$ are divided by larger values $i(\boldsymbol{p})$. Therefore, we can simply sort the points of P according to the values of the function b, from smallest to largest. Thus, points $\boldsymbol{p}$ with larger value of $b(\boldsymbol{p})$ are given larger index $i(\boldsymbol{p})$.

Theorem 4.4 now immediately follows from Theorem 2.3. □

C. Main Proofs

In this section, we first prove a series of technical claims. We then utilize those claims to prove the main results of this work.

C.1. Technical Claims

Lemma C.1. *Let $f : \mathbb{R} \to (0,\infty)$ be a monotonic increasing function such that $f(0) > 0$. Let $c, k > 0$. There is exactly one number $x_{kc} > 0$ that simultaneously satisfies the following claims.*

(i) $f\left(-\sqrt{ck}x_{kc}\right) = x_{kc}^2$.

(ii) *For every $x > 0$, if $f\left(-\sqrt{ck}x\right) > x^2$ then $x < x_{kc}$.*

(iii) *For every $x > 0$, if $f\left(-\sqrt{ck}x\right) < x^2$ then $x > x_{kc}$.*

(iv) *There is $k_0 > 0$ such that for every $k' \geq k_0$*

$$\frac{1}{x_{kc}} \leq \sqrt{ck'}.$$

Proof. Let $g(x) = x^2$. Define

$$h_{kc}(x) = f(-\sqrt{ck}x) - g(x). \tag{19}$$

(i): It holds that

$$h_{kc}(0) = f(0) \tag{20}$$

and

$$h_{kc}\left(\sqrt{f(0)+1}\right) < 0, \tag{21}$$

where (21) holds since $f\left(-\sqrt{ck}x\right) \leq f(0)$ for every $x > 0$, and $g\left(\sqrt{f(0)+1}\right) = f(0)+1$. From (20) and (21) we have that $0 \in \left[h_{kc}\left(\sqrt{f\left(0\right)+1}\right), h_{kc}(0)\right]$. Using the Intermediate Value Theorem (Theorem E.1) we have that there is $x_1 \in \left(0, \sqrt{f(0)+1}\right)$ such that

$$h_{kc}(x_1) = 0. \tag{22}$$

We prove that x_1 is unique. By contradiction. Assume that there is $x_2 \neq x_1$ such that

$$h_{kc}(x_1) = h_{kc}(x_2) = 0. \tag{23}$$

Wlog assume that $x_1 < x_2$. By The Mean Value Theorem (Theorem E.2), there is $r \in (x_1, x_2)$ such that

$$\begin{align} h'_{kc}(r) &= \frac{h_{kc}(x_2) - h_{kc}(x_1)}{x_2 - x_1} \tag{24} \\ &= 0, \tag{25} \end{align}$$

where (25) is by (23). The derivative of h_{kc} is

$$\begin{align} h'_{kc}(x) &= \left(f\left(-\sqrt{ck}x\right) - g(x)\right)' \tag{26} \\ &= -\sqrt{ck}f'\left(-\sqrt{ck}x\right) - g'(x) < 0, \tag{27} \end{align}$$

where (26) is by (19) and (27) is since f is monotonic increasing and thus $f'(x) > 0$ for every $x \in \mathbb{R}$ and $x, k, c > 0$. (27) is a contradiction to (25). Thus the Assumption (23) is false and x_1 is unique.

By (19) and (22)

$$f\left(-\sqrt{ck}x_1\right) = g(x_1). \tag{28}$$

By letting $x_{kc} = x_1$ and recalling that $g(x) = x^2$ we obtain

$$f\left(-\sqrt{ck}x_{kc}\right) = x_{kc}^2.$$

(ii): Let $x > 0$ such that $f\left(-\sqrt{ck}x\right) > x^2$. Plugging this and the definition $g(x) = x^2$ in (19) yields

$$h_{kc}(x) > 0. \tag{29}$$

We already proved that $h'_{kc}(x) < 0$ always. By the Inverse of Strictly Monotone Function Theorem (Theorem E.3) we have that the inverse h_{kc}^{-1} of h_{kc} is a strictly monotone decreasing function. Applying h_{kc}^{-1} on both sides of (29) gives

$$x < x_{kc}.$$

(iii): Let $x > 0$ such that $f\left(-\sqrt{ck}x\right) < x^2$. By this and by the definition of g and (19) we have

$$h_{kc}(x) < 0. \tag{30}$$

We already proved that $h'_{kc}(x) < 0$ always. By the Inverse of Strictly Monotone Function Theorem (Theorem E.3) we have that h_{kc} has a strictly monotone decreasing inverse function h_{kc}^{-1}. Applying h_{kc}^{-1} on both sides of (30) gives

$$x > x_{kc}.$$

(iv): We need to prove that there is k_0 such that for every $k' > k_0$ we have

$$x_{kc} \geq \frac{1}{\sqrt{ck'}} \tag{31}$$

By contradiction, assume that for every $k' > 0$,

$$x_{kc} < \frac{1}{\sqrt{ck'}}. \tag{32}$$

Since f is increasing and by (32) $-c\sqrt{k}x_{kc} > -1$ we have that

$$f\left(-\sqrt{ck}x_{kc}\right) > f(-1). \tag{33}$$

where (33) holds since f is increasing and by (32) $-\sqrt{ck}x_{kc} > -1$. Since $\lim_{k\to\infty}\frac{1}{ck} = 0$, there is $k_0 > 0$ such that for every $k > k_0$

$$\begin{aligned} f(-1) >& \frac{1}{ck} \\ >& x_{kc}^2, \end{aligned} \tag{34}$$

where (34) is by (32). Plugging (34) in (33) yields

$$f\left(-c\sqrt{k}x_{kc}\right) > x_{kc}^2. \tag{35}$$

In contradictions to **(i)**. Thus

$$x_{kc} \geq \frac{1}{\sqrt{ck}} \tag{36}$$

□

Lemma C.2. *Let f be either the sigmoid or the logistic regression function and let $x_{1,1} > 0$ which is obtained by applying Lemma C.1(i) with f and $k = c = 1$. Then, For every $x \geq 0$*

$$\frac{f(x) + x^2}{f(-x) + x^2} \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\}.$$

Proof. Let $x \geq 0$. Substituting $k = c = 1$ in Lemma C.1**(i)** yields that $f(-x_{1,1}) = x_{1,1}^2$. We show that $\frac{f(x)+x^2}{f(-x)+x^2} \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\}$ via the following case analysis. **(i)** $f(x) \geq x^2$ and $f(-x) \geq x^2$, **(ii)** $f(x) \geq x^2$ and $f(-x) < x^2$, **(iii)** $f(x) < x^2$and $f(-x) \geq x^2$, and **(iv)** $f(x) < x^2$and $f(-x) < x^2$.

Case (i): $f(x) \geq x^2$ and $f(-x) \geq x^2$. Since $f(-x) \geq x^2$, by substituting $k = c = 1$ in Lemma C.1**(ii)**, we have that $x \leq x_{1,1}$. Hence

$$\begin{aligned} f(-x) + x^2 \geq& f(-x) && (37)\\ \geq& f(-x_{1,1}) && (38)\\ =& x_1^2, && (39) \end{aligned}$$

where (37) is since $x^2 > 0$, (38) is since f is increasing and $x \leq x_{1,1}$, and (39) is by definition of $x_{1,1}$. By adding $f(x)$ to both sides of the assumption $f(x) \geq x^2$ of Case (i) we obtain

$$2f(x) \geq f(x) + x^2. \tag{40}$$

By (40) and (39) we obtain

$$\frac{f(x) + x^2}{f(-x) + x^2} \leq \frac{2f(x)}{x_{1,1}^2} \leq \frac{2}{x_{1,1}^2} \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\}. \tag{41}$$

where the second inequality holds since $f(x) \leq 1$ due to f being the sigmoid function.

Case (ii): $f(x) \geq x^2$ and $f(-x) < x^2$. Since $f(-x) < x^2$, substituting $k = c = 1$ in Lemma C.1**(iii)**, there is $x_{1,1}$ such that

$$\begin{aligned} f(-x) + x^2 \geq& x^2 && (42)\\ >& x_{1,1}^2, && (43) \end{aligned}$$

where (42) is since f is a positive function and (43) is since $x > x_{1,1}$. By adding $f(x)$ to both sides of the assumption $f(x) \geq x^2$ of Case (ii) we have that

$$f(x) + x^2 \leq 2f(x). \tag{44}$$

By (44)) and (43) we obtain

$$\frac{f(x)+x^2}{f(-x)+x^2} \leq \frac{2f(x)}{x_{1,1}^2} \leq \frac{2}{x_{1,1}^2} \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\}. \tag{45}$$

where the second inequality holds since $f(x) \leq 1$ due to f being either the sigmoid or the logistic regression function.

Case (iii): $f(x) < x^2$and $f(-x) \geq x^2$. By adding x^2 to both sides of the assumption $f(x) < x^2$ of Case (iii) we have that

$$f(x) + x^2 \leq 2x^2. \tag{46}$$

Furthermore, since $f(-x) > 0$ we have that

$$f(-x) + x^2 \geq x^2. \tag{47}$$

Combining (46) and (47) we obtain

$$\frac{f(x)+x^2}{f(-x)+x^2} \leq \frac{2x^2}{x^2} \leq 2 \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\}. \tag{48}$$

Case (iv): $f(x) < x^2$ and $f(-x) < x^2$. By adding x^2 to both sides of the assumption $f(x) < x^2$ of Case (iv) we have that

$$f(x) + x^2 \leq 2x^2. \tag{49}$$

Furthermore, since $f(-x) > 0$ we have that

$$f(-x) + x^2 \geq x^2. \tag{50}$$

Combining (49) and (50) we obtain

$$\frac{f(x)+x^2}{f(-x)+x^2} \leq \frac{2x^2}{x^2} \leq 2 \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\}. \tag{51}$$

Combining the results of the case analysis: (41), (45), (48),and (51) we have that

$$\frac{f(x)+x^2}{f(-x)+x^2} \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\}. \tag{52}$$

□

Lemma C.3. *Let f be the sigmoid function, let $x_{1,1}$ be as in Lemma C.2, and let $c > 0$. Assume that there is $D > 1$ such that $\frac{f(cy)}{f\left(\frac{y}{\sqrt{k}}\right)} \leq D$ for every $y \geq 0$ and $k > 0$. Then, there is $k_0 > 0$ such that for every $k \geq k_0$ and for every $x \geq 0$,*

$$\frac{f(\sqrt{c}x) + \frac{x^2}{k}}{f(-\sqrt{c}x) + \frac{x^2}{k}} \leq 3D \max\left\{2, \frac{2}{x_{1,1}^2}\right\} ck.$$

Proof. Let $x \geq 0$ and $k, c > 0$. We have that

$$\begin{aligned} f(cx) + \frac{x^2}{k} \leq & Df\left(\frac{x}{\sqrt{k}}\right) + \frac{x^2}{k} && (53)\\ \leq & D\max\left\{2, \frac{2}{x_{1,1}^2}\right\}\left(f\left(-\frac{x}{\sqrt{k}}\right) + \frac{x^2}{k}\right), && (54)\end{aligned}$$

where (53) holds since$\frac{f(cy)}{f\left(\frac{y}{\sqrt{k}}\right)} < D$ for every $y \geq 0$ and (54) holds since $\frac{x^2}{k} \leq D\frac{x^2}{k}$, and since, by Lemma C.2, for every positive z we have that

$$\frac{f(z)+z^2}{f(-z)+z^2} \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\}.$$

Dividing (54) by $f\left(-\sqrt{c}x\right)+\frac{x^2}{k}$ yields

$$\frac{f\left(cx\right)+\frac{x^2}{k}}{f\left(-\sqrt{c}x\right)+\frac{x^2}{k}} \leq D\max\left\{2,\frac{2}{x_{1,1}^2}\right\}\left(\frac{f\left(-\frac{x}{\sqrt{k}}\right)+\frac{x^2}{k}}{f\left(-\sqrt{c}x\right)+\frac{x^2}{k}}\right). \tag{55}$$

We now proceed to bound $R_{ck}=\frac{f\left(-\frac{x}{\sqrt{k}}\right)+\frac{x^2}{k}}{f\left(-\sqrt{c}x\right)+\frac{x^2}{k}}$. By denoting $z=\frac{x}{\sqrt{k}}$ we have that

$$R_{ck}=\frac{f\left(-z\right)+z^2}{f\left(-\sqrt{ck}z\right)+z^2}. \tag{56}$$

We now compute an upper bound for R_{ck} using the following case analysis: **(i)** $f\left(-z\right)\geq z^2$ and $f\left(-\sqrt{ck}z\right)\geq z^2$, **(ii)** $f\left(-z\right)<z^2$ and $f\left(-\sqrt{ck}z\right)<z^2$, **(iii)**, and **(iv)** $f\left(-z\right)<z^2$ and $f\left(-\sqrt{ck}z\right)\geq z^2$. Let $z_{ck}>0$ be such that $f\left(-\sqrt{ck}z_{ck}\right)=z_{ck}^2$ as given by Lemma C.1**(i)**. There are four cases

Case (i): $f\left(-z\right)\geq z^2$ and $f\left(-\sqrt{ck}z\right)\geq z^2$. Since $f\left(-\sqrt{ck}z\right)\geq z^2$, by Lemma C.1**(iii)** we have that $z\leq z_{ck}$. Thus

$$f\left(-\sqrt{ck}z\right) \geq f\left(-\sqrt{ck}z_{ck}\right) \tag{57}$$

$$= z_{ck}^2, \tag{58}$$

where (57) holds since f is monotonic and $z\leq z_{ck}$, and (58) is from the definition of z_{ck}. Furthermore, by adding $f\left(-z\right)$ to both sides of the assumption $f\left(-z\right)\geq z^2$, we have that

$$f\left(-z\right)+z^2\leq 2f\left(-z\right). \tag{59}$$

Substituting (59) and (58) in (56) yields

$$R_{ck}=\frac{f\left(-z\right)+z^2}{f\left(-\sqrt{ck}z\right)+z^2}\leq\frac{2f\left(-z\right)}{z_{ck}^2}\leq\frac{1}{z_{ck}^2}, \tag{60}$$

where the last inequality, is since $f(-z)\leq 1/2$ for every $z\geq 0$.

Case (ii): $f\left(-z\right)<z^2$ and $f\left(-\sqrt{ck}z\right)<z^2$. By adding z^2 to both sides of the assumption $f\left(-z\right)<z^2$, we have that

$$f\left(-z\right)+z^2\leq 2z^2. \tag{61}$$

Furthermore, since $f\left(-\sqrt{ck}z\right)>0$ we have that

$$f\left(-\sqrt{ck}z\right)+z^2\geq z^2. \tag{62}$$

Combining (61) and (62) yields

$$R_{ck}=\frac{f\left(-z\right)+z^2}{f\left(-\sqrt{ck}z\right)+z^2}\leq\frac{2z^2}{z^2}=2. \tag{63}$$

Case (iii): $f\left(-z\right)\geq z^2$ and $f\left(-\sqrt{ck}z\right)<z^2$. Since $f\left(-\sqrt{ck}z\right)<z^2$, by Lemma C.1 we have that $z>z_{ck}$. Thus

$$f\left(-\sqrt{ck}z\right)+z^2\geq z^2\geq z_{ck}^2. \tag{64}$$

By adding $f\left(-z\right)$ to both sides of the assumption $f\left(-z\right) \geq z^2$, we have that

$$2f\left(-z\right) \geq f\left(-z\right) + z^2. \tag{65}$$

Substituting (64) and (65) in (56) yields

$$R_{ck} = \frac{f\left(-z\right) + z^2}{f\left(-\sqrt{ck}z\right) + z^2} \leq \frac{2f\left(-z\right)}{z_{ck}^2} \leq \frac{1}{z_{ck}^2}. \tag{66}$$

Case (iv): $f\left(-z\right) < z^2$ and $f\left(-\sqrt{ck}z\right) \geq z^2$. By adding z^2 to both sides of the assumption $f\left(-z\right) < z^2$, we have that

$$f\left(-z\right) + z^2 \leq 2z^2. \tag{67}$$

Since $f\left(-\sqrt{ck}z\right) > 0$ we have that

$$f\left(-\sqrt{ck}z\right) + z^2 > z^2. \tag{68}$$

Plugging (67) and (68) in (56) yields

$$R_{ck} = \frac{f\left(-z\right) + z^2}{f\left(-\sqrt{ck}z\right) + z^2} \leq \frac{2z^2}{z^2} = 2. \tag{69}$$

Combining the results of the case analysis: (60), (63), (66),and (69) we have that

$$R_{ck} \leq 2 + \frac{1}{z_{ck}^2}. \tag{70}$$

Furthermore, there exists $k_0 > 0$ such that for every $k \geq k_0$,

$$\frac{1}{z_{ck}^2} \leq ck. \tag{71}$$

Substituting (71) in (70) yields

$$R_{ck} \leq 2 + ck, \tag{72}$$

by (55) we have

$$\frac{f\left(cx\right) + \frac{x^2}{k}}{f\left(-\sqrt{c}x\right) + \frac{x^2}{k}} \leq D \max\left\{2, \frac{2}{x_{1,1}^2}\right\} R_{ck}.$$

Substituting (72) in the last term gives

$$\frac{f\left(cx\right) + \frac{x^2}{k}}{f\left(-\sqrt{c}x\right) + \frac{x^2}{k}} \leq D \max\left\{2, \frac{2}{x_{1,1}^2}\right\} \left(2 + ck\right).$$

It holds that for every $k \geq \frac{1}{c}$ we have $2 \leq 2ck$ plugging this in the above term yields

$$\frac{f\left(cx\right) + \frac{x^2}{k}}{f\left(-\sqrt{c}x\right) + \frac{x^2}{k}} \leq 3D \max\left\{2, \frac{2}{x_{1,1}^2}\right\} ck.$$

□

The following lemma is similar to Lemma C.3 above, but for the logistic regression function. The proof is similar to the proof of Lemma C.3.

Lemma C.4. *Let f be the logistic regression function, let $x_{1,1}$ be as in Lemma C.2, and let $c, R > 0$. Assume that there is $D > 1$ such that $\frac{f(cy)}{f\left(\frac{y}{\sqrt{k}}\right)} \leq D$ for every $y \geq 0$ and $k > 0$. Then, there is $k_0 > 0$ such that for every $k \geq k_0$ and for every $0 \leq x \leq R$,*

$$\frac{f\left(cx\right) + \frac{x^2}{k}}{f\left(-cx\right) + \frac{x^2}{k}} \leq 3RD \max\left\{2, \frac{2}{x_{1,1}^2}\right\} ck.$$

Lemma C.5. *Let $f(x) = \frac{1}{1+e^{-x}}$ for every $x \in \mathbb{R}$ and let $c > 0$. Then, there is $k_0 > 0$ such that for every $k \geq k_0$ and for every $x \geq 0$*

$$\frac{f\left(cx\right) + \frac{x^2}{k}}{f\left(-cx\right) + \frac{x^2}{k}} \leq 66ck$$

Proof. It holds that $f\left(0\right) > 0$. Applying Lemma C.1 with $k = c = 1$ yields $x_{1,1}$ such that $f\left(-x_{1,1}\right) = x_{1,1}^2$. We now bound $x_{1,1}$. Calculation shows that

$$f\left(-\sqrt{\ln\left(1.2\right)}\right) > \left(\sqrt{\ln\left(1.2\right)}\right)^2.$$

Plugging $x = \sqrt{\ln\left(1.2\right)}, k = 1, c = 1$ in Lemma C.1(ii) yields

$$x_{1,1} \geq \sqrt{\ln\left(1.2\right)}. \tag{73}$$

By applying Lemma C.2 with f we have

$$\frac{f\left(x\right) + x^2}{f\left(-x\right) + x^2} \leq \max\left\{2, \frac{2}{x_{1,1}^2}\right\} \leq 11, \tag{74}$$

where the last inequality is by (77).

Since $f\left(y\right) \leq 1$ for every $y > 0$ and $f\left(\frac{x}{\sqrt{k}}\right) \geq \frac{1}{2}$, for every $c, k > 0$ we have that

$$\frac{f\left(cx\right)}{f\left(\frac{x}{\sqrt{k}}\right)} \leq 2. \tag{75}$$

Applying Lemma C.3 with $f, D = 2$ yields

$$\frac{f\left(cx\right) + \frac{x^2}{k}}{f\left(-cx\right) + \frac{x^2}{k}} \leq 66kc. \tag{76}$$

□

Lemma C.6. *Let $f = \log(1 + e^x)$ for every $x \in \mathbb{R}$ and let $c, R > 0$. Then, there is $k_0 > 0$ such that for every $k \geq k_0$ and for every $0 \leq x \leq R$*

$$\frac{f(cx) + \frac{x^2}{k}}{f(-cx) + \frac{x^2}{k}} \leq 3R\frac{\log\left(2e^{cR}\right)}{\log(2)}kc.$$

Proof. Let $0 \leq x \leq R$. Applying Lemma C.1 with $k = c = 1$ yields $x_{1,1}$ such that $f\left(-x_{1,1}\right) = x_{1,1}^2$. We now bound $x_{1,1}$. By simple calculations we have that

$$f\left(-\sqrt{\ln\left(1.2\right)}\right) > \left(\sqrt{\ln\left(1.2\right)}\right)^2.$$

Plugging $x = \sqrt{\ln\left(1.2\right)}, k = 1, c = 1$ in Lemma C.1(ii) yields

$$x_{1,1} \geq \sqrt{\ln\left(1.2\right)}. \tag{77}$$

For every $c, k > 0$, since $x \leq R$ and f is non-decreasing we have that

$$f\left(cx\right) \leq f\left(cR\right), \tag{78}$$

furthermore, since $x \geq 0$ and f is increasing we have that

$$f\left(\frac{x}{\sqrt{k}}\right) \geq \log(2). \tag{79}$$

We have that

$$\frac{f(cx)}{f\left(\frac{x}{\sqrt{k}}\right)} \leq \frac{f(cR)}{\log(2)} \tag{80}$$

$$= \frac{\log\left(1+e^{cR}\right)}{\log(2)} \tag{81}$$

$$\leq \frac{\log\left(2e^{cR}\right)}{\log(2)}, \tag{82}$$

where (80) is by (78) and (79), (81) is by the definition of f and (82) holds since $Rc > 0$. Applying Lemma C.4 with $f, D = \frac{\log\left(2e^{cR}\right)}{\log(2)}$ yields

$$\frac{f(cx) + \frac{x^2}{k}}{f(-cx) + \frac{x^2}{k}} \leq 3R\frac{\log\left(2e^{cR}\right)}{\log(2)}kc. \tag{83}$$

□

C.2. Proofs of Our Main Claims

We start by proving the main claims with respect to the sigmoid activation function; see Lemma C.7 and Theorem C.8. We then prove the main claims for the logistic regression activation function; see Lemma C.9 and Theorem C.10.

Lemma C.7. *Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\} \subset \mathbb{R}^d$ be a set of points, sorted by their length. I.e. $\|\boldsymbol{p}_i\| \leq \|\boldsymbol{p}_j\|$ for every $1 \leq i \leq j \leq n$. Let $k > 0$ be a sufficiently large constant and $c_{\text{sigmoid},k}(\boldsymbol{p}, \boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{p}\cdot\boldsymbol{x}}} + \frac{\|\boldsymbol{x}\|^2}{k}$ for every $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{p} \in P$. Then the sensitivity of every $p_j \in P$ is bounded by $s(\boldsymbol{p}) = s_{P,\boldsymbol{1},\mathbb{R}^d,c_{\text{sigmoid},k}}(\boldsymbol{p}) \in O\left(\frac{\|\boldsymbol{p}_j\|k+1}{j}\right)$, and the total sensitivity is*

$$t = \sum_{\boldsymbol{p}\in P} s(\boldsymbol{p}) \in O\left(\log n + k\sum_{j=1}^{n}\frac{\|\boldsymbol{p}_j\|}{j}\right).$$

Proof. Define $f(z) = \frac{1}{1+e^{-z}}$ and $g(z) = z^2$ for every $z \in \mathbb{R}$. Let $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{p}_j \in P$ and $i \in [1, j]$ be an integer. We substitute $c = \|\boldsymbol{p}_i\|$ in Lemma C.5 to obtain that for every $z > 0$

$$\frac{f(\|\boldsymbol{p}_i\| z) + \frac{z^2}{k}}{f(-\|\boldsymbol{p}_i\| z) + \frac{z^2}{k}} \leq 66\|\boldsymbol{p}_i\| k.$$

Denote $b_{\boldsymbol{p}_i} = 66\|\boldsymbol{p}_i\| k$ and multiply the above term by $f(-\|\boldsymbol{p}_i\| z) + \frac{z^2}{k}$ to get

$$f(\|\boldsymbol{p}_i\| z) + \frac{z^2}{k} \leq b_{\boldsymbol{p}_i}\left(f(-\|\boldsymbol{p}_i\| z) + \frac{z^2}{k}\right).$$

Substituting in Lemma 4.2 $\boldsymbol{p} = \boldsymbol{p}_i, f(z) = \frac{1}{1+e^{-z}}, g(z) = z^2, M = 1, f(0) = \frac{1}{2}$ yields

$$\max_{\boldsymbol{p'}\in P} c_{\text{sigmoid},k}(\boldsymbol{p'}, \boldsymbol{x}) \leq 2(b_{\boldsymbol{p}_i} + 1)\, c_{\text{sigmoid},k}(\boldsymbol{p}_i, \boldsymbol{x}). \tag{84}$$

Thus

$$c_{\text{sigmoid},k}(\boldsymbol{p}_j, \boldsymbol{x}) \leq \max_{\boldsymbol{p'}\in P} c_{\text{sigmoid},k}(\boldsymbol{p'}, \boldsymbol{x}) \tag{85}$$

$$\leq 2(b_{\boldsymbol{p}_i} + 1)\, c_{\text{sigmoid},k}(\boldsymbol{p}_i, \boldsymbol{x}), \tag{86}$$

where (92) is since $\boldsymbol{p}_j \in P$ and (93) is by (84). Dividing both sides by $2\left(b_{\boldsymbol{p}_i}+1\right)$ yields

$$c_{\text{sigmoid},k}\left(\boldsymbol{p}_i, \boldsymbol{x}\right) \geq \frac{c_{\text{sigmoid},k}\left(\boldsymbol{p}_j, \boldsymbol{x}\right)}{2\left(b_{\boldsymbol{p}_i}+1\right)}. \tag{87}$$

We now proceed to bound the sensitivity of $\boldsymbol{p}_j$. Since the set of points $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_j\}$ is a subset of P, and since the cost function $c_{\text{sigmoid},k}\left(\boldsymbol{p}_j, \boldsymbol{x}\right)$ is positive we have that

$$\sum_{\boldsymbol{p'} \in P} c_{\text{sigmoid},k}\left(\boldsymbol{p'}, \boldsymbol{x}\right) \geq \sum_{i=1}^{j} c_{\text{sigmoid},k}\left(\boldsymbol{p}_i, \boldsymbol{x}\right). \tag{88}$$

By summing (94) over $i \leq j$, we obtain

$$\begin{aligned}\sum_{i=1}^{j} c_{\text{sigmoid},k}\left(\boldsymbol{p}_i, \boldsymbol{x}\right) &\geq c_{\text{sigmoid},k}(\boldsymbol{p}_j, \boldsymbol{x}) \sum_{i=1}^{j} \frac{1}{2(b_{\boldsymbol{p}_i}+1)} \\ &\geq c_{\text{sigmoid},k}(\boldsymbol{p}_j, \boldsymbol{x}) \frac{j}{2(b_{\boldsymbol{p}_j}+1)},\end{aligned} \tag{89}$$

where the last inequality holds since $b_{p_i} = 66\left\|p_i\right\| k \leq b_{\boldsymbol{p}_j}$ for every $i \leq j$. Combining (95) and (96) yields

$$\sum_{\boldsymbol{p'} \in P} c_{\text{sigmoid},k}(\boldsymbol{p'}, \boldsymbol{x}) \geq \frac{j c_{\text{sigmoid},k}(\boldsymbol{p}_j, \boldsymbol{x})}{2(b_{\boldsymbol{p}_j}+1)} \tag{90}$$

Therefore, the sensitivity is bounded by

$$\begin{aligned}s_{P,\mathbf{1},\mathbb{R}^d,c_{\text{sigmoid},k}}(\boldsymbol{p}_j) &= \sup_{\boldsymbol{x} \in \mathbb{R}^d} \frac{c_{\text{sigmoid},k}(\boldsymbol{p}_j, \boldsymbol{x})}{\sum_{\boldsymbol{p'} \in P} c_{\text{sigmoid},k}(\boldsymbol{p'}, \boldsymbol{x})} \\ &\leq \frac{2(b_{p_j}+1)}{j} \leq \frac{2(66\left\|p_j\right\| k+1)}{j}.\end{aligned}$$

Summing this sensitivity bounds the total sensitivity by

$$\sum_{j=1}^{n} \frac{2(66\left\|p_j\right\| k+1)}{j} \in O\left(\log n + k \sum_{j=1}^{n} \frac{\left\|p_j\right\|}{j}\right).$$

□

In what follows is the main claim and proof for the sigmoid activation function.

Theorem C.8 (Theorem 5.1)**.** *Let P be a set of n points in the unit ball of $\mathbb{R}^d$, $\varepsilon, \delta \in (0,1)$, and $k > 0$ be a sufficiently large constant. For every $p, x \in \mathbb{R}^d$, let $c_{\text{sigmoid},k}\left(\boldsymbol{p}, \boldsymbol{x}\right) = \frac{1}{1+e^{-\boldsymbol{p} \cdot \boldsymbol{x}}} + \frac{\|\boldsymbol{x}\|^2}{k}$. Finally, let (Q, u) be the output of a call to* MONOTONIC-CORESET$(P, \varepsilon, \delta, k)$*; see Algorithm 1. Then, with probability at least $1-\delta$, (Q, u) is an ε-coreset for $(P, \mathbf{1}, \mathbb{R}^d, c_{\text{sigmoid},k})$. Moreover, for $t = (1+k)\log n$ we have $|Q| \in O\left(\frac{t}{\varepsilon^2}\left(d \log t + \log \frac{1}{\delta}\right)\right)$, and (Q, u) can be computed in $O(dn + n \log n)$ time.*

Proof. By (Huggins et al., 2016), the dimension of $(P, w, \mathbb{R}^d, c)$ is at most $d+1$, where (P, w) is a weighted set, $P \subseteq \mathbb{R}^d$, and $c(p, x) = f\left(\boldsymbol{p} \cdot \boldsymbol{x}\right)$ for some monotonic and invertible function f. By Lemma C.7, the total sensitivity of $(P, \mathbf{1}, \mathbb{R}^d, c_{\text{sigmoid},k})$ is bounded by

$$\begin{aligned}t \in O\left(\log n + k \sum_{j=1}^{n} \frac{\left\|p_j\right\|}{j}\right) &= O\left(\log n + k \sum_{j=1}^{n} \frac{1}{j}\right) \\ &= O\left((1+k)\log n\right),\end{aligned}$$

where the last equality holds since the input points are in the unit ball.

Plugging these upper bounds on the dimension and total sensitivity of the query space in Theorem 2.3, yields that a call to Algorithm 1, which samples points from P based on their sensitivity bound, returns the desired coreset (Q, u). The running time is dominated by sorting the length of the points in $O(n \log n)$ time after computing them in $O(nd)$ time. □

Lemma C.9. *Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\} \subset \mathbb{R}^d$ be a set of points, sorted by their length, i.e. $\|\boldsymbol{p}_i\| \leq \|\boldsymbol{p}_j\|$ for every $1 \leq i \leq j \leq n$. Let $R > 0$, $k > 0$ be a sufficiently large constant and $c_{\text{logistic},k}(\boldsymbol{p}, \boldsymbol{x}) = \log(1 + e^{\boldsymbol{p} \cdot \boldsymbol{x}}) + \frac{\|\boldsymbol{x}\|^2}{k}$ for every $\boldsymbol{x} \in B(\boldsymbol{0}, R)$ and $\boldsymbol{p} \in P$. Denote by $B(\boldsymbol{0}, R)$ the ball of radius R centered at the origin. Then the sensitivity of every $p_j \in P$ is bounded by $s(\boldsymbol{p}) = s_{P,\boldsymbol{1},B(\boldsymbol{0},R),c_{\text{logistic},k}}(\boldsymbol{p}) \in O\left(\frac{R^3 \|\boldsymbol{p}_j\| k + R^2}{j}\right)$, and the total sensitivity is*

$$t = \sum_{p \in P} s(p) \in O\left(R^2 \log n + R^3 k \sum_{j=1}^{n} \frac{\|p_j\|}{j}\right).$$

Proof. Define $f(z) = \log(1 + e^z)$ and $g(z) = z^2$ for every $z \in \mathbb{R}$. Let $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{p}_j \in P$ and $i \in [1, j]$ be an integer. We substitute $c = \|\boldsymbol{p}_i\|$ in Lemma C.6 to obtain that for every $z > 0$

$$\frac{f(\|\boldsymbol{p}_i\| x) + \frac{x^2}{k}}{f(-\|\boldsymbol{p}_i\| x) + \frac{x^2}{k}} \leq 3R \frac{\log\left(2e^{\|\boldsymbol{p}_i\| R}\right)}{\log(2)} k \|\boldsymbol{p}_i\|.$$

Denote $b_{\boldsymbol{p}_i} = 3R \frac{\log\left(2e^{\|\boldsymbol{p}_i\| R}\right)}{\log(2)} k \|\boldsymbol{p}_i\|$ and multiply the above term by $f(-\|\boldsymbol{p}_i\| z) + \frac{z^2}{k}$ to get

$$f(\|\boldsymbol{p}_i\| z) + \frac{z^2}{k} \leq b_{\boldsymbol{p}_i} \left(f(-\|\boldsymbol{p}_i\| z) + \frac{z^2}{k}\right).$$

Substituting in Lemma B.1 $\boldsymbol{p} = \boldsymbol{p}_i, f(z) = \log(1 + e^z), g(z) = z^2, M = \log(1 + e^R), f(0) = \log(2)$ yields

$$\begin{aligned} &\max_{\boldsymbol{p'} \in P} c_{\text{logistic},k}(\boldsymbol{p'}, \boldsymbol{x}) \leq \\ &\frac{\log(1 + e^R)(b_{\boldsymbol{p}_i} + 1) c_{\text{logistic},k}\,(\boldsymbol{p}_i, \boldsymbol{x})}{log(2)}. \end{aligned} \tag{91}$$

Thus

$$c_{\text{logistic},k}(\boldsymbol{p}_j, \boldsymbol{x}) \leq \max_{\boldsymbol{p'} \in P} c_{\text{logistic},k}(\boldsymbol{p'}, \boldsymbol{x}) \leq \tag{92}$$

$$\frac{\log(1 + e^R)(b_{\boldsymbol{p}_i} + 1) c_{\text{logistic},k}\,(\boldsymbol{p}_i, \boldsymbol{x})}{log(2)}, \tag{93}$$

where (92) is since $\boldsymbol{p}_j \in P$ and (93) is by (91). Dividing both sides by $\frac{\log(1+e^R)}{log(2)}(b_{\boldsymbol{p}_i} + 1)$ yields

$$c_{\text{logistic},k}(\boldsymbol{p}_i, \boldsymbol{x}) \geq \frac{c_{\text{logistic},k}(\boldsymbol{p}_j, \boldsymbol{x})}{\frac{\log(1+e^R)}{log(2)}(b_{\boldsymbol{p}_i} + 1)}. \tag{94}$$

We now proceed to bound the sensitivity of $\boldsymbol{p}_j$. Since the set of points $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_j\}$ is a subset of P, and since the cost function $c_{\text{logistic},k}(\boldsymbol{p}_j, \boldsymbol{x})$ is positive we have that

$$\sum_{\boldsymbol{p'} \in P} c_{\text{logistic},k}(\boldsymbol{p'}, \boldsymbol{x}) \geq \sum_{i=1}^{j} c_{\text{logistic},k}(\boldsymbol{p}_i, \boldsymbol{x}). \tag{95}$$

By summing (94) over $i \leq j$, we obtain

$$\begin{aligned}
&\sum_{i=1}^{j} c_{\text{logistic},k}\boldsymbol{p}_i, \boldsymbol{x}) \geq \\
&c_{\text{logistic},k}(\boldsymbol{p}_j, \boldsymbol{x}) \sum_{i=1}^{j} \frac{\log(2)}{\log(1+e^R)(b_{\boldsymbol{p}_i}+1)} \geq \\
&c_{\text{logistic},k}(\boldsymbol{p}_j, \boldsymbol{x}) \frac{j\log(2)}{\log(1+e^R)(b_{\boldsymbol{p}_j}+1)},
\end{aligned} \tag{96}$$

where the last inequality holds since $b_{p_i} = 3R\frac{\log(1+e^R)}{log(2)}\left\|p_i\right\|k \leq b_{\boldsymbol{p}_j}$ for every $i \leq j$. Combining (95) and (96) yields

$$\sum_{\boldsymbol{p'} \in P} c_{\text{logistic},k}(\boldsymbol{p'}, \boldsymbol{x}) \geq \frac{j\log(2) c_{\text{logistic},k}(\boldsymbol{p}_j, \boldsymbol{x})}{\log(1+e^R)(b_{\boldsymbol{p}_j}+1)} \tag{97}$$

Therefore, the sensitivity is bounded by

$$\begin{aligned}
&s_{P,\mathbf{1},B(\mathbf{0},R),c_{\text{logistic},k}}(\boldsymbol{p}_j) = \\
&\sup_{\boldsymbol{x} \in B(\mathbf{0},R)} \frac{c_{\text{logistic},k}(\boldsymbol{p}_j, \boldsymbol{x})}{\sum_{\boldsymbol{p'} \in P} c_{\text{logistic},k}(\boldsymbol{p'}, \boldsymbol{x})} \leq \\
&\frac{\log(1+e^R)(b_{\boldsymbol{p}_j}+1)}{j\log(2)} \leq \\
&\frac{\log(1+e^R)\left(\frac{3R\log(1+e^R)}{log(2)}\left\|p_j\right\|k+1\right)}{j\log(2)}.
\end{aligned}$$

Thus, $s_{P,\mathbf{1},B(\mathbf{0},R),c_{\text{logistic},k}}(\boldsymbol{p}_j) \in O\left(\frac{R^3\left\|\boldsymbol{p}_j\right\|k+R^2}{j}\right)$. Summing this sensitivity bounds the total sensitivity by

$$\sum_{j=1}^{n} \frac{R^3\left\|\boldsymbol{p}_j\right\|k+R^2}{j} \in O\left(R^2\log n + R^3 k \sum_{j=1}^{n} \frac{\left\|p_j\right\|}{j}\right).$$

□

In what follows is the main claim and proof for the logistic regression function.

Theorem C.10 (Theorem 5.2)**.** *Let P be a set of n points in the unit ball of $\mathbb{R}^d$, $\varepsilon, \delta \in (0,1)$, $R, k > 0$ where k is a sufficiently large constant, and $t = R\log n(1+Rk)$. For every $\boldsymbol{p} \in \mathbb{R}^d$,$\boldsymbol{x} \in B(\mathbf{0},R)$ let $c_{\text{logistic},k}(\boldsymbol{p},\boldsymbol{x}) = \log\left(1+e^{\boldsymbol{p}\cdot\boldsymbol{x}}\right) + \frac{\left\|\boldsymbol{x}\right\|^2}{k}$. Finally, let (Q,u) be the output of a call to* MONOTONIC-CORESET(P,k,m) *where $m \in \Omega\left(\frac{t}{\varepsilon^2}\left(d^2\ln t + \ln\frac{1}{\delta}\right)\right)$; see Algorithm 1. Then, with probability at least $1-\delta$, (Q,u) is an ε-coreset for $(P,\mathbf{1},\mathbb{R}^d,c_{\text{logistic},k})$. Moreover, $|Q| \in O(m)$ and (Q,u) can be computed in $O(nd+n\log n)$ time.*

Proof. By (Huggins et al., 2016), the dimension of $(P,w,\mathbb{R}^d,c)$ is at most $d+1$, where (P,w) is a weighted set, $P \subseteq \mathbb{R}^d$, and $c(p,x) = f\left(\boldsymbol{p}\cdot\boldsymbol{x}\right)$ for some monotonic and invertible function f. By Lemma C.9, the total sensitivity of $(P,\mathbf{1},\mathbb{R}^d,c_{\text{logistic},k})$ is bounded by

$$\begin{aligned}
&t \in O\left(R^2\log n + R^3 k \sum_{j=1}^{n} \frac{\left\|p_j\right\|}{j}\right) = \\
&O\left(R^2\log n + R^3 k \sum_{j=1}^{n} \frac{1}{j}\right) = O\left(R^2\log n(1+Rk)\right),
\end{aligned}$$

where the last equality holds since the input points are in the unit ball.

Plugging these upper bounds on the dimension and total sensitivity of the query space in Theorem 2.3, yields that a call to MONOTONIC-CORESET, which samples points from P based on their sensitivity bound, returns the desired coreset (Q, u). The running time is dominated by sorting the length of the points in $O(n \log n)$ time after computing them in $O(nd)$ time. Sampling $m = |Q|$ points from n points according to such a given distribution takes $O(1)$ time after pre-processing of $O(n)$ time. □

C.3. Additional Problems

Theorem C.11. *Let P be a set of n points in the unit ball of $\mathbb{R}^d$, $\varepsilon, \delta \in (0, 1)$, $R, k > 0$ where k is a sufficiently large constant, and $t = R \log n(1 + Rk)$. For every $\boldsymbol{p} \in \mathbb{R}^d$,$\boldsymbol{x} \in B(\boldsymbol{0}, R)$ let $c_{\text{svm},k}(\boldsymbol{p}, \boldsymbol{x}) = \max(0, 1 + \boldsymbol{p} \cdot \boldsymbol{x}) + \frac{\|\boldsymbol{x}\|^2}{k}$. Finally, let (Q, u) be the output of a call to* MONOTONIC-CORESET(P, k, m) *where $m \in \Omega\left(\frac{t}{\varepsilon^2}\left(d^2 \ln t + \ln \frac{1}{\delta}\right)\right)$; see Algorithm 1. Then, with probability at least $1 - \delta$, (Q, u) is an ε-coreset for $(P, \mathbf{1}, \mathbb{R}^d, c_{\text{logistic},k})$. Moreover, $|Q| \in O(m)$ and (Q, u) can be computed in $O(nd + n \log n)$ time.*

The proof of the theorem above follows similarly to the proof of Theorem C.10 from the previos section.

D. Bounding the VC-dimension

In what follows we first give the formal definition of the VC dimension of a given query space. We then formally bound the VC dimensions of the sigmoid and logistic regression cost functions.

Definition D.1 (VC-dimension)**.** (Feldman & Langberg, 2011; Vapnik & Chervonenkis, 1971) For a query space (P, w, X, c) we define

$$range\,(\boldsymbol{x}, r) = \{\boldsymbol{p} \in P \mid w\,(\boldsymbol{p})\, c\,(\boldsymbol{p}.\boldsymbol{x}) \leq r\}\,,$$

for every $x \in X$ and $r \geq 0$. The (VC) dimension of (P, w, X, c) is the size $|G|$ of the largest subset $G \subseteq P$ such that have

$$|\{G \cap range\,(\boldsymbol{x}, r)\,|\boldsymbol{x} \in X, r \geq 0\}| = 2^{|G|}.$$

Theorem D.2 (Theorem 8.14 in (Lucic et al., 2017) and generalized in (Lucic et al., 2017))**.** *Let h be a function from $\mathbb{R}^m \times \mathbb{R}^d$ to $\{0, 1\}$, determining the class*

$$\mathcal{H} = \{h_\theta(\cdot) \mid h_\theta : \mathcal{X} \to \mathbb{R}_{++}, \theta \in \mathbb{R}^m\}\,.$$

Suppose that h can be computed by an algorithm that takes as input the pair $(\theta, x) \in \mathbb{R}^m \times \mathbb{R}^n$ and returns $h_\theta(x)$ after no more than t of the following operations:

1. *the arithmetic operations $+, -, \times$ and $/$ on real numbers,*
2. *jumps conditioned on $>, \geq, <, \leq, =$ and $\neq$ comparisons of real numbers, and*
3. *output $0, 1$*

and no more than p operations of the exponential function $x \to e^x$ on real numbers, then the VC-dimension of $\mathcal{H}$ is $O(m^2p^2 + mp(t + \log(mp)))$.

Lemma D.3 (VC-dimension of the Sigmoid loss function)**.** *Let $k > 0$ be a constant and $(P, w, \mathbb{R}^d, c_{\text{sigmoid},k})$ be a query space where $c_{\text{sigmoid},k}(\boldsymbol{p}, \boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{p} \cdot \boldsymbol{x}}} + \frac{\|\boldsymbol{x}\|^2}{k}$ for every $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{p} \in P$. Then the VC-dimension of $(P, w, \mathbb{R}^d, c_{\text{sigmoid},k})$ is at most $O(d^2)$.*

Proof. Observe that for every $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{p} \in P$ we can evaluate $c_{\text{sigmoid},k}(\boldsymbol{p}, \boldsymbol{x})$ using $t \in O(d)$ addition, multiplication, and division operations and $p \in O(1)$ operations of the exponential function $x \to e^x$. Then, by Theorem D.2, the VC-dimension of $(P, w, \mathbb{R}^d c_{\text{sigmoid},k})$ is bounded by $O(d^2)$. □

Lemma D.4 (VC-dimension of the Logistic Regression loss function)**.** *Let $k > 0$ be constants and $(P, \mathbf{1}, \mathbb{R}^d, c_{\text{logistic},k})$ be a query space where $c_{\text{logistic},k}(\boldsymbol{p}, \boldsymbol{x}) = \log(1 + e^{\boldsymbol{p} \cdot \boldsymbol{x}}) + \frac{\|\boldsymbol{x}\|^2}{k}$ for every $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{p} \in P$. Then the VC-dimension of $(P, \mathbf{1}, \mathbb{R}^d, c)$ is at most $O(d^2)$.*

Proof. We first bound the VC-dimension of $(P, \mathbf{1}, \mathbb{R}^d, g)$, where

$$g(\boldsymbol{p}, \boldsymbol{x}) = e^{c(\boldsymbol{p},\boldsymbol{x})} = (1 + e^{\boldsymbol{p}\cdot\boldsymbol{x}})e^{\frac{\|\boldsymbol{x}\|^2}{k}}.$$

Observe that we can evaluate $g(\boldsymbol{p}, \boldsymbol{x})$ using $t \in O(d)$ addition, multiplication, and division operations and $p \in O(1)$ operations of the exponential function $x \to e^x$. Then, by Theorem D.2, the VC-dimension of $(P, \mathbf{1}, \mathbb{R}^d, g)$ is bounded by $O(d^2)$.

We now show that the VC-dimension of $(P, \mathbf{1}, \mathbb{R}^d, c)$ is upper bounded by the VC-dimension of $(P, \mathbf{1}, \mathbb{R}^d, g)$. Recall that for the query space $(P, \mathbf{1}, \mathbb{R}^d, c)$ and every $\boldsymbol{x} \in \mathbb{R}^d$ and $r \geq 0$ we have that $range(\boldsymbol{x}, r) = \{\boldsymbol{p} \in P \mid c(\boldsymbol{p}, \boldsymbol{x}) \leq r\}$. For the query space $(P, \mathbf{1}, \mathbb{R}^d, g)$ and every $\boldsymbol{x} \in \mathbb{R}^d$ and $r \geq 0$ we have that $range'(\boldsymbol{x}, r) = \{\boldsymbol{p} \in P \mid g(\boldsymbol{p}, \boldsymbol{x}) \leq r\}$.

For every $r \geq 0$ let $r_g := e^r$. Then we have that

$$\begin{aligned} range(\boldsymbol{x}, r) &= \{\boldsymbol{p} \in P \mid c(\boldsymbol{p}, \boldsymbol{x}) \leq r\} = \left\{\boldsymbol{p} \in P \mid e^{c(\boldsymbol{p},\boldsymbol{x})} \leq e^r\right\} \\ &= \{\boldsymbol{p} \in P \mid g(\boldsymbol{p}, \boldsymbol{x}) \leq r_g\} = range'(\boldsymbol{x}, r_g). \end{aligned}$$

Therefore, for every $G \subseteq P$ we have that

$$\left|\left\{G \cap range\left(\boldsymbol{x}, r\right) | \boldsymbol{x} \in X, r \geq 0\right\}\right| \leq \left|\left\{G \cap range'\left(\boldsymbol{x}, r_g\right) | \boldsymbol{x} \in X, r_g \geq 0\right\}\right|.$$

Hence, by the definition of VC-dimension (see Definition D.1), we have that the VC-dimension of the query space $(P, \mathbf{1}, \mathbb{R}^d, c)$ is upper bounded by the VC-dimension of the query space $(P, \mathbf{1}, \mathbb{R}^d, g)$ which is upper bounded by $O(d^2)$. □

E. Known results

For completeness, in what follows we formally state known claims, which were utilized in the proofs of the previous sections.

Theorem E.1 (Intermediate Value Theorem)**.** *Let $a, b \in \mathbb{R}$ such that $a < b$ and let $f : [a, b] \to \mathbb{R}$ be a continuous function. Then for every u such that*

$$\min\left\{f\left(a\right), f\left(b\right)\right\} \leq u \leq \max\left\{f\left(a\right), f\left(b\right)\right\},$$

there is $c \in (a, b)$ such that $f\left(c\right) = u$.

Theorem E.2 (Mean Value Theorem)**.** *Let $a, b \in \mathbb{R}$ such that $a < b$ and $f : [a, b] \to \mathbb{R}$ a continuous function on the closed interval $[a, b]$ and differentiable on the open interval (a, b). Then there is $c \in (a, b)$ such that*

$$f'\left(c\right) = \frac{f\left(b\right) - f\left(a\right)}{b - a}.$$

Theorem E.3 (Inverse of Strictly Monotone Function Theorem)**.** *Let $I \subseteq \mathbb{R}$. Let $f : I \to \mathbb{R}$ be strictly monotonic function. Let the image of f be J. Then f has an inverse function $f^{-1}$and*

- If f is strictly increasing then so is f^{-1}.
- If f is strictly decreasing then so is f^{-1}.