Importance Sampling via Local Sensitivity

Anant Raj
MPI for Intelligen Systems, Tübingen

Cameron Musco
UMass Amherst

Lester Mackey
Microsoft Research New England

Abstract

Given a loss function $F : \mathcal{X} \to \mathbb{R}^+$ that can be written as the sum of losses over a large set of inputs $a_1, \ldots, a_n$, it is often desirable to approximate F by subsampling the input points. Strong theoretical guarantees require taking into account the importance of each point, measured by how much its individual loss contributes to $F(x)$. Maximizing this importance over all $x \in \mathcal{X}$ yields the *sensitivity score* of a_i. Sampling with probabilities proportional to these scores gives strong guarantees, allowing one to approximately minimize of F using just the subsampled points.

Unfortunately, sensitivity sampling is difficult to apply since (1) it is unclear how to efficiently compute the sensitivity scores and (2) the sample size required is often impractically large. To overcome both obstacles we introduce *local sensitivity*, which measures data point importance in a ball around some center x_0. We show that the local sensitivity can be efficiently estimated using the *leverage scores* of a quadratic approximation to F and that the sample size required to approximate F around x_0 can be bounded. We propose employing local sensitivity sampling in an iterative optimization method and analyze its convergence when F is smooth and convex.

1 Introduction

In this work we consider finite sum minimization problems of the following form.

Definition 1 (Finite Sum Problem)**.** Given data points $a_1, \ldots, a_n \in \mathbb{R}^d$, nonnegative functions $f_1, \ldots, f_n : \mathbb{R} \to \mathbb{R}^+$, and a nonnegative function $\gamma : \mathbb{R}^d \to \mathbb{R}^+$, minimize over $x \in \mathcal{X} \subseteq \mathbb{R}^d$

$$F(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(a_i^T x) + \gamma(x). \qquad (1)$$

Definition 1 captures a number of important problems, including penalized empirical risk minimization (ERM) for linear regression, generalized linear models, and support vector machines. When n is large, minimizing $F(x)$ can be expensive. In some cases, for example, it may be impossible to load the full dataset $a_1, \ldots, a_n$ into memory.

1.1 Function Approximation via Data Subsampling

To reduce the burden of solving a finite sum problem, one commonly minimizes an approximation to F formed by independently subsampling data points a_i (and hence summands $f_i(a_i^T x)$) with some fixed probability weights. More formally:

Definition 2 (Subsampled Finite Sum Problem)**.** Consider the setting of Definition 1. Given a target sample size m and a probability distribution $P = \{p_1, \ldots, p_n\}$ over $[n] \triangleq \{1, \ldots, n\}$, select $i_1, ..., i_m$ i.i.d. from P and minimize over $x \in \mathcal{X} \subseteq \mathbb{R}^d$

$$F^{(P,m)}(x) := \frac{1}{mn} \sum_{j=1}^{m} \frac{f_{i_j}(a_{i_j}^T x)}{p_{i_j}} + \gamma(x). \qquad (2)$$

We can see that for any x, $\mathbb{E}[F^{(P,m)}(x)] = F(x)$. If the sampled function concentrates well around $F(x)$, then it can serve effectively as a surrogate for minimizing F. Most commonly, P is set to the uniform distribution. Unfortunately, if $F(x)$ is dominated by the values of a relatively few large $f_i(a_i^T x)$, unless m is very large, uniform subsampling will miss these important data points and $F^{(P,m)}(x)$ will often underestimate $F(x)$. This can happen, for example, when $a_1, ..., a_n$ fall into clusters of non-uniform size. Data points in smaller clusters are important in selecting an optimal x but are often underrepresented in a uniform sample.

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

1.2 Importance Sampling via Sensitivity

A remedy to the weakness of uniform subsampling is to apply importance sampling: preferentially sample the functions $f_i(a_i^T x)$ that contribute most significantly to $F(x)$. If, for example, we set $p_i \propto \frac{f_i(a_i^T x)}{\sum_{i=1}^n f_i(a_i^T x)+\gamma(x)}$ for each $i \in [n]$, then a standard concentration argument would imply that $(1-\epsilon)F(x) \leq F^{(P,m)}(x) \leq (1+\epsilon)F(x)$ with probability at least $\mathbf{1}-\delta$ if $m = \Theta\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$. However, typically the relative the importance of each point, $\frac{f_i(a_i^T x)}{\sum_{i=1}^n f_i(a_i^T x)+\gamma(x)}$, will depend on the choice of x. This motivates the definition of *sensitivity* [Langberg and Schulman, 2010].

Definition 3 (Sensitivity)**.** For $a_1, \ldots, a_n \in \mathbb{R}^d$, the *sensitivity* of point a_i with respect to a finite sum function F (Definition 1) with domain $\mathcal{X} \subseteq \mathbb{R}^d$ is

$$\sigma_{F,\mathcal{X}}(a_i) = \sup_{x \in \mathcal{X}} \frac{f_i(a_i^T x)}{\sum_{j=1}^n f_j(a_j^T x) + n\gamma(x)}.$$

The *total sensitivity* is defined as $\mathcal{G}_{F,\mathcal{X}} = \sum_{i=1}^n \sigma_{F,\mathcal{X}}(a_i)$.

A standard concentration argument yields the following approximation guarantee for sensitivity sampling.

Lemma 4. *Consider the setting of Definition 1. For all $i \in [n]$, let $s_i \geq \sigma_{F,\mathcal{X}}(a_i)$, $S = \sum_{i=1}^n s_i$, and $P = \left\{\frac{s_1}{S}, \ldots, \frac{s_n}{S}\right\}$. There is a fixed constant c such that, for any $\epsilon, \delta \in (0,1)$, any fixed $x \in \mathcal{X}$, and $m \geq \frac{c \cdot S \log(2/\delta)}{\epsilon^2}$,*

$$(1-\epsilon)F(x) \leq F^{(P,m)}(x) \leq (1+\epsilon)F(x)$$

with probability $\geq 1-\delta$.

That is, subsampling data points by their sensitivities approximately preserves the value of F *for any fixed* $x \in \mathcal{X}$ with high probability. It can thus be argued that F can be approximately minimized by minimizing the sampled function $F^{(P,m)}$. We first define:

Definition 5 (Range Space)**.** A range space is a pair $\mathcal{R} = (\mathcal{F}, \text{ranges})$, where $\mathcal{F}$ is a set and ranges is a set of subsets of $\mathcal{F}$. The VC dimension $\Delta(\mathcal{R})$ is the size of the largest $G \subseteq \mathcal{F}$ such that G is shattered by ranges: i.e., $|\{G \cap R | R \in \text{ranges}\}| = 2^{|G|}$.

Let $\mathcal{F}$ be a finite set of functions mapping $\mathbb{R}^d \to \mathbb{R}^+$. For every $x \in \mathbb{R}^d$ and $r \in \mathbb{R}^+$, let $\text{range}_{\mathcal{F}}(x,r) = \{f \in \mathcal{F} | f(x) \geq r\}$ and $\text{ranges}(\mathcal{F}) = \{\text{range}_{\mathcal{F}}(x,r) | x \in \mathbb{R}^d, r \in \mathbb{R}^+\}$. We say $R_{\mathcal{F}} = (\mathcal{F}, \text{ranges}(\mathcal{F}))$ is the range space induced by $\mathcal{F}$.

With the notion of range space in place, we can recall the following general approximation theorem.

Theorem 6 (Theorem 9 [Munteanu et al., 2018])**.** *Consider the setting of Definition 1. For all $i \in [n]$, let $s_i \geq \sigma_{F,\mathcal{X}}(a_i)$, $S = \sum_{i=1}^n s_i$, and $P = \left\{\frac{s_1}{S}, \ldots, \frac{s_n}{S}\right\}$. For some finite c and all $\epsilon, \delta \in (0, 1/2)$, if*

$$m \geq c \cdot \frac{S}{\epsilon^2}\left(\Delta \log S + \log\left(\frac{1}{\delta}\right)\right),$$

then, with probability at least $1-\delta$,

$$(1-\epsilon)F(x) \leq F^{(P,m)}(x) \leq (1+\epsilon)F(x), \forall x \in \mathcal{X}$$

Here, Δ is an upper bound on the VC-dimension $\Delta(\mathcal{R}_{\mathcal{F}})$ where $\mathcal{F}$ is the set $\left\{\frac{f_1(a_1^T x)}{mn \cdot p_1}, \ldots, \frac{f_n(a_n^T x)}{mn \cdot p_n}\right\}$.

Munteanu et al. [2018] show that $\Delta = d+1$ suffices for logistic regression where d is the dimension of the input points. If all f_i are from the class of invertible functions, then a similar bound on Δ can be expected.

1.2.1 Barriers to the Sensitivity Sampling in Practice

Theorem 6 is quite powerful: it can be used to achieve sensitivity-sampling-based approximation algorithms with provable guarantees for a wide range of problems [Feldman and Langberg, 2011, Huggins et al., 2016, Lucic et al., 2016, Munteanu et al., 2018]. However, there are two major barriers that have hindered more widespread practical adoption of sensitivity sampling:

1. **Computability:** It is difficult to compute or even approximate the sensitivity $\sigma_{F,\mathcal{X}}(a_i)$ since it is not clear how to take the supremum over all $x \in \mathcal{X}$ in the expression of Definition 3. Closed form expressions for the sensitivity are known only in a few special cases, such as least squares regression (where the sensitivity is closely related to the well-studied *statistical leverage scores*).
2. **Pessimistic Bounds:** The sensitivity score is a very 'worst case' importance metric, since it considers the supremum of $\frac{f_i(a_i^T x)}{\sum_{j=1}^n f_j(a_j^T x)+n\gamma(x)}$ over all $x \in \mathcal{X}$, including, e.g., x that may be very far from the true minimizer of F. In many cases, it is possible to construct, for each a_i, some worst case x that forces this ratio to be high. Thus, all sensitivities are large and the total sensitivity $\mathcal{G}_{F,\mathcal{X}}$ is large. The sample complexities in Lemma 4 and Theorem 6 depend on $S \geq \mathcal{G}_{F,\mathcal{X}}$ and so will be too large to be useful in practice. See Figure 1 for a simple example of when this issue can arise.

1.3 Our Approach: Local Sensitivity

We propose to overcome the above barriers via a simple idea: *local sensitivity.* Instead of sampling with the sensitivity over the full domain $\mathcal{X}$ as in Definition 3, we consider the sensitivity over a small ball.

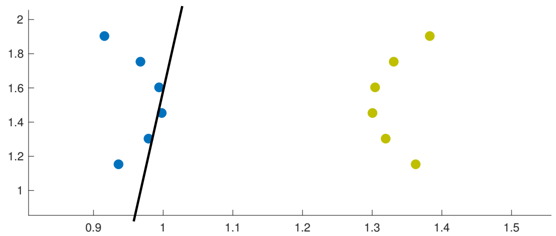


Figure 1: Consider a classification problem with two classes A_1, A_2, shown in blue and green. Let $f_i(a_i^T x)$ be any loss function with $f_i(a_i^T x) = 0$ if a_i is correctly classified by the hyperplane defined by x. Since for each a_i, there is some x (e.g., corresponding to the black line shown) that misclassifies *only* a_i, we have $\sigma_{\mathcal{F},\mathbb{R}^d}(a_i) = 1$ for all a_i. Thus, the total sensitivity is $\mathcal{G}_{F,\mathcal{X}} = n$ and so the sampling results of Lemma 4 and Theorem 6 are vacuous – they require sampling $\geq n$ points, even for this simple task.

Specifically, for some radius r and center y we let $B(r, y) = \{x \in \mathbb{R}^d : \|x - y\| < r\}$ and consider $\sigma_{F,\mathcal{X}\cap B(r,y)}(a_i)$. Sampling by this local sensitivity will give us a function $F^{(P,m)}$ that *approximates* F *well on the entire ball* $B(r, y)$. Thus, we can approximately minimize F on this ball. We can approximately minimize F globally via an iterative scheme: at each step we set x_i to the approximate optimum of F over the ball $B(r_i, x_{i-1})$ (computed via local sensitivity sampling). This approach has two major advantages:

1. We can often locally approximate each F by a simple function, for which we can compute the local sensitivities in closed form. This will yield an approximation to the true local sensitivities. Specifically, we will consider a local quadratic approximation to F, whose sensitivities are given by the *leverage scores* of an appropriate matrix.

2. By definition, the local sensitivity $\sigma_{F,\mathcal{X}\cap B(r,y)}$ is *always* upper bounded by the global sensitivity $\sigma_{F,\mathcal{X}}$, and typically the sum of local sensitivities will be much smaller than the total sensitivity $\mathcal{G}_{F,\mathcal{X}}$. This allows us to take fewer samples to approximately minimize F locally over $B(r, y)$.

1.4 Related Work

The sensitivity sampling framework has been successfully applied to a number of problems, including clustering [Bachem et al., 2015, Feldman and Langberg, 2011, Lucic et al., 2016], logistic regression [Huggins et al., 2016, Munteanu et al., 2018], and least squares regression, in the form of leverage score sampling [Cohen et al., 2015, Drineas et al., 2006, Mahoney, 2011]. In these works, upper bounds are given on the sensitivity of each data point, and it is shown that the sum of these bounds, and thus the required sample size for approximate optimization, is small. We aim to expand the applicability of sensitivity-based methods to functions for which a bound on the sensitivity cannot be obtained or for which the total sensitivity is inherently large.

The local-sensitivity-based iterative method that we will discuss is closely related to quasi-Newton methods [Dennis and Moré, 1977], especially those that approximate the Hessian via leverage score sampling [Xu et al., 2016, Ye et al., 2017]. In each iteration, we estimate local sensitivities by considering the sensitivities of a local quadratic approximation to F. As shown in Section 2, these sensitivities can be bounded using the leverage scores of the Hessian, and thus our sampling probabilities are closely related to those used in the above works. Unlike a quasi-Newton method however, we use the sensitivities to directly optimize F locally, rather than the quadratic approximation itself. In this way, our method is closer to a trust region method [Chen et al., 2018] or an approximate proximal point method [Frostig et al., 2015].

Recently, [Agarwal et al., 2017] and [Chowdhury et al., 2018] have suggested iterative algorithms for regularized least squares regression and ERM for linear models that sample a subset of data points by their leverage scores (closely related to sensitivities) in each step. These works employ this sampling in a different way than us, using the subsample to precondition each iterative step. While they give strong theoretical guarantees for the problems studied, this technique applies to a less general class of problems than our method.

The sensitivity scores for ℓ_2 regression are commonly known as leverage scores, and a long line of work [Altschuler et al., 2018, Rudi et al., 2018, see, e.g.,] has focused on approximating these scores more quickly. These approximation techniques do not extend to general sensitivity score approximation however. Additionally, our paper in no way attempts to develop a faster algorithm for leverage score sampling. We focus on introducing the notion of local sensitivity, which allows leverage score based methods to be applied to optimization problems well beyond ℓ_2 regression.

1.5 Road Map

Our contributions are presented as follows. In Section 2 we show that the sensitivity scores of a quadratic approximation to a function are given by the leverage scores of an appropriate matrix. We use these scores to bound the local sensitivity scores of the true function. In Section 3 we discuss how to subsample using these approximate local sensitivities with the aim of approximately minimizing the function over a small ball. We describe how to use this approach to iteratively optimize the function. In Section 4 we give an analysis of this iterative method for convex functions.

2 Leverage Scores as Sensitivities of Quadratic Functions

We start by showing how to approximate the local sensitivity $\sigma_{F,\mathcal{X}\cap B(r,y)}$ over some ball by approximating F with a quadratic function on this ball. F's sensitivities can be approximated by those of this quadratic function, which we in turn bound in closed form by the leverage scores of an appropriate matrix (a rank-1 perturbation of F's Hessian at y). The leverage scores are given by:

Definition 7 (Leverage Scores [Alaoui and Mahoney, 2015, Cohen et al., 2017])**.** For any $C \in \mathbb{R}^{n\times p}$ with i^{th} row c_i, the i^{th} λ-ridge leverage score is the sensitivity of $F(z) = \|Cz\|_2^2 + \lambda\|z\|_2^2$:

$$\ell_i^\lambda(C) := \max_{\{z\in\mathbb{R}^p : \|z\|_2 > 0\}} \frac{[Cz]_i^2}{\|Cz\|_2^2 + \lambda\|z\|_2^2}.$$

We have $\ell_i^\lambda(C) = c_i^T(C^TC + \lambda I)^{-1}c_i$. (See Lemma 17 in Appendix A).

Our eventual iterative method will employ a proximal function, and thus in this section we consider this function, which reduces to F when $\lambda = 0$:

Definition 8 (Proximal Function)**.** For a function $F : \mathcal{X} \to \mathbb{R}$, define $F_{\lambda,y}(x) = F(x) + \lambda\|x - y\|_2^2$.

Using Definition 7 and the associated Lemma 17 we establish the following in Appendix A.

Theorem 9 (Sensitivity of Quadratic Approximation)**.** *Consider F as in Def. 1 along with the quadratic approximation to the proximal function $F_{\lambda,y}$ (Def. 8) around $y \in \mathcal{X}$. If $A \in \mathbb{R}^{n\times d}$ is the data matrix with i^{th} row equal to a_i, then*

$$\begin{aligned}\tilde{F}_{\lambda,y}(x) &:= \frac{1}{n}\sum_{i=1}^{n}\Big[f_i(a_i^Ty) + a_i^T(x-y)\cdot f'(a_i^Ty) \\ &\quad + \frac{1}{2}(a_i^T(x-y))^2 \cdot f''(a_i^Ty)\Big] + \gamma(x) + \lambda\|x-y\|_2^2 \\ &:= F(y) + (x-y)^TA^T\alpha_y + \frac{1}{2}(x-y)^TA^TH_yA(x-y) \\ &\quad + \gamma(x) + \lambda\|x-y\|_2^2\end{aligned} \tag{3}$$

where $[\alpha_y]_i = \frac{1}{n}f_i'(a_i^Ty)$, and H_y is the diagonal matrix with $[H_y]_{i,i} = \frac{1}{n}f''(a_i^Ty)$. Assuming that H_y is nonnegative, the sensitivity scores of $\tilde{F}_{\lambda,y}$ with respect to $B(r,y)$ can be bounded as

$$\sigma_{\tilde{F}_{\lambda,y},B(r,y)}(a_i) \le \beta\cdot\ell_i^\lambda(C) + \frac{f_i(a_i^Ty)}{\eta}, \tag{4}$$

where $C = [H_y^{1/2}A, \frac{1}{\delta}H_y^{-1/2}\alpha_y]$, $\ell_i^\lambda(C)$ is the leverage score of Def. 7, $\eta = \min_{x\in B(r,y)} \tilde{F}_{\lambda,y}(x)$, $\delta = \min_{x\in B(r,y)} \gamma(x)$, and $\beta = \max\left(1, 1 - \frac{F(y) - \frac{1}{n}\sum_{i=1}^n \frac{f'(a_i^Ty)^2}{4f''(a_i^Ty)}}{\eta}\right)$.

Note that if we consider a small enough ball, where $\tilde{F}_{\lambda,y}$ well approximates $F_{\lambda,y}$, we expect $\eta = \min_{x\in B(r,y)} \tilde{F}_{\lambda,y}(x) = \Theta(F(y))$. Thus, the additive $\frac{f_i(a_i^Ty)}{\eta}$ term on each sensitivity will contribute only a $\frac{\sum f_i(a_i^Ty)}{\Theta(F(y))} = O(1)$ additive factor to the total sensitivity bound and sample size.

2.1 Efficient Computation of Leverage Score Sensitivities

The sensitivity upper bound (4) of Theorem 9 can be approximated efficiently as long as we can efficiently approximate the leverage scores $\ell_i^\lambda(C) = c_i^T(C^TC + \lambda I)^{-1}c_i$, where $C = [H_y^{1/2}A, \frac{1}{\delta}H_y^{-1/2}\alpha_y]$. We can use a block matrix inversion formula to find that

$$\begin{aligned}(C^TC + \lambda I)^{-1} &= \begin{bmatrix} A^TH_yA + \lambda I & \frac{1}{\delta}A^T\alpha_y \\ \frac{1}{\delta}\alpha_y^TA & \|\alpha_y\|_2^2 + \lambda \end{bmatrix}^{-1} \\ &= \begin{bmatrix} A_1 & A_2 \\ A_2^\top & \frac{1}{k} \end{bmatrix}\end{aligned}$$

where $A_1 = (A^TH_yA + \lambda I)^{-1} + \frac{1}{k}(A^TH_yA + \lambda I)^{-1}A^T\alpha_y\alpha_y^TA(A^TH_yA + \lambda I)^{-1}$, $k = \|\alpha_y\|_2^2 + \delta^2\lambda - \alpha_y^TA(A^TH_yA + \lambda I)^{-1}A^T\alpha_y$, and $A_2 = -\frac{\delta}{k}(A^TH_yA + \lambda I)^{-1}A^T\alpha_y$.

Thus, if we have a fast algorithm for applying $(A^TH_yA + \lambda I)^{-1}$ to a vector we can quickly apply $(C^TC + \lambda I)^{-1}$ to a vector and compute the leverage scores $\ell_i^\lambda(C) = c_i^T(C^TC + \lambda I)^{-1}c_i$. Via standard Johnson-Lindenstrauss sketching techniques [Spielman and Srivastava, 2011] it in fact suffices to apply this inverse to $O(\log n/\delta)$ vectors to approximate each score up to constant factor with probability $\ge 1-\delta$. In practice, one can use traditional iterative methods such as conjugate gradient, iterative sampling methods such as those presented in [Cohen et al., 2015, 2017], or fast sketching methods [Clarkson and Woodruff, 2017, Drineas et al., 2012].

2.2 True Local Sensitivity from Quadratic Approximation

As long as the quadratic approximation $\tilde{F}_{\lambda,y}$ approximates $F_{\lambda,y}$ sufficiently well on the ball $B(r,y)$, we can use Theorem 9 to approximate the true local sensitivity $\sigma_{F_{\lambda,y},\mathcal{X}\cap B(r,y)}(a_i)$. We start by discussing our approximation assumptions.

Defining α_y as in Theorem 9, for some $B_y(x)$ which itself is a function of x we have:

$$F(x) = F(y) + (x-y)^\top A^\top \alpha_y + (x-y)^\top A^\top H_y A(x-y) \\ + \gamma(x) + B_y(x)\|x-y\|_2^3.$$

Without loss of generality, we assume that $B_y(x) > 0$ for x in the above equation or we just shift the overall function vertically by adjusting $\gamma(\cdot)$ to have the quadratic appropriator be an under approximation of the true function. If the function F has a C Lipschitz-Hessian then we have:

$$F(x) \le F(y) + (x-y)^\top A^\top \alpha_y + (x-y)^\top A^\top H_y A(x-y) \\ + \gamma(x) + \frac{C}{6}\|x-y\|_2^3. \tag{5}$$

For simplicity, we also assume that (5) holds componentwise with Lipschitz Hessian constant C_i for $i \in [n]$. Adding the second order approximation of $F(x)$ to $\lambda\|x-y\|_2^2$ gives the approximate function $\tilde{F}_{\lambda,y}(x)$ as defined in (3). Theorem 9 shows how to bound the sensitivities of $\tilde{F}_{\lambda,y}(x)$. Using (5) we prove a bound on the local sensitivities of $F_{\lambda,y}(x)$ itself in Appendix B:

Theorem 10. *Consider $F_{\lambda,y}$ as in Defs. 1, 8, $y \in \mathcal{X}$, a radius r, and $\alpha = \min_{x\in B(r,y)} F_{\lambda,y}(x)$. Then, $\forall\, i \in [n]$,*

$$\sigma_{F_{\lambda,y},B(r,y)}(a_i) \le \sigma_{\tilde{F}_{\lambda,y},B(r,y)}(a_i) + \min\left(\frac{C_i r}{6n\lambda}, \frac{C_i r^3}{6n\alpha}\right).$$

Using this sensitivity bound, we can independently sample components with the computed scores as in Definition 2, obtaining a $(1+\epsilon)$ approximation of the function $F_{\lambda,y}(x)$. That is, letting $F^s_{\lambda,y}(x)$ represent the subsampled empirical loss function (sampled as in Theorem 6), for $\tilde{O}\left(\frac{\Delta}{\epsilon^2}\right)$ samples, we have $F^s_{\lambda,y}(x) \in (1\pm\epsilon)F_{\lambda,y}(x)\ \forall\, x \in B(y,R)$ with high probability.

3 Optimization via Local Sensitivity Sampling

In Theorem 10 we showed how to bound the local sensitivities of a function $F := \sum_{i=1}^n f_i(a_i^T x) + \gamma(x)$ using the local sensitivities of a quadratic approximation to F, which are given by the leverage scores of an appropriate matrix (Theorem 9). These sensitivities are only valid in a sufficiently small ball around some starting point y, roughly, where the quadratic approximation is accurate. In this section we show how they can be used to optimize F beyond this ball, specifically as part of an iterative method that locally optimizes F until convergence to a global optimum.

In the optimization literature, there are two popular techniques that iteratively optimize a function via local optimizations over a ball: (i) trust region methods [Conn et al., 2000] and (ii) proximal point methods [Parikh et al., 2014]. Local sensitivity sampling can be combined with both of these classes of methods. We first focus on proximal point methods, discussing a related trust region approach in Section 5. In the proximal point method, the idea is in each step to approximate a regularized minimum:

$$x^\star_{\lambda_t,y} = \arg\min F_{\lambda_t,y}(x) = \arg\min \left[F(x) + \lambda_t\|x-y\|_2^2\right] \\ \text{and } F^\star_{\lambda_t,y} = F_{\lambda_t,y}(x^\star_{\lambda_t,y}). \tag{6}$$

Here λ_t is a regularization parameter depending on the iteration t. As discussed below, minimizing this regularized function is equivalent to minimizing F on a ball of a given radius.

3.1 Equivalence between Constrained and Penalized Formulation

When F is convex it is well known that for any λ minimizing the proximal function $F_{\lambda,y}$ is equivalent to minimizing F constrained to some ball around y. Consider the constrained optimization problem given in equation (7) where $B(r,y)$ is the ball of radius r centered at y:

$$x^\star_{r,y} = \arg\min_{x\in B(r,y)} F(x). \tag{7}$$

Lemma 11. *Let $x^\star = \arg\min_{x\in\mathbb{R}^d} F(x)$ for a convex function F. If $x^\star$ does not lie inside $B(r,y)$ then $x^\star_{r,y}$ also solves the following optimization problem:*

$$x^\star_{r,y} = \arg\min_{x\in\mathbb{R}^d}\ F(x) + \frac{\|\nabla F(x^\star_{r,y})\|}{2r}\cdot\|x-y\|_2^2. \tag{8}$$

Comparing equations (6) and (8), se see that $\lambda = \frac{\|\nabla F(x^\star_{r,y})\|}{2r} \Rightarrow r = \frac{\|\nabla F(x^\star_{r,y})\|}{2\lambda}$. While it is not directly possible to compute radius r in closed form without computing $x^\star_{r,y}$ itself, we can give a computable upper bound on r which will be crucial for our analysis.

Lemma 12. *Consider the optimization problem (6) and its corresponding constrained counterpart (7) where F is a μ strongly convex function. Then, $x^\star_{\lambda,y}$ falls within a ball of radius $r = \frac{\|\nabla F(y)\|}{2\lambda+\mu}$ around y.*

Proofs for this sections are provided in the Appendix C.

Using the local sensitivity bound of Section 2.2 we can approximate $F_{\lambda,y}$ on a ball of small enough radius.

In applying sensitivity sampling to a proximal point method, it will be critical to ensure that λ_t is not too small. This will ensure that, by Lemma 12, $x^\star_{\lambda_y}$ falls in a sufficiently small radius, and so an approximate minimum can be found via local sensitivity sampling.

3.2 Algorithmic Intuition

By Theorem 6 if we subsample the proximal function $F_{\lambda_t,y}$ using the local sensitivity bound of Theorem 10 for a sufficiently large radius r (as a function of λ_t via Lemma 12), optimizing this function will return a value within a $1+\epsilon$ factor of the true minimum $x^\star_{\lambda_t,y}$ with high probability. Abstracting away the sensitivity sampling technique, our goal becomes to analyze the convergence of the approximate proximal point method (APPM) when the optimum is computed up to $1+\epsilon$ error in each iteration. We give pseudocode for this general method in Algorithm 1.

Algorithm 1 APPM

```
input $x_0 \in \mathbb{R}^d$, $\lambda_t > 0\ \forall t \in [T]$.
input Black-box $\epsilon$-oracle $\mathcal{P}_{F_{\lambda_1,x_0}}$
for $t = 1 \ldots$ T do
    $x_t \leftarrow P_{F_{\lambda_t,x_{t-1}}}(x)$
end for
output $x_T$
```

Definition 13. An algorithm $\mathcal{P}_f$ is called *multiplicative* ϵ-oracle for a given function F if $F(x^\star) \leq F\big(\mathcal{P}_F(x)\big) \leq (1+\epsilon)F(x^\star)$ where $x^\star$ if the true minimizer of F.

In Algorithm 1, we provide the pseudocode for APPM under the access of a *multiplicative* ϵ-oracle at each iterate. In our setting, $\mathcal{P}_F$ employs local sensitivity sampling.

4 Convergence Analysis for Smooth Convex Functions

In this section, we analyze the convergence of Algorithm 1 with an ϵ oracle obtained via local sensitivity sampling. We demonstrate how to set the regularization parameters λ_t in each step and then in the end provide a complete algorithm. Let $F^\star$ denote $F(x^\star)$. Throughout we make the following assumption about $F(x)$:

- F is μ-strongly convex, *i.e.*, for all $x, y \in \mathbb{R}^d$, $F(y) \geq F(x) + \langle \nabla F(x), y-x\rangle + \frac{\mu}{2}\|y-x\|_2^2$.

4.1 Approximate Proximal Point Method with Multiplicative Oracle

We first state convergence bounds for Approximate Proximal Point Method (Algorithm 1) with a black-box multiplicative ϵ-oracle. Our first bound assumes strong convexity, our second does not. Proofs are given in Appendix D.

Theorem 14. *For μ-strongly convex F, consider $\epsilon_1, \ldots \epsilon_T \in (0,1)$ and $x_0, \ldots, x_T \in \mathbb{R}^d$ such that $x_t = P_{F_{\lambda_t,x_{t-1}}}(x_{t-1})$ where $P_{F_{\lambda_t,x_{t-1}}}$ is an ϵ_t-oracle (see Algorithm 1). Then if $\epsilon_t \leq \frac{\mu}{\mu+\lambda_t}\forall t \in [T]$, we have $F(x_t) - F^\star \leq \frac{1}{1-\epsilon_t}\frac{\lambda_t}{\mu+\lambda_t}\left(F(x_{t-1}) - F^\star\right) + \frac{\epsilon_t}{1-\epsilon_t}F^\star\ \forall t \in [T]$ and*

$$F(x_T) - F^\star \leq \rho(F(x_0) - F^\star) + \delta F^\star$$

where $\rho = \prod_{t=1}^{T}\frac{1}{1-\epsilon_t}\frac{\lambda_t}{\mu+\lambda_t}$ and $\delta = \sum_{t=1}^{T}\left(\frac{\epsilon_t}{1-\epsilon_t}\prod_{j=t+1}^{T}\frac{1}{1-\epsilon_t}\frac{\lambda_j}{\mu+\lambda_j}\right)$.

Theorem 15. *For a smooth convex function F, let $\epsilon_1, \ldots, \epsilon_T = \epsilon$* where $\epsilon \in (0, 1/2)$ *and $x_0, \ldots, x_T \in \mathbb{R}^d$ be as in Theorem 14. Then, we have*

$$F(x_T) - F^\star \leq \frac{2}{(1-\epsilon)}\frac{\|x^\star - x_0\|_2^2}{\sum_{t=1}^{T}\frac{2}{\lambda_t}} + \frac{3\epsilon}{1-\epsilon}F^\star.$$

4.2 Local Sensitivity Sampling

We now discuss how to choose the parameters for Algorithm 1 when using local sensitivity sampling to implement the ϵ-oracle in each step. From Lemmas 11 and 12 it is clear that if λ_t goes down, the corresponding radius r_t goes up. However, in Theorem 10, we bound the true local sensitivity at iteration t by a quantity depending on $\frac{r_t}{\lambda_t}$, which comes from the error in the quadratic approximation. Thus, if we choose λ_t very small, the term $\frac{r_t}{\lambda_t}$ will dominate in the local sensitivity approximation, and we won't see any advantage from local sensitivity sampling over, e.g., uniform sampling. Making λ_t large will improve the local sensitivity approximation but slow down convergence.

To balance these factors, we will choose λ_t of the order of r_t. In particular, considering Lemma 12, we choose $\lambda_t = \sqrt{\|\nabla F(x_{t-1})\|}_2$. The lemma then gives that $r_t \leq \frac{\|\nabla F(x_{t-1})\|}{\sqrt{\|\nabla F(x_{t-1})\|+\mu}} \leq \sqrt{\|\nabla F(x_{t-1})\|}_2$. We here now provide an end to end algorithm which utilizes local sensitivity sampling in the approximate proximal point method framework presented in Algorithm 1. The pseudo-code and details of the algorithm are given in Algorithm 2 where we denote $F^s_{\lambda_t,x_{t-1}}(x)$ as the importance sampled subset of $F_{\lambda_t,x_{t-1}}(x)$ which has been obtained via local sensitivity sampling. Line 9 of Algorithm 2 can be considered as a black-optimization problem which is apparently a strongly-convex optimization problem and can be optimized exponentially fast.

On Convergence: With this choice of λ_t, the convergence rate of APPM under our strong convexity assumption will be $\mathcal{O}\left(\frac{\|\sqrt{\tilde{\nabla} F(x)}\|_2}{\mu}\log(1/\varepsilon)\right)$ where $\sqrt{\|\tilde{\nabla} F(x)\|_2}$ represents $\frac{1}{T}\sum_{i=0}^{T-1}\sqrt{\|\nabla F(x_i)\|_2}$. If F is smooth with smoothness parameter L, we have: $\|\nabla F(x)\|_2 \leq L\|x - x^\star\|_2$. For the smooth but non-strongly convex problem, if we assume $\lambda_t \leq \epsilon$ for some ϵ for all t then, $\|\nabla F(x_t)\|_2^2 \in \mathcal{O}(1/T)$ in the worst case. Hence, the rate of for non-strongly convex smooth function will behave like $\mathcal{O}(1/T^{5/4})$.

Algorithm 2 APPM with Local Sensitivity Sampling

```
 1: **input** $x_0 \in \mathbb{R}^d$, $\epsilon_t$, and $\mu$.
 2: Compute $\|\nabla F(x_0)\|_2$, $F(x_0)$, and $C_0$
 3: **for** $t = 1 \ldots$ T **do**
 4:     Compute regularizer $\lambda_t \leftarrow \sqrt{\|\nabla f(x_{t-1})\|_2}$.
 5:     Compute radius $r_t \leftarrow \frac{\|\nabla f(x_{t-1})\|_2}{\sqrt{\|\nabla f(x_{t-1})\|_2}+\mu}$.
 6:     Get $\tilde{F}_{\lambda_t, x_{t-1}}$ via Taylor Expansion.
 7:     Compute the local sensitivity for $F_{\lambda_t, x_{t-1}}$ using Theorem 10.
 8:     Local sensitivity based sampling of $F^s_{\lambda_t, x_{t-1}}(x)$ from $F_{\lambda_t, x_{t-1}}(x)$.
 9:     $x_t \leftarrow \arg\min_{x \in B(r_t, x_{t-1})} F^s_{\lambda_t, x_{t-1}}(x)$.
10:     Compute $\|\nabla F(x_t)\|_2$.
11: **end for**
12: **output** $x_T$
```

5 An Adaptive Stochastic Trust Region Method

Related to the proximal point approach, sensitivity sampling can be used to obtain an adaptive stochastic trust region. In each iteration t, we approximately minimize a quadratic approximation to F over a ball, using local sensitivity sampling and directly applying the sensitivity score bound of Theorem 9. At iteration t the center of the ball is at x_{t-1} and the radius is set to $r_t = \frac{\|\nabla F(x_{t-1})\|_2}{\lambda_t + \mu}$. We provide pseudocode in Algorithm 4 and a proof of a convergence bound in Appendix E. Here we just state the main result.

Theorem 16. *For a given set of constants C_k, $\delta_k \in (0,1)$, and $\tilde{\epsilon}_k = \delta_k \frac{\mu}{\lambda_k + \mu}$ which is an error tolerance for the quadratic approximation of the function $F_{\lambda_k, x_{k-1}}(x)$ for all k, if λ_{k+1} is chosen of $\mathcal{O}(\sqrt{\|\nabla F(x_k)\|_2})$ then at iteration $k+1$ Algorithm 4 satisfies:*

$$F(x_{k+1}) - F^\star \leq (1 + 2\epsilon_{k+1})\frac{2\lambda_{k+1}}{2\lambda_{k+1} + \mu}\left(F(x_k) - F^\star\right) + 2\epsilon_{k+1}F^\star, \tag{9}$$

where $\epsilon_{k+1} = 2\tilde{\epsilon}_{k+1}\left(1 + \frac{1}{m}\right)$, m and c are positive constants.

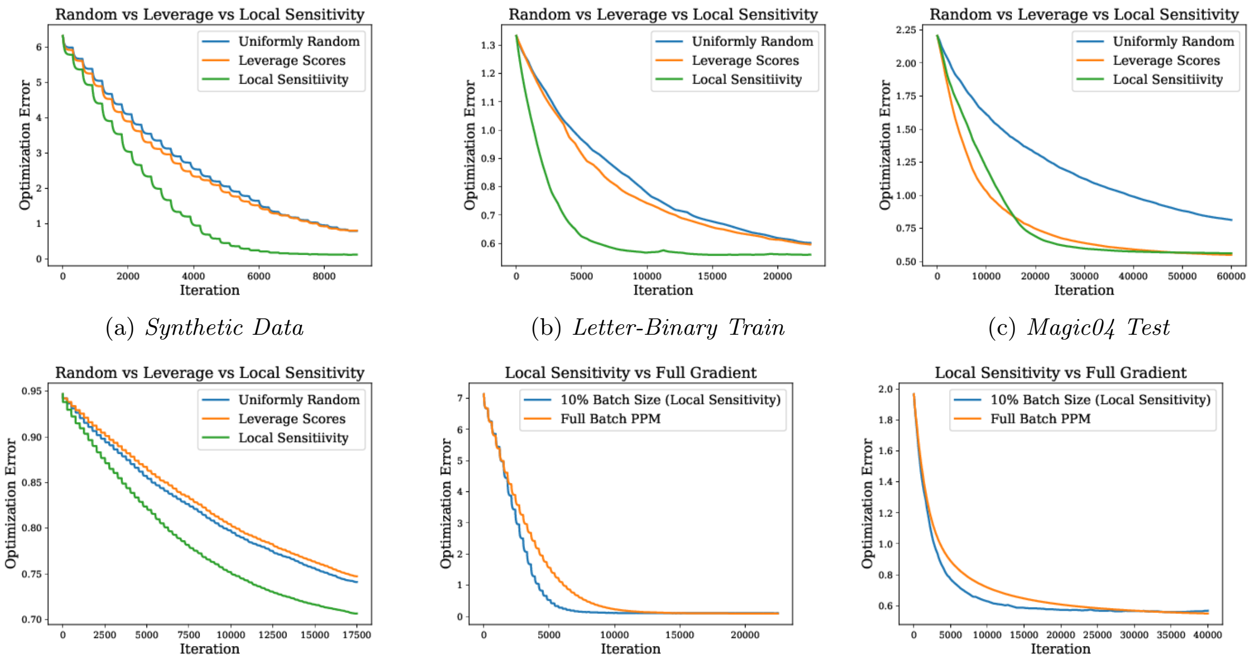
Comparing equation (9) in Theorem 16 with the bound in Theorem 14, we can see that we have obtained a similar recursive relation in both equations, and hence the trust region method will have a similar convergence rate to APPM in the presence of an ϵ-*multiplicative* oracle.

6 Experiments

We conclude by giving some initial experimental evidence to justify the performance of our proposed algorithm in practice. We provide the experiments for *Approximate Proximal Point Method with Local Sensitivity Sampling* (Algorithm 2). We run our algorithm on the following four datatsets[1] : (a) *Synthetic Data* (b) *Letter Binary* [Frey and Slate, 1991] (c) *Magic04* [Bock et al., 2004] and (d) *MNIST Binary* [LeCun et al., 1998]. Prefix 'Train' or 'Test' denotes if the train or test split was used for the experiment. The *Synthetic Data* was generated by first generating a matrix A of size 3000×300 drawn from a 300 dimensional standard normal random variable. Then another vector x_0 of size 300 was fixed which is also drawn from a normal random variable to obtain $\hat{y} = Ax_0 + \eta$ where $\eta \sim 0.1 * \mathcal{N}(0,1)$. Finally, the classification label vector y was chosen as $\text{sign}(\hat{y})$. We perform all our experiments for logistic regression with an ℓ_2^2 regularization parameter of 0.001. For the experiments plotted in the Figure 2, we have considered a fixed sample size of 100 data points for every iteration of the proximal algorithm. In the first four subfigures of Figure 2, we compare compare local sensitivity sampling with two base lines: uniform random sampling and sampling using the leverage scores of the data matrix A. On the horizontal axis, we report the total number of iterations which is the number of times the sampling oracle is called (outer loop in Algorithm 2) multiplied by number of times the gradient call to solve the optimization problem given in Line 9 in Algorithm 2. We report the optimization error on vertical axis.

From the plots in Figures 2a, 2b, 2c and 2d, it is evident that our method outperforms uniform random sampling with a large margin on the synthetic and real datasets. It also often performs much better than

[1] Datasets can be downloaded from: manikvarma.org/code/LDKL/download.html.



(a) *Synthetic Data*

(b) *Letter-Binary Train*

(c) *Magic04 Test*

(d) *MNIST Test*

(e) *Synthetic Data*

(f) *Letter-Binary Test*

Figure 2: (a-d) Local sensitivity sampling vs. uniform random sampling and leverage score sampling on four datasets: (a) *Synthetic Data* (3000 points), (b) *Letter Binary Train* (12000 points), (c) *Magic04 Test* (4795 points), and (d) *MNIST Test* (10000 points). (e-f) Local Sampling Method is compared with Full Batch Gradient for (e) *Synthetic* and (f) *Letter Binary Test*.

leverage score sampling. Since the local sensitively approximations of Theorems 9 and 10 are the leverage scores of a matrix with essentially the same dimensions as A, these methods have the same order of computational cost.

We perform a second set of experiments to compare our sampling technique with full batch gradient iteration for each proximal point iteration on *Synthetic* and *Letter Binary Test* which we plot in Figures 2e and 2f. We can see in Figures 2e and 2f that our sampling method outperforms the full gradient just with 10% of total points. In both plots, the sampling method needs just half of the number of iterations taken by full gradient to saturate to similar value.

In both of the experiments, we set the number of inner loop iteration (number of calls to the gradient oracle for solving Line 9 in Algorithm 2) in advance to let the optimization error saturate for that particular outer loop; however the plots demonstrate that it can be set to a much smaller number or can be set adaptively to achieve gains of multiple folds.

7 Conclusion

In this work, we study how the elegant approach of function approximation via sensitivity sampling can be made practical. We overcome two barriers: (1) the difficulty of approximating the sensitivity scores and (2) the high sample complexities required by theoretical bounds. We handle both by considering a *local* notion of sensitivity, which we can efficiently approximate and bound. We demonstrate that this notion can be combined with methods that globally optimize a function via iterative local optimizations, including proximal point and trust region methods.

Our work leaves open a number of questions. Most importantly, since local sensitivity approximation incurs some computational overhead (a leverage score computation along with some derivative computations), we believe it will be especially useful for functions that are difficult to optimize, e.g., non-strongly-convex functions. Understanding how our theory extends and how our method performs in practice on such functions would be very interesting. It would be especially interesting to compare performance to related approaches, such as quasi-Newton and other trust region approaches.

References

N. Agarwal, S. Kakade, R. Kidambi, Y. T. Lee, P. Netrapalli, and A. Sidford. Leverage score sampling for faster accelerated regression and ERM. *arXiv:1711.08426*, 2017.

A. Alaoui and M. W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, pages 775–783, 2015.

J. Altschuler, F. Bach, A. Rudi, and J. Weed. Massively scalable sinkhorn distances via the nystr\" om method. *arXiv preprint arXiv:1812.05189*, 2018.

H. Avron, M. Kapralov, C. Musco, C. Musco, A. Velingker, and A. Zandieh. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 253–262, 2017.

O. Bachem, M. Lucic, and A. Krause. Coresets for nonparametric estimation-the case of DP-means. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

R. Bock, A. Chilingarian, M. Gaug, F. Hakl, T. Hengstebeck, M. Jiřina, J. Klaschka, E. Kotrč, P. Savickỳ, S. Towers, et al. Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 516(2-3):511–528, 2004.

R. Chen, M. Menickelly, and K. Scheinberg. Stochastic optimization using a trust-region method and random models. *Mathematical Programming*, 169(2): 447–487, 2018.

A. Chowdhury, J. Yang, and P. Drineas. An iterative, sketching-based framework for ridge regression. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 988–997, 2018.

K. L. Clarkson and D. P. Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):54, 2017.

M. B. Cohen, Y. T. Lee, C. Musco, C. Musco, R. Peng, and A. Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2015.

M. B. Cohen, C. Musco, and C. Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1758–1777. SIAM, 2017.

A. R. Conn, N. I. Gould, and P. L. Toint. *Trust region methods*, volume 1. SIAM, 2000.

J. E. Dennis, Jr and J. J. Moré. Quasi-Newton methods, motivation and theory. *SIAM review*, 19(1): 46–89, 1977.

P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.

P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(December):3475–3506, 2012.

D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 569–578, 2011.

P. W. Frey and D. J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine learning*, 6(2):161–182, 1991.

R. Frostig, R. Ge, S. Kakade, and A. Sidford. Unregularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2540–2548, 2015.

J. Huggins, T. Campbell, and T. Broderick. Coresets for scalable bayesian logistic regression. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, 2016.

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

Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

M. Lucic, O. Bachem, and A. Krause. Strong coresets for hard and soft Bregman clustering with applications to exponential family mixtures. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

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

N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

A. Rudi, D. Calandriello, L. Carratino, and L. Rosasco. On fast leverage score sampling and optimal learning. In *Advances in Neural Information Processing Systems*, pages 5672–5682, 2018.

D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

R. Tichatschke. Proximal point methods for variational problems, 2011.

P. Xu, J. Yang, F. Roosta-Khorasani, C. Ré, and M. W. Mahoney. Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, 2016.

H. Ye, L. Luo, and Z. Zhang. Approximate Newton methods and their local convergence. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.