Choosing the Sample with Lowest Loss makes SGD Robust

Vatsal Shah
vatsalshah1106@utexas.edu
UT Austin

Xiaoxia Wu
xwu@math.utexas.edu
UT Austin

Sujay Sanghavi
sanghavi@mail.utexas.edu
UT Austin

Abstract

The presence of outliers can potentially significantly skew the parameters of machine learning models trained via stochastic gradient descent (SGD). In this paper we propose a simple variant of the SGD method: in each step, first choose a set of k samples, then from these choose the one with the smallest current loss, and do an SGD-like update with this chosen sample. Vanilla SGD corresponds to $k = 1$, i.e. no choice; $k \geq 2$ represents a new algorithm that is however effectively minimizing a non-convex surrogate loss. Our main contribution is a theoretical analysis of the robustness properties of this idea for machine learning problems which are sums of convex losses; these are backed up with synthetic and neural network experiments.

1 Introduction

This paper focuses on machine learning problems that can be formulated as optimizing the sum of n convex loss functions:

$$\min_{\boldsymbol{w}} \ F(\boldsymbol{w}) \tag{1}$$

where $F(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w})$ is the sum of convex, continuously differentiable loss functions.

Stochastic gradient descent (SGD) is a popular way to solve such problems when n is large; the simplest SGD update is:

$$\text{SGD: } \boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \eta_t \nabla f_{i_t}(\boldsymbol{w_t}) \tag{2}$$

where the sample i_t is typically chosen uniformly at random from [n].

However, as is well known, the performance of SGD and most other stochastic optimization methods is highly sensitive to the quality of the available training data. A small fraction of outliers can cause SGD to converge far away from the true optimum. While there has been a significant amount of work on more robust algorithms for special problem classes (e.g. linear regression, PCA etc.) in this paper our objective is to make a modification to the basic SGD method itself; one that can be easily applied to the many settings where vanilla SGD is already used in the training of machine learning models.

We call our method Min-k Loss SGD (MKL-SGD)[1], given below. In each iteration, we first choose a set of k samples and then select the sample with the smallest current loss in that set; this sample is then used for the update step.

Algorithm 1 MKL-SGD

```
Initialize $\boldsymbol{w_0}$
Given samples $D = (\boldsymbol{x_t}, y_t)_{t=1}^{\infty}$
for $t = 1, \ldots$ do
    Choose a set $S_t$ of $k$ samples
    Select $i_t = \arg\min_{i \in S_t} f_i(\boldsymbol{w_t})$
    Update $\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \eta \nabla f_{i_t}(\boldsymbol{w_t})$
end for
Return $\boldsymbol{w_t}$
```

The effectiveness of our algorithm relies on a simple observation: *in a situation where most samples adhere to a model but a few are outliers skewing the output, the outlier points that contribute the most to the skew are often those with high loss.* In this paper, our focus is on the stochastic setting for standard convex functions. We show that it provides a certain degree of robustness against outliers/bad training samples that may otherwise skew the estimate.

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

[1] Code: `https://github.com/vatsal2020/mkl`

Our Contributions

- To keep the analysis simple yet insightful, we define three and natural *deterministic* problem settings - noiseless with no outliers, noiseless with outliers, and noisy with outliers - in which we study the performance of MKL-SGD . In all of these settings the individual losses are assumed to be convex, and the overall loss is additionally strongly convex. We are interested in finding the optimum $\boldsymbol{w}^*$ of the "good" samples, but we do not a-priori know which samples are good and which are outliers.
- The expected MKL-SGD update (over the randomness of sample choice) is *not* the gradient of the original loss function (as would have been the case with vanilla SGD); it is instead the gradient of a different non-convex surrogate loss, even for the simplest and friendliest setting of noiseless with no outliers. Our first result establishes that this non-convexity however does not yield any bad local minima or fixed points for MKL-SGD in this particular setting, ensuring its success.
- We next turn to the setting of noiseless with outliers, where the surrogate loss can now potentially have many spurious local minima. We show that by picking a value of k high enough (depending on a condition number of the loss functions that we define) the local minima of MKL-SGD closest to $\boldsymbol{w}^*$ is better than the (unique) fixed point of SGD.
- We establish the convergence rates of MKL-SGD - with and without outliers - for both the noiseless and noisy settings .
- We back up our theoretical results with both synthetic linear regression experiments that provide insight, as well as encouraging results on the MNIST and CIFAR-10 datasets.

2 Related Work

The related work can be divided into the following four main subparts:

Stochastic optimization and weighted sampling The proposed MKL-SGD algorithm inherently implements a weighted sampling strategy to pick samples. Weighted sampling is one of the popular variants of SGD that can be used for matching one distribution to another (importance sampling), improving the rate of convergence, variance reduction or all of them has been considered in [Kahn and Marshall, 1953, Strohmer and Vershynin, 2009, Zhao and Zhang, 2015, Katharopoulos and Fleuret, 2018]. Other popular weighted sampling techniques include [Needell et al., 2014, Moulines and Bach, 2011, Lee and Sidford, 2013]. Without the assumption of strong convexity for each $f_i(.)$, the weighted sampling techniques often lead to biased estimators which are difficult to analyze. Another idea that is analogous to weighted sampling includes boosting [Freund et al., 1999] where harder samples are used to train subsequent classifiers. *However, in presence of outliers and label noise, learning the hard samples may often lead to over-fitting the solution to these bad samples.* This serves as a motivation for picking samples with the lowest loss in MKL-SGD .

Robust linear regression Learning with bad training samples is challenging and often intractable even for simple convex optimization problems. For example, OLS is quite susceptible to arbitrary corruptions by even a small fraction of outliers. Least Median Squares (LMS) and least trimmed squares (LTS) estimator proposed in [Rousseeuw, 1984, Víšek et al., 2002, Víšek, 2006] are both sample efficient, have a relatively high break-down point, but require exponential running time to converge. [Huber, 2011] provides a detailed survey on some of these robust estimators for OLS problem. Recently, [Bhatia et al., 2015, 2017, Shen and Sanghavi, 2019] have proposed robust learning algorithms for linear regression which require the computation of gradient over the entire dataset. In this version, our focus is on stochastic optimization in presence of outliers.

Robust optimization Robust optimization has received a renewed impetus following the works in [Diakonikolas et al., 2019, Lai et al., 2016, Charikar et al., 2017, Awasthi et al., 2014]. In most modern machine learning problems, however, simultaneous access to gradients over the entire dataset is time consuming and often, infeasible. [Diakonikolas et al., 2018, Prasad et al., 2018] provides robust meta-algorithms for stochastic optimization under adversarial corruptions. However, both these algorithms require the computation of one or many principal components per epoch which requires atleast $O(p^2)$ computation ([Anaraki and Hughes, 2014]). In contrast, MKL-SGD algorithm runs in $O(k)$ computations per iteration where k is the number of loss evaluations per epoch. In this paper, we don't consider the adversarial model, our focus is on the simpler corruption model where we consider outliers as defined in the next section.

Label noise in deep learning [Angluin and Laird, 1988, Kumar et al., 2010, Bengio et al., 2009] describe different techniques to learn in presence of label noise and outliers. [Rolnick et al., 2017] showed that deep neural networks are robust to random label noise especially for datasets like MNIST and CIFAR10. [Jiang et al., 2017, Ren et al., 2018] propose optimization methods based on re-weighting samples that often require significant pre-processing. In this paper, our aim is to propose a computationally inexpensive optimization approach that can also provide a certain degree of robustness.

3 Problem Setup

We make the following assumptions about our problem setting (1). Let $\mathbb{O}$ be the set of outlier samples; this set is unknown to the algorithm. We denote the optimum of the non-outlier samples by $\boldsymbol{w}^*$, i.e.

$$\boldsymbol{w}^* \ := \ \arg\min_{\boldsymbol{w}} \sum_{i \notin \mathbb{O}} f_i(\boldsymbol{w})$$

In this paper we show that MKL-SGD allows us to estimate $\boldsymbol{w}^*$ without a-priori knowledge of the set $\mathbb{O}$, under certain conditions.

Assumption 1 (**Individual losses**). *Each $f_i(\boldsymbol{w})$ is convex in $\boldsymbol{w}$, with Lipschitz continuous gradients with constant L_i.*

$$\|\nabla f_i(\boldsymbol{w_1}) - \nabla f_i(\boldsymbol{w_2})\| \leq L_i \|\boldsymbol{w_1} - \boldsymbol{w_2}\|$$

Define $L := max_i L_i$. It is common to also assume strong convexity of the overall loss function $F(\cdot)$. Here, since we are dropping samples, we need a slightly stronger assumption.

Assumption 2 (**Overall loss**). *For any $n-k$ size subset S of the samples, we assume the loss function $\sum_{i \in S} f_i(\boldsymbol{w})$ is* strongly convex *in $\boldsymbol{w}$. Recall k is the size of the sample set in the MKL-SGD algorithm.*

Assumption 3 (**Equal minimum values**). *Each of the functions $f_i(.)$ shares the same minimum value* $\min_{\boldsymbol{w}} f_i(\boldsymbol{w}) = \min_{\boldsymbol{w}} f_j(\boldsymbol{w}) \ \forall \ i, j$.

Assumption 3 is often satisfied by most standard loss functions such as squared loss, hinge loss, etc. We are now in a position to define three problem settings we will consider in this paper. For each i let $C_i := \{\hat{\boldsymbol{w}} : \hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} f_i(\boldsymbol{w})\}$ denote the set of optimal solutions (there is more than one since $f_i(\cdot)$ is convex but not strongly convex). Denote $d(a, S)$ as the shortest distance between point a and set S.

Noiseless setting with no outliers: As a first step and sanity check, we consider what happens in the easiest case: where there are no outliers. There is also no "noise", by which we mean that the optimum $\boldsymbol{w}^*$ we seek is also in the optimal set of every one of the individual sample losses, i.e.

$$\boldsymbol{w}^* \in C_i \text{ for all } i.$$

In this case, vanilla SGD (and many other methods) will converge to $\boldsymbol{w}^*$ as well; we just study this setting as a first step and also to build insight.

Outlier setting: Finally, we consider the case where a subset $\mathbb{O}$ of the samples are outliers. Specifically, we assume that for outlier samples the $\boldsymbol{w}^*$ we seek lies far from their optimal sets, while for the others it is in the optimal sets:

$$\begin{aligned} d(\boldsymbol{w}^*, C_i) &\geq 2\delta \text{ for all } i \in \mathbb{O} \\ \boldsymbol{w}^* &\in C_i \text{ for all } i \notin \mathbb{O} \end{aligned}$$

Note that vanilla SGD on the entire loss function will *not* converge to $\boldsymbol{w}^*$.

Noisy setting: As a second step, we consider the case when samples are noisy but there are no outliers. We model noise by allowing $\boldsymbol{w}^*$ to be outside of individual optimal sets C_i, but not too far; specifically,

$$\begin{array}{rl} \textit{No outliers} & d(\boldsymbol{w}^*, C_i) \leq \ \delta \text{ for all } i \\ \textit{With outliers} & d(\boldsymbol{w}^*, C_i) \leq \delta \text{ for all } i \notin \mathbb{O} \\ & d(\boldsymbol{w}^*, C_i) > 2\delta \text{ for all } i \in \mathbb{O} \end{array}$$

For the noisy setting, we will focus on the convergence guarantees. We will show that MKL-SGD gets close to $\boldsymbol{w}^*$ in this setting; again in this case vanilla SGD will do so as well for the no outliers setting.

4 Understanding MKL-SGD

We now build some intuition for MKL-SGD with some simple notation and looking at some simple settings. Recall MKL-SGD takes k samples and then retains the one with lowest current loss; this means it is sampling non-uniformly. For any $\boldsymbol{w}$, let $m_1(\boldsymbol{w}), m_2(\boldsymbol{w}), m_3(\boldsymbol{w}), \ldots m_n(\boldsymbol{w})$ be the sorted order w.r.t. the loss at that $\boldsymbol{w}$, i.e.

$$f_{m_1(\boldsymbol{w})}(\boldsymbol{w}) \leq f_{m_2(\boldsymbol{w})}(\boldsymbol{w}) \leq \cdots \leq f_{m_n(\boldsymbol{w})}(\boldsymbol{w})$$

Recall that for a sample to be the one picked by MKL-SGD for updating $\boldsymbol{w}$, it needs to first be part of the set of k samples, and then have the lowest loss among them. A simple calculation shows that probability that the i^{th} best sample $m_i(\boldsymbol{w})$ is the one picked by MKL-SGD is given by

$$p_{m_i(\boldsymbol{w})}(\boldsymbol{w}) = \begin{cases} \dfrac{\binom{n-i}{k-1}}{\binom{n}{k}} \quad \text{without replacement} \\ \dfrac{(n-(i-1))^k - (n-i)^k}{n^k} \\ \quad\quad \text{with replacement} \end{cases} \tag{3}$$

In the rest of the paper, we will focus on the "with replacement" scenario for ease of presentation; this choice does not change our main ideas or results. With this notation, we can rewrite the expected update step of MKL-SGD as

$$\mathbb{E}[\boldsymbol{w}_+|\boldsymbol{w}] = \boldsymbol{w} - \eta \sum_i p_{m_i(\boldsymbol{w})} \nabla f_{m_i(\boldsymbol{w})}(\boldsymbol{w})$$

To simplify the notation in the rest of the paper, we relabel the above update term by defining as follows:

$$\nabla \widetilde{F}(\boldsymbol{w}) \; := \; \sum_i p_{m_i(\boldsymbol{w})} \nabla f_{m_i(\boldsymbol{w})}(\boldsymbol{w})$$

Underlying this notation is the idea that, in expectation, MKL-SGD is akin to gradient descent on a *surrogate* loss function $\tilde{F}(\cdot)$ which is different from the original loss function $F(\cdot)$; indeed if needed this surrogate loss can be found (upto a constant shift) from the above gradient. We will not do that explicitly here, but instead note that even with all our assumptions, indeed even without any outliers or noise, this surrogate loss can be non-convex. It is thus important to understand MKL-SGD in all of our settings, which is what we build to now.

4.1 Noiseless setting with no outliers

As a first step (and for the purposes of sanity check), we look at MKL-SGD in the simplest setting when there are no outliers and no noise. Recall from above that this means that $\boldsymbol{w}^*$ is in the optimal set of every single individual loss $f_i(\cdot)$. However as mentioned above, even in this case the surrogate loss can be non-convex, as seen e.g. in Figure 1 for a simple example. However, in the following lemma we show that even though the overall surrogate loss $\tilde{F}(\cdot)$ is non-convex, in this no-noise no-outlier setting it has a special property with regards to the point $\boldsymbol{w}^*$.

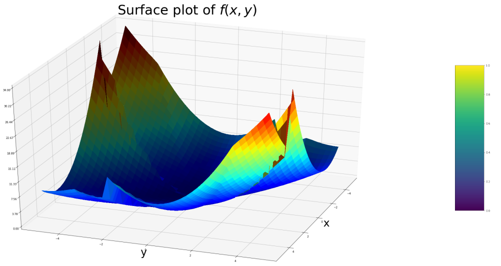

Figure 1: Non-convexity of the surface plot with three samples in the two-dimensional noiseless linear regression setting

Lemma 1. *In the noiseless setting, for any $\boldsymbol{w}$ there exists a $\lambda_{\boldsymbol{w}} > 0$ such that*

$$\nabla \widetilde{F}(\boldsymbol{w})^\top (\boldsymbol{w} - \boldsymbol{w}^*) \geq \lambda_{\boldsymbol{w}} \|\boldsymbol{w} - \boldsymbol{w}^*\|^2. \qquad (4)$$

This lemma implies that on the line between any point $\boldsymbol{w}$ and the point $\boldsymbol{w}^*$, the surrogate loss function $\tilde{F}$ is convex from any point – even though it is not convex overall. If λ_w in (4) is replaced with a constant λ, this condition is called Restricted Secant Inequality (RSI) [Karimi et al., 2016]. RSI property can imply Polyak- Lojasiewicz Inequality, which is often assumed for non-convex optimization and possibly can achieve fast convergence using SGD or its variants [Karimi et al., 2016, Lei et al., 2017, Vaswani et al., 2018, Xie et al., 2019].

We will utilize this lemma to establish our first result: in the noiseless setting with no outliers, $\boldsymbol{w}^*$ is the only fixed point (in expectation) of MKL-SGD.

Theorem 1 (**Unique stationary point**)**.** *For the noiseless setting with no outliers, and under assumptions* $1 - 3$, *the expected MKL-SGD update satisfies* $\nabla \widetilde{F}(\boldsymbol{w}) = 0$ *if and only if* $\boldsymbol{w} = \boldsymbol{w}^*$.

4.2 Outlier setting

In presence of outliers, the surrogate loss can have multiple local minima that are far from $\boldsymbol{w}^*$ and indeed potentially even worse than what we could have obtained with vanilla SGD on the original loss function. We now analyze MKL-SGD in the simple setting of scalar functions and squared losses; and then show how these results provide useful insights into the landscape of MKL-SGD loss function for standard vector settings. We would like to point out that the analysis in the next part serves as a clean template and can be extended for many other standard loss functions used in convex optimization.

Squared loss in the scalar setting: Figure 2 will be a handy tool for visualizing and understanding both the notation and results of this subsection. Consider the case where all losses are squared losses, with all the clean samples centered at w^* and all the outliers at w_B, but all having different Lipschitz constants. Specifically, consider:

$$f_i(w) = \begin{cases} l_i(w - w^*)^2 & \forall\, i \notin \mathbb{O} \\ l_i(w - w_B)^2 & \forall\, i \in \mathbb{O}, \end{cases} \qquad (5)$$

Let $l_m := \min_{i \notin \mathbb{O}} l_i$ and Let $l_M := \max_{i \in \mathbb{O}} l_i$ and $l_{max} = \max_{i \in [n]} l_i$, $l_{min} = \min_{i \in [n]} l_i$. Let us define $\kappa = \dfrac{l_{max}}{l_{min}} \geq \dfrac{l_M}{l_m}$. We initialize MKL-SGD at $w_0 = w_B$, a point where the losses of outlier samples are 0 and all the clean samples have non-zero losses. As a result at w_B, MKL-SGD has a tendency to pick all the outlier samples with a higher probability than any of the clean samples. This does not bode well for the algorithm since this implies that the final stationary point will be heavily influenced by outliers. Let $\bar{w}_{MKL}$ be the stationary point of MKL-SGD for this scalar case when initialized at w_B.

Let us define $\widetilde{w}$ as follows:

$$\widetilde{w} := \left\{ w \mid \begin{array}{l} w = \min_{\alpha} \alpha w^* + (1-\alpha) w_B, \\ \alpha \in (0,1), f_{l_m}(w) = f_{l_M}(w) \end{array} \right\} \qquad (6)$$

Thus, $\widetilde{w}$ is the closest point to w_B on the line joining w_B and w^* where the loss function of one of the clean samples and one of the outliers intersect as illustrated in Figure 2.

By observation, we know for the above scalar case $\widetilde{w} = \dfrac{\sqrt{l_m} w^* + \sqrt{l_M} w_B}{\sqrt{l_m} + \sqrt{l_M}}$. Let $\hat{p}(\boldsymbol{w}_0) = \sum_{j \in \mathbb{O}} p_j(\boldsymbol{w}_0)$ represent the total probability of picking outliers at the starting point $\boldsymbol{w}_0$. The maximum possible value attained by $\hat{p}(\boldsymbol{w}_0)$ over the entire landscape is:

$$\hat{p}_{max} = \max_{w} \hat{p}(\boldsymbol{w}) = \sum_{i=1}^{|\mathbb{O}|} p_{m_i(\boldsymbol{w})}(\boldsymbol{w}) \qquad (7)$$

where for any $\boldsymbol{w}$, $p_{m_i(\boldsymbol{w})}(\boldsymbol{w})$ are ordered i.e. $p_{m_1(\boldsymbol{w})}(\boldsymbol{w}) > p_{m_2(\boldsymbol{w})}(\boldsymbol{w}) > \cdots > p_{m_n(\boldsymbol{w})}(\boldsymbol{w})$. The next condition gives a sufficient condition to avoid all the bad local minima no matter where we initialize for the simple scalar case:

Condition 1. $\hat{p}_{max} < \dfrac{1}{1 + \kappa\sqrt{\kappa}}$

To further elaborate on this, for the loss functions and $\widetilde{w}$ defined in equations (5) and (6) respectively, if condition 1 is not satisfied, then we cannot say anything about where MKL-SGD converges. However, if condition 1 holds true, then we are in Case 1 (Figure 2), i.e. the stationary point attained by MKL-SGD will be such that it is possible to avoid the existence of the first bad local minima. The first bad local minima occurs by solving the optimization problem where the top-$|\mathbb{O}|$ highest probabilities are assigned to the bad samples.

Following the above analysis recursively, we can show that all other subsequent bad local minimas are avoided as well, until we reach the local minima which assigns the largest $(n - |\mathbb{O}|)$ probabilities to the clean samples. This indicates that irrespective of where we initialize in the $1D$ landscape, we are bound to end up at a local minima with the highest probabilities assigned to the clean samples. In the latter part of this section, we will show that MKL-SGD solution attained when Case 1 holds is provably better than the SGD solution. However, if condition 1 is false (Case 2, Figure 2), then it is possible that MKL-SGD gets stuck at any one of the many local minimas that exist close to the outlier center w_B and we cannot say anything about the relative distance from $\boldsymbol{w}^*$.

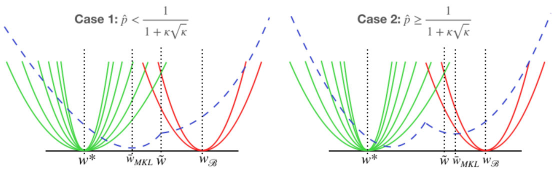

Figure 2: Illustration with conditions when bad local minima will or will not exist. Here, we demonstrate that even if we start at an initialization $\boldsymbol{w}_{\boldsymbol{\mathcal{B}}}$ that assigns the highest probabilities to bad samples (red), it is possible to avoid the existence of a bad local minima if Condition 1 is satisfied. Recursively, we show in Lemma 2 that it is possible to avoid all bad local minima and reach a good local minima (where the good samples have the highest probabilities)

A key takeaway from the above condition is that *for a fixed n as κ increases, we can tolerate smaller $\hat{p}$ and consequently smaller fraction of corruptions ϵ.* For a fixed ϵ and n, increasing the parameter k (upto $k < \frac{n}{2}$) in MKL-SGD leads to an increase in $\hat{p}$ and thus increasing k can lead to the violation of the above condition. This happens because samples with lower loss will be picked with increasing probability as k increases and as a result the propensity of MKL-SGD to converge towards the closest stationary point it encounters is higher.

For the scalar setting with squared loss and all the outliers are centered at the same point, the condition to avoid the worst MKL-SGD solution does not depend on the location of w_B. However, this is usually not the case. We will provide a detailed description of the analogous setting where outliers are centered at different points in the Appendix.

Squared loss in the vector setting The loss functions are redefined as follows:

$$f_i(\boldsymbol{w}) = \begin{cases} l_i \|\boldsymbol{w} - \boldsymbol{w}^*\|^2 & \forall\, i \notin \mathbb{O} \\ l_i \|\boldsymbol{w} - \boldsymbol{w}_{b_i}\|^2 & \forall\, i \in \mathbb{O}, \end{cases} \qquad (8)$$

Without loss of generality, assume that $2\delta < \|\boldsymbol{w}_{b_1} - \boldsymbol{w}^*\| \leq \|\boldsymbol{w}_{b_2} - \boldsymbol{w}^*\| \leq \cdots \leq \left\|\boldsymbol{w}_{b_{|\mathbb{O}|}} - \boldsymbol{w}^*\right\|$ and $\gamma = \dfrac{2\delta}{\left\|\boldsymbol{w}_{b_{|\mathbb{O}|}} - \boldsymbol{w}^*\right\|}$. Let $\bar{\boldsymbol{w}}$ be any stationary attained by MKL-SGD. Suppose $\theta_{i,\bar{\boldsymbol{w}}}$ be the angle between the line passing through $\boldsymbol{w}_{b_i}$ and $\boldsymbol{w}^*$ and the line connecting $\bar{\boldsymbol{w}}$ and $\boldsymbol{w}^*$. Let us define $\theta_{M,\bar{\boldsymbol{w}}} := \max_i \theta_{i,\bar{\boldsymbol{w}}}$ and $\kappa = \max_{i \in [n]} l_i / \min_{i \in [n]} l_i$.

At $\boldsymbol{w^*}$, by definition, we know that $\forall\ i \notin \mathbb{O}$, $f_i(\boldsymbol{w^*}) = 0$ and $\forall\ j \in \mathbb{O}$, $f_j(\boldsymbol{w^*}) > 0$. By continuity arguments, there exists a ball of radius $r > 0$ around $\boldsymbol{w^*}$, $\mathcal{B}_r(\boldsymbol{w^*})$, defined as follows:

$$\mathcal{B}_r(\boldsymbol{w^*}) = \left\{ \boldsymbol{w} \mid \begin{array}{l} f_i(\boldsymbol{w}) < f_j(\boldsymbol{w}) \ \forall\ i \notin \mathbb{O},\ j \in \mathbb{O}, \\ \|\boldsymbol{w} - \boldsymbol{w^*}\| \le r \end{array} \right\} \quad (9)$$

In the subsequent lemma, we show that that it is possible to drift into the ball $\mathcal{B}_r(\boldsymbol{w^*})$ where the clean samples have the highest probability or lowest loss[2].

Lemma 2. *Consider the loss function and $\mathcal{B}_r(\boldsymbol{w^*})$ defined in equations* (8) *and* (9) *respectively. Suppose*

$$q = \frac{\cos\theta_{M,\bar{\boldsymbol{w}}}}{\gamma} - 1 + \frac{\sqrt{\kappa}\cos\theta_{M,\bar{\boldsymbol{w}}}}{\gamma} > 0$$

and $\hat{p}_{max}$ as defined in Equation (7) *satisfies*

$$\hat{p}_{max} \le \frac{1}{1+\kappa q}.$$

Starting from any initialization $\boldsymbol{w}_0$, for any stationary point $\bar{\boldsymbol{w}}$ attained by MKL-SGD, we have that

$$\bar{\boldsymbol{w}} \in \mathcal{B}_r(\boldsymbol{w^*})$$

In other words, initializing at any point in the landscape, the final stationary point attained by MKL-SGD will inevitably assign the largest $n - |\mathbb{O}|$ probabilities to the clean samples. The proof is availabe in Appendix Section 7.2.4. For the scalar case, $d = 1$, we have $\theta_{j,\bar{\boldsymbol{w}}} = 0 \ \forall\ j$. If $\gamma = 1$ and all the outliers are centered at the same point, then in the scalar setting the condition in Lemma 2 reduces to condition 1. Note that, the above lemma leads to a very strong worst-case guarantee. It states that the farthest optimum for MKL-SGD will always be within a bowl of distance r from $\boldsymbol{w^*}$ no matter where we initialize as long as the conditions are satisfied. The proof and further discussion on other parameters in Lemma 2 is deferred to Appendix Section 7.2.4.

Analysis for the general outlier setting: In this part, we analyze the fixed point equations associated with MKL-SGD and SGD and try to understand the behavior *in a ball $\mathcal{B}_r(\boldsymbol{w^*})$ around the optimum.*

For the sake of simplicity, we will assume that $\|\nabla f_i(\boldsymbol{w})\| \le G \ \forall\ i \in \mathbb{O}$. Next, we analyze the following two quantities: i) distance of $\bar{\boldsymbol{w}}_{\boldsymbol{SGD}}$ from $\boldsymbol{w^*}$ and distance of the any of the solutions attained by $\bar{\boldsymbol{w}}_{\boldsymbol{MKL}}$ from $\boldsymbol{w^*}$.

Lemma 3. *Let $\bar{\boldsymbol{w}}_{\boldsymbol{SGD}}$ indicate the solution attained SGD. Under assumptions 1-3, there exists an ϵ' such that for all $\epsilon \le \epsilon'$,*

$$\epsilon G \le (1-\epsilon)L\|\bar{\boldsymbol{w}}_{\boldsymbol{SGD}} - \boldsymbol{w^*}\|$$

Using Lemma 1, we will define λ as follows:

$$\lambda := \min_{\boldsymbol{w}} \lambda_{\boldsymbol{w}} \quad (10)$$

Assumption 2 ensures that $\lambda > 0$, however the lower bounds for this λ are loss function dependent.

Lemma 4. *Let $\bar{\boldsymbol{w}}_{\boldsymbol{MKL}}$ be any first order stationary point attained by MKL-SGD . Under assumptions 1-3, for a given $\epsilon < 1$ and λ as defined in equation* (10)*, there exists a k' such that for all $k \ge k'$,*

$$\|\bar{\boldsymbol{w}}_{\boldsymbol{MKL}} - \boldsymbol{w^*}\| \le \epsilon^k G/\lambda$$

Finally, we show that any solution attained by MKL-SGD is provably better than the solution attained by SGD. We would like to emphasize that this is a very strong result. The MKL-SGD has numerous local minima and here we show that even the worst[3] solution attained by MKL-SGD is closer to $\boldsymbol{w^*}$ than the solution attained by SGD. Let us define $\alpha(\epsilon, L, k, \lambda) = (1-\epsilon)L\epsilon^{k-1}/\lambda$

Theorem 2. *Let $\bar{\boldsymbol{w}}_{\boldsymbol{SGD}}$ and $\bar{\boldsymbol{w}}_{\boldsymbol{MKL}}$ be the the stationary points attained by SGD and MKL-SGD algorithms respectively for the noiseless setting with outliers. Under assumptions 1-3, for any $\bar{\boldsymbol{w}}_{\boldsymbol{MKL}} \in \mathcal{B}_r(\boldsymbol{w^*})$ and λ defined in equation* (10)*, there exists an ϵ' and k' such that for all $\epsilon \le \epsilon'$ and $k \ge k'$, we have $\alpha(\epsilon, L, k, \lambda) < 1$ and,*

$$\|\bar{\boldsymbol{w}}_{\boldsymbol{MKL}} - \boldsymbol{w^*}\| < \alpha(\epsilon, L, k, \lambda)\|\bar{\boldsymbol{w}}_{\boldsymbol{SGD}} - \boldsymbol{w^*}\| \quad (11)$$

For squared loss in scalar setting, we claimed that for a fixed n and ϵ, using a large k may not be a good idea. Here, however once we are in the ball, $\mathcal{B}_r(\boldsymbol{w^*})$, using larger k (any $k < \frac{n}{2}$), reduces $\alpha(\epsilon, L, k, \lambda)$ and allows MKL-SGD to get closer to $\boldsymbol{w^*}$.

The conditions required in Lemma 2 and Theorem 2 enable us to provide guarantees for only a subset of relatively well-conditioned problems. We would like to emphasize that the bounds we obtain are worst case bounds and not in expectation. As we will note in the Section 6 and the Appendix, however these bounds may not be necessary, for convex optimization problems MKL-SGD easily outperforms SGD.

5 Convergence Rates

In this section, we go back to the in expectation convergence analysis which is standard for the stochastic settings. For smooth functions with strong convexity, [Moulines and Bach, 2011, Needell et al., 2014] provided guarantees for linear rate of convergence. We

[2] It is trivial to show the existence of a ball of radius $r > 0$ for any set of continuously differentiable $f_i(.)$.

[3] farthest solution from $\boldsymbol{w^*}$

restate the theorem here and show that the theorem still holds for the non-convex landscape obtained by MKL-SGD in noiseless setting.

Lemma 5 (**Linear Convergence** [Needell et al., 2014])**.** *Let $F(\boldsymbol{w}) = \mathbb{E}[f_i(\boldsymbol{w})]$ be λ-strongly convex. Set $\sigma^2 = \mathbb{E}[\|\nabla f_i(\boldsymbol{w^*})\|^2]$ with $\boldsymbol{w}^* := argminF(\boldsymbol{w})$. Suppose $\eta \leq \frac{1}{\sup_i L_i}$. Let $\boldsymbol{\Delta}_t = \boldsymbol{w^*} - \boldsymbol{w_t}$. After T iterations, SGD satisfies:*

$$\mathbb{E}\left[\|\boldsymbol{\Delta}_T\|^2\right] \leq (1 - 2\eta\hat{C})^T \|\boldsymbol{\Delta}_0\|^2 + \eta R_\sigma \qquad (12)$$

where $\hat{C} = \lambda(1 - \eta \sup_i L_i)$ and $R_\sigma = \sigma^2/\hat{C}$.

In the noiseless setting, we have $\|\nabla f_i(\boldsymbol{w^*})\| = 0$ and so $\sigma := 0$. $\boldsymbol{w}^*$ in (12) is the same as $\boldsymbol{w}^*$ stated in Theorem 1. Even though above theorem is for SGD, it still can be applied to our algorithm 1. At each iteration there exists a parameter $\lambda_{\boldsymbol{w_t}}$ that could be seen as the strong convexity parameter (c.f. Lemma 1). For MKL-SGD, the parameter λ in (12) should be $\lambda = \min_t \lambda_{\boldsymbol{w}_t}$. Thus, MKL-SGD algorithm still guarantees *linear convergence* result but with an implication of slower speed of convergence than SGD.

However, Lemma 5 will not hold for MKL-SGD in noisy setting since the objective is not strongly convex. Even for noiseless setting, the rate of convergence for MKL-SGD in Lemma 5 is not tight. The upper bound in (12) is loosely set to the constant $\lambda := \min_t \lambda_{\boldsymbol{w}_t}$ for all iterations. Using a per-iterate analysis, we provide a general bound for any stochastic algorithm (c.f. Theorem 3) for both noiseless and noisy setting in absence and presence of outliers.

Theorem 3 (**Distance to $\boldsymbol{w^*}$**)**.** *Let $\boldsymbol{\Delta}_t = \boldsymbol{w^*} - \boldsymbol{w_t}$. Denote the strong convexity parameter λ_{good} for all the good samples. Let*

$$\psi = 2\eta_t \lambda_{good}(1 - \eta_t \sup_i L_i) \min_{i \notin \mathbb{O}} p_i(\boldsymbol{w_t})$$

Suppose at t^{th} iteration, the stepsize is set as η_t, then conditioned on the current parameter $\boldsymbol{w}_t$, the expectation of the distance between the $\boldsymbol{w}_{t+1}$ and $\boldsymbol{w^*}$ can be upper bounded as:

$$\mathbb{E}_i\left[\|\boldsymbol{\Delta}_{t+1}\|^2 | \boldsymbol{w_t}\right] \leq (1 - \psi)\|\boldsymbol{\Delta}_t\|^2 + \eta_t R_t \qquad (13)$$

where

$$R_t = 2\sum_{i \notin \mathbb{O}} p_i(\boldsymbol{w_t})\left(\langle \boldsymbol{\Delta}_t, \nabla f_i(\boldsymbol{w^*})\rangle + \eta_t \|\nabla f_i(\boldsymbol{w^*})\|^2\right)$$
$$+ \sum_{i \in \mathbb{O}} p_i(\boldsymbol{w_t})\left(\eta_t \|\nabla f_i(\boldsymbol{w_t})\|^2 + 2\left(f_i(\boldsymbol{w^*}) - f_i(\boldsymbol{w_t})\right)\right)$$

Theorem 3 implies that for any stochastic algorithm in the both noisy and noiseless setting, the presence of outliers can make the upper bound (R_t) much worse due to an extra term (the second term in R_t). *The second term in R_t has a lower bound that could be an increasing function of $|\mathbb{O}|$*. However, its impact can be reduced by appropriately setting $p_i(\boldsymbol{w_t})$, for instance using a larger k in MKL-SGD. Corollary 1 in the Appendix also provides a sufficient condition when MKL-SGD is always better than standard SGD (in terms of its distance from $\boldsymbol{w^*}$ in expectation).

The convergence rate depends on the constant $\psi \propto \min_{i \notin \mathbb{O}} p_i(\boldsymbol{w_t})$. Note that this term $\min_{i \notin \mathbb{O}} p_i(\boldsymbol{w_t})$ is not too small for our algorithm MKL-SGD since it is a minimum amongst all *good* samples (not including the outliers). However, when compared with vanilla SGD where $\min_{i \notin \mathbb{O}} p_i(\boldsymbol{w_t}) = 1/N$, $\min_{i \notin \mathbb{O}} p_i(\boldsymbol{w_t})$ with $p_i(\boldsymbol{w_t})$ defined in (3) for MKL-SGD, in some sense, could be smaller than $1/N$.

To understand the residual term R_t. Let us take the noiseless setting with outliers for an example. We

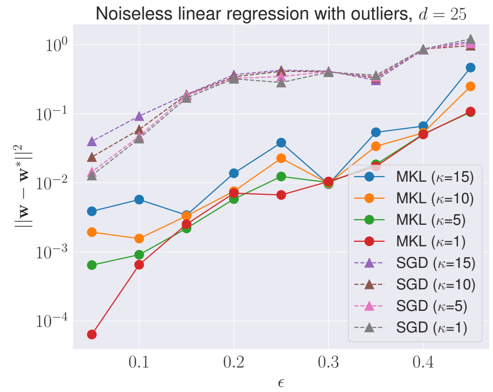

Figure 3: Comparing the performance of MKL-SGD ($k = 2$) and SGD for different values of κ in noiseless linear regression against varying fraction of outliers.

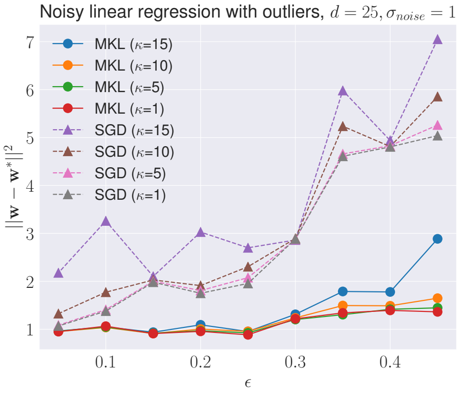

Figure 4: Comparing the performance of MKL-SGD ($k = 2$) and SGD for different values of κ in noisy linear regression against varying fraction of outliers.

Dataset	**MNIST**			**CIFAR10**		
Optimizer ϵ	**SGD**	**MKL-SGD**	**Oracle**	**SGD**	**MKL-SGD**	**Oracle**
0.1	**96.76**	96.49	98.52	79.1	**81.94**	84.56
0.2	92.54	**95.76**	98.33	72.29	**77.77**	84.40
0.3	85.77	**95.96**	98.16	63.96	**66.49**	84.66
0.4	71.95	**94.20**	97.98	52.4	**53.57**	84.42

Table 1: Comparing the test accuracy of SGD and MKL-SGD ($k = 5/3$) over MNIST and CIFAR-10 datasets in presence of corruptions via directed label noise.

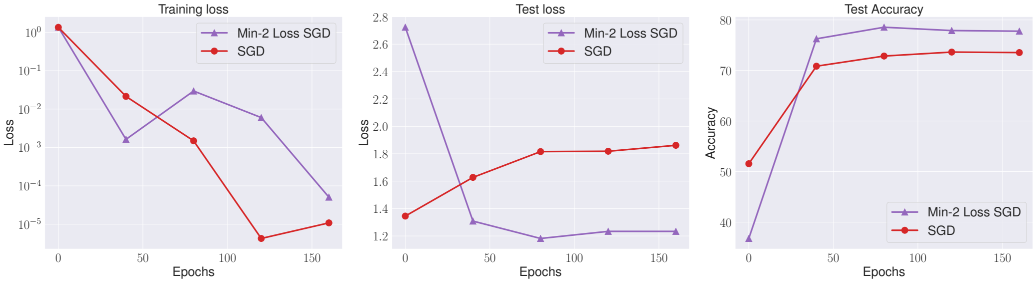

Figure 5: Comparing training loss, test loss and test accuracy of MKL-SGD and SGD. Parameters: $\epsilon = 0.2$, $k = 2$, $b = 16$. The training loss is lower for SGD which means that SGD overfits to the noisy data. The lower test loss and higher accuracy demonstrates the robustness MKL-SGD provides for corrupted data.

have $\nabla f_i(\boldsymbol{w}^*) = 0$ and $f_i(\boldsymbol{w}^*) = 0$ for all $i \notin \mathbb{O}$. But for $i \in \mathbb{O}$, $\nabla f_i(\boldsymbol{w}^*) \neq 0$ and $f_i(\boldsymbol{w}^*) \neq 0$. Then the term R_t can be reduced to

$$R_t = \sum_{i \in \mathbb{O}} p_i(\boldsymbol{w_t}) \left(\eta_t \|\nabla f_i(\boldsymbol{w_t})\|^2 + 2(f_i(\boldsymbol{w}^*) - f_i(\boldsymbol{w_t})) \right)$$

If we are at the same point $\boldsymbol{w_t}$ for both SGD and MKL-SGD and $p_i(\boldsymbol{w_t}) < 1/N$ for $i \in \mathbb{O}$, we have $R_t^{(SGD)} > R_t^{(MKL)}$. It means that MKL-SGD enjoys linear convergence, with a good speed proportional to $\min_{i \notin \mathbb{O}} p_i(\boldsymbol{w_t})$ (but not necessarily faster than vanilla SGD) up to a neighborhood of potentially smaller radius than vanilla SGD.

6 Experiments

In this section, we compare the performance of MKL-SGD and SGD for synthetic datasets for linear regression and small-scale neural networks.

6.1 Linear Regression

For simple linear regression, we assume that X_i are sampled from normal distribution with different condition numbers. $X_i \sim \mathcal{N}(0, \boldsymbol{D})$ where $\boldsymbol{D}$ is a diagonal matrix such that $D_{11} = \kappa$ and $D_{ii} = 1$ for all i). We compare the performance of MKL-SGD and SGD for different values of κ (Figs. 3 and 4) under noiseless and noisy settings against varying levels of corruption ϵ. It is important to note that different κ values correspond to different rates of convergence. To ensure fair comparison, we run the algorithms till the error values stop decaying and take the distance of $\boldsymbol{w}^*$ from the exponential moving average of the iterates.

6.2 Neural Networks

For deep learning experiments, we corrupt the data using directed noise model. In this corruption model, all the samples of class a that are in error are assigned the same wrong label b. This is a stronger corruption model than corruption by random noise. For the MKL-SGD algorithm, we run a more practical batched (size b) variant such that if $k = 2$ the algorithm picks $b/2$ samples out of b sample loss evaluations. The results in Oracle are obtained by running SGD over only non-corrupted samples.

MNIST: We train standard 2 layer convolutional network on subsampled MNIST (5000 samples with labels). We train over 80 epochs using an initial learning rate of 0.05 with the decaying schedule of factor 5 after every 30 epochs. The results of the MNIST dataset are averaged over 5 runs.

CIFAR10: We train Resnet-18 [He et al., 2016] on CIFAR-10 (50000 training samples with labels) for over 200 epochs using an initial learning rate of 0.05 with the decaying schedule of factor 5 after every 90 epochs. The reported accuracy is based on the true validation set. The results of the CIFAR-10 dataset are averaged over 3 runs.

References

Farhad Pourkamali Anaraki and Shannon Hughes. Memory and computation efficient pca via very sparse random projections. In *International Conference on Machine Learning*, pages 1341–1349, 2014.

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Pranjal Awasthi, Maria Florina Balcan, and Philip M Long. The power of localization for efficiently learning linear separators with noise. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 449–458, 2014.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. In *Advances in Neural Information Processing Systems*, pages 721–729, 2015.

Kush Bhatia, Prateek Jain, Parameswaran Kamalaruban, and Purushottam Kar. Consistent robust regression. In *Advances in Neural Information Processing Systems*, pages 2110–2119, 2017.

Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60. ACM, 2017.

Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. *arXiv preprint arXiv:1803.02815*, 2018.

Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.

Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780): 1612, 1999.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

Peter J Huber. *Robust statistics*. Springer, 2011.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 2017.

Herman Kahn and Andy W Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. *arXiv preprint arXiv:1803.00942*, 2018.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016.

Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 147–156. IEEE, 2013.

Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pages 2348–2358, 2017.

Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in neural information processing systems*, pages 1017–1025, 2014.

Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pages 5739–5748, 2019.

Thomas Strohmer and Roman Vershynin. A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262, 2009.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for overparameterized models and an accelerated perceptron. *arXiv preprint arXiv:1810.07288*, 2018.

Jan Víšek et al. The least weighted squares ii. consistency and asymptotic normality. *Bulletin of the Czech Econometric Society*, 9, 2002.

Jan Ámos Víšek. The least trimmed squares. part i: Consistency. *Kybernetika*, 42(1):1–36, 2006.

Yuege Xie, Xiaoxia Wu, and Rachel Ward. Linear convergence of adaptive stochastic gradient descent. *arXiv preprint arXiv:1908.10525*, 2019.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *international conference on machine learning*, pages 1–9, 2015.