

---

# Safe Adaptive Importance Sampling

---

Sebastian U. Stich

EPFL

sebastian.stich@epfl.ch

Anant Raj

Max Planck Institute for Intelligent Systems

anant.raj@tuebingen.mpg.de

Martin Jaggi

EPFL

martin.jaggi@epfl.ch

## Abstract

Importance sampling has become an indispensable strategy to speed up optimization algorithms for large-scale applications. Improved adaptive variants—using importance values defined by the complete gradient information which changes during optimization—enjoy favorable theoretical properties, but are typically computationally infeasible. In this paper we propose an efficient approximation of gradient-based sampling, which is based on safe bounds on the gradient. The proposed sampling distribution is (i) provably the *best sampling* with respect to the given bounds, (ii) always better than uniform sampling and fixed importance sampling and (iii) can efficiently be computed—in many applications at negligible extra cost. The proposed sampling scheme is generic and can easily be integrated into existing algorithms. In particular, we show that coordinate-descent (CD) and stochastic gradient descent (SGD) can enjoy significant a speed-up under the novel scheme. The proven efficiency of the proposed sampling is verified by extensive numerical testing.

## 1 Introduction

Modern machine learning applications operate on massive datasets. The algorithms that are used for data analysis face the difficult challenge to cope with the enormous amount of data or the vast dimensionality of the problems. A simple and well established strategy to reduce the computational costs is to split the data and to operate only on a small part of it, as for instance in coordinate descent (CD) methods and stochastic gradient (SGD) methods. These kind of methods are state of the art for a wide selection of machine learning, deep learning and signal processing applications [9, 11, 35, 27]. The application of these schemes is not only motivated by their practical performance, but also well justified by theory [18, 19, 2].

Deterministic strategies are seldom used for the data selection—examples are steepest coordinate descent [4, 34, 20] or screening algorithms [14, 15]. Instead, randomized selection has become ubiquitous, most prominently uniform sampling [27, 29, 7, 8, 28] but also non-uniform sampling based on a *fixed* distribution, commonly referred to as *importance sampling* [18, 19, 2, 33, 16, 6, 25, 24]. While these sampling strategies typically depend on the input data, they do not adapt to the information of the current parameters during optimization. In contrast, *adaptive* importance sampling strategies constantly re-evaluate the relative importance of each data point during training and thereby often surpass the performance of static algorithms [22, 5, 26, 10, 21, 23]. Common strategies are *gradient-based* sampling [22, 36, 37] (mostly for SGD) and *duality gap-based* sampling for CD [5, 23].

The drawbacks of adaptive strategies are twofold: often the provable theoretical guarantees can be worse than the complexity estimates for uniform sampling [23, 3] and often it is computationally

inadmissible to compute the optimal adaptive sampling distribution. For instance gradient based sampling requires the computation of the full gradient in each iteration [22, 36, 37]. Therefore one has to rely on approximations based on upper bounds [36, 37], or stale values [22, 1]. But in general these approximations can again be worse than uniform sampling.

This makes it necessary to develop adaptive strategies that can efficiently be computed in every iteration and that come with theoretical guarantees that show their advantage over fixed sampling.

**Our contributions.** In this paper we propose an efficient approximation of the gradient-based sampling in the sense that (i) it can efficiently be computed in every iteration, (ii) is provably better than uniform or fixed importance sampling and (iii) recovers the gradient-based sampling in the full-information setting. The scheme is completely generic and can easily be added as an improvement to both CD and SGD type methods.

As our key contributions, we

- (1) show that gradient-based sampling in CD methods is theoretically better than the classical fixed sampling, the speed-up can reach a factor of the dimension  $n$  (Section 2);
- (2) propose a generic and efficient *adaptive importance sampling* strategy that can be applied in CD and SGD methods and enjoys favorable properties—such as mentioned above (Section 3);
- (3) demonstrate how the novel scheme can efficiently be integrated in CD and SGD on an important class of structured optimization problems (Section 4);
- (4) supply numerical evidence that the novel sampling performs well on real data (Section 5).

**Notation.** For  $\mathbf{x} \in \mathbb{R}^n$  define  $[\mathbf{x}]_i := \langle \mathbf{x}, \mathbf{e}_i \rangle$  with  $\mathbf{e}_i$  the standard unit vectors in  $\mathbb{R}^n$ . We abbreviate  $\nabla_i f := [\nabla f]_i$ . A convex function  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  with  $L$ -Lipschitz continuous gradient satisfies

$$f(\mathbf{x} + \eta \mathbf{u}) \leq f(\mathbf{x}) + \eta \langle \mathbf{u}, \nabla f(\mathbf{x}) \rangle + \frac{\eta^2 L_{\mathbf{u}}}{2} \|\mathbf{u}\|^2 \quad \forall \mathbf{x} \in \mathbb{R}^n, \forall \eta \in \mathbb{R}, \quad (1)$$

for every direction  $\mathbf{u} \in \mathbb{R}^n$  and  $L_{\mathbf{u}} = L$ . A function with coordinate-wise  $L_i$ -Lipschitz continuous gradients<sup>1</sup> for constants  $L_i > 0, i \in [n] := \{1, \dots, n\}$ , satisfies (1) just along coordinate directions, i.e.  $\mathbf{u} = \mathbf{e}_i, L_{\mathbf{e}_i} = L_i$  for every  $i \in [n]$ . A function is coordinate-wise  $L$ -smooth if  $L_i \leq L$  for  $i = 1, \dots, n$ . For convenience we introduce vector  $\mathbf{l} = (L_1, \dots, L_n)^\top$  and matrix  $\mathbf{L} = \text{diag}(\mathbf{l})$ . A probability vector  $\mathbf{p} \in \Delta^n := \{\mathbf{x} \in \mathbb{R}_{\geq 0}^n: \|\mathbf{x}\|_1 = 1\}$  defines a probability distribution  $\mathcal{P}$  over  $[n]$  and we denote by  $i \sim \mathbf{p}$  a sample drawn from  $\mathcal{P}$ .

## 2 Adaptive Importance Sampling with Full Information

In this section we argue that adaptive sampling strategies are theoretically well justified, as they can lead to significant improvements over static strategies. In our exhibition we focus first on CD methods, as we also propose a novel stepsize strategy for CD in this contribution. Then we revisit the results regarding stochastic gradient descent (SGD) already present in the literature.

### 2.1 Coordinate Descent with Adaptive Importance Sampling

We address general minimization problems  $\min_{\mathbf{x}} f(\mathbf{x})$ . Let the objective  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be convex with coordinate-wise  $L_i$ -Lipschitz continuous gradients. Coordinate descent methods generate sequences  $\{\mathbf{x}_k\}_{k \geq 0}$  of iterates that satisfy the relation

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla_{i_k} f(\mathbf{x}_k) \mathbf{e}_{i_k}. \quad (2)$$

Here, the direction  $i_k$  is either chosen deterministically (cyclic descent, steepest descent), or randomly picked according to a probability vector  $\mathbf{p}_k \in \Delta^n$ . In the classical literature, the stepsize is often chosen such as to minimize the quadratic upper bound (1), i.e.  $\gamma_k = L_{i_k}^{-1}$ . In this work we propose to set  $\gamma_k = \alpha_k [\mathbf{p}_k]_{i_k}^{-1}$  where  $\alpha_k$  does not depend on the chosen direction  $i_k$ . This leads to

---

<sup>1</sup>  $|\nabla_i f(\mathbf{x} + \eta \mathbf{e}_i) - \nabla_i f(\mathbf{x})| \leq L_i |\eta|, \quad \forall \mathbf{x} \in \mathbb{R}^n, \forall \eta \in \mathbb{R}.$

directionally-unbiased updates, like it is common among SGD-type methods. It holds

$$\begin{aligned}\mathbb{E}_{i_k \sim \mathbf{p}_k} [f(\mathbf{x}_{k+1}) \mid \mathbf{x}_k] &\stackrel{(1)}{\leq} \mathbb{E}_{i_k \sim \mathbf{p}_k} \left[ f(\mathbf{x}_k) - \frac{\alpha_k}{[\mathbf{p}_k]_{i_k}} (\nabla_{i_k} f(\mathbf{x}_k))^2 + \frac{L_i \alpha_k^2}{2[\mathbf{p}_k]_{i_k}^2} (\nabla_{i_k} f(\mathbf{x}_k))^2 \mid \mathbf{x}_k \right] \\ &= f(\mathbf{x}_k) - \alpha_k \|\nabla f(\mathbf{x}_k)\|_2^2 + \sum_{i=1}^n \frac{L_i \alpha_k^2}{2[\mathbf{p}_k]_i} (\nabla_i f(\mathbf{x}_k))^2.\end{aligned}\quad (3)$$

In adaptive strategies we have the freedom to chose both variables  $\alpha_k$  and  $\mathbf{p}_k$  as we like. We therefore propose to chose them in such a way that they *minimize* the upper bound (3) in order to maximize the expected progress. The optimal  $\mathbf{p}_k$  in (3) is independent of  $\alpha_k$ , but the optimal  $\alpha_k$  depends on  $\mathbf{p}_k$ . We can state the following useful observation.

**Lemma 2.1.** *If  $\alpha_k = \alpha_k(\mathbf{p}_k)$  is the minimizer of (3), then  $\mathbf{x}_{k+1} := \mathbf{x}_k - \frac{\alpha_k}{[\mathbf{p}_k]_{i_k}} \nabla_{i_k} f(\mathbf{x}_k) \mathbf{e}_{i_k}$  satisfies*

$$\mathbb{E}_{i_k \sim \mathbf{p}_k} [f(\mathbf{x}_{k+1}) \mid \mathbf{x}_k] \leq f(\mathbf{x}_k) - \frac{\alpha_k(\mathbf{p}_k)}{2} \|\nabla f(\mathbf{x}_k)\|_2^2. \quad (4)$$

Consider two examples. In the first one we pick a sub-optimal, but very common [18] distribution:

**Example 2.2** ( $L_i$ -based sampling). *Let  $\mathbf{p}_L \in \Delta^n$  defined as  $[\mathbf{p}_L]_i = \frac{L_i}{\text{Tr}[\mathbf{L}]}$  for  $i \in [n]$ , where  $\mathbf{L} = \text{diag}(L_1, \dots, L_n)$ . Then  $\alpha_k(\mathbf{p}_L) = \frac{1}{\text{Tr}[\mathbf{L}]}$ .*

The distribution  $\mathbf{p}_L$  is often referred to as (fixed) *importance* sampling. In the special case when  $L_i = L$  for all  $i \in [n]$ , this boils down to uniform sampling.

**Example 2.3** (Optimal sampling<sup>2</sup>). *Equation (3) is minimized for probabilities  $[\mathbf{p}_k^*]_i = \frac{\sqrt{L_i} \|\nabla_i f(\mathbf{x}_k)\|_1}{\|\sqrt{\mathbf{L}} \nabla f(\mathbf{x}_k)\|_1}$  and  $\alpha_k(\mathbf{p}_k^*) = \frac{\|\nabla f(\mathbf{x}_k)\|_2^2}{\|\sqrt{\mathbf{L}} \nabla f(\mathbf{x}_k)\|_1^2}$ . Observe  $\frac{1}{\text{Tr}[\mathbf{L}]} \leq \alpha_k(\mathbf{p}_k^*) \leq \frac{1}{L_{\min}}$ , where  $L_{\min} := \min_{i \in [n]} L_i$ .*

To prove this result, we rely on the following Lemma—the proof of which, as well as for the claims above, is deferred to Section A.1 of the appendix. Here  $|\cdot|$  is applied entry-wise.

**Lemma 2.4.** *Define  $V(\mathbf{p}, \mathbf{x}) := \sum_{i=1}^n \frac{L_i [\mathbf{x}]_i^2}{[\mathbf{p}]_i}$ . Then  $\arg \min_{\mathbf{p} \in \Delta^n} V(\mathbf{p}, \mathbf{x}) = \frac{|\sqrt{\mathbf{L}} \mathbf{x}|}{\|\sqrt{\mathbf{L}} \mathbf{x}\|_1}$ .*

**The ideal adaptive algorithm.** We propose to chose the stepsize and the sampling distribution for CD as in Example 2.3. One iteration of the resulting CD method is illustrated in Algorithm 1. Our bounds on the expected one-step progress can be used to derive convergence rates of this algorithm with the standard techniques. This is exemplified in Appendix A.1. In the next Section 3 we develop a practical variant of the ideal algorithm.

**Efficiency gain.** By comparing the estimates provided in the examples above, we see that the expected progress of the proposed method is always at least as good as for the fixed sampling. For instance in the special case where  $L = L_i$  for  $i \in [n]$ , the  $L_i$ -based sampling is just uniform sampling with  $\alpha_k(\mathbf{p}_{\text{unif}}) = \frac{1}{Ln}$ . On the other hand  $\alpha_k(\mathbf{p}_k^*) = \frac{\|\nabla f(\mathbf{x}_k)\|_2^2}{L \|\nabla f(\mathbf{x}_k)\|_1^2}$ , which can be  $n$  times larger than  $\alpha_k(\mathbf{p}_{\text{unif}})$ . The expected one-step progress in this extreme case coincides with the one-step progress of steepest coordinate descent [20].

## 2.2 SGD with Adaptive Sampling

SGD methods are applicable to objective functions which decompose as a sum

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x}) \quad (5)$$

with each  $f_i: \mathbb{R}^d \rightarrow \mathbb{R}$  convex. In previous work [22, 36, 37] it has been argued that the following gradient-based sampling  $[\tilde{\mathbf{p}}_k^*]_i = \frac{\|\nabla f_i(\mathbf{x}_k)\|_2}{\sum_{i=1}^n \|\nabla f_i(\mathbf{x}_k)\|_2}$  is optimal in the sense that it maximizes the expected progress (3). Zhao and Zhang [36] derive complexity estimates for composite functions. For non-composite functions it becomes easier to derive the complexity estimate. For completeness, we add this simpler proof in Appendix A.2.

<sup>2</sup>Here “optimal” refers to the fact that  $\mathbf{p}_k^*$  is optimal with respect to the given model (1) of the objective function. If the model is not accurate, there might exist a sampling that yields larger expected progress on  $f$ .

Algorithm 1 Optimal sampling	Algorithm 2 Proposed safe sampling	Algorithm 3 Fixed sampling
(compute full gradient)	(update l.- and u.-bounds)	
<b>Compute</b> $\nabla f(\mathbf{x}_k)$	Update $\ell, \mathbf{u}$	
(define optimal sampling)	(compute safe sampling)	(define fixed sampling)
Define $(\mathbf{p}_k^*, \alpha_k^*)$ as in Example 2.3	Define $(\hat{\mathbf{p}}_k, \hat{\alpha}_k)$ as in (7)	Define $(\mathbf{p}_L, \bar{\alpha})$ as in Example 2.2
$i_k \sim \mathbf{p}_k^*$	$i_k \sim \hat{\mathbf{p}}_k$	$i_k \sim \mathbf{p}_L$
<b>Compute</b> $\nabla_{i_k} f(\mathbf{x}_k)$	<b>Compute</b> $\nabla_{i_k} f(\mathbf{x}_k)$	<b>Compute</b> $\nabla_{i_k} f(\mathbf{x}_k)$
$\mathbf{x}_{k+1} := \mathbf{x}_k - \frac{\alpha_k^*}{[\mathbf{p}_k^*]_{i_k}} \nabla_{i_k} f(\mathbf{x}_k)$	$\mathbf{x}_{k+1} := \mathbf{x}_k - \frac{\hat{\alpha}_k}{[\hat{\mathbf{p}}_k]_{i_k}} \nabla_{i_k} f(\mathbf{x}_k)$	$\mathbf{x}_{k+1} := \mathbf{x}_k - \frac{\bar{\alpha}}{[\mathbf{p}_L]_{i_k}} \nabla_{i_k} f(\mathbf{x}_k)$

Figure 1: CD with different sampling strategies. Whilst Alg. 1 requires to compute the full gradient, the compute operation in Alg. 2 is as cheap as for fixed importance sampling, Alg. 3. Defining the safe sampling  $\hat{\mathbf{p}}_k$  requires  $O(n \log n)$  time.

### 3 Safe Adaptive Importance Sampling with Limited Information

In the previous section we have seen that gradient-based sampling (Example 2.3) can yield a massive speed-up compared to a static sampling distribution (Example 2.2). However, sampling according to  $\mathbf{p}_k^*$  in CD requires the knowledge of the full gradient  $\nabla f(\mathbf{x}_k)$  in each iteration. And likewise, sampling from  $\hat{\mathbf{p}}_k^*$  in SGD requires the knowledge of the gradient norms of all components—both these operations are in general inadmissible, i.e. the compute cost would void all computational benefits of the iterative (stochastic) methods over full gradient methods.

However, it is often possible to efficiently compute *approximations* of  $\mathbf{p}_k^*$  or  $\hat{\mathbf{p}}_k^*$  instead. In contrast to previous contributions, we here propose a *safe* way to compute such approximations. By this we mean that our approximate sampling is provably never worse than static sampling, and moreover, we show that our solution is the *best possible* with respect to the limited information at hand.

#### 3.1 An Optimization Formulation for Sampling

Formally, we assume that we have in each iteration access to two vectors  $\ell_k, \mathbf{u}_k \in \mathbb{R}_{\geq 0}^n$  that provide safe upper and lower bounds on either the absolute values of the gradient entries ( $[\ell_k]_i \leq |\nabla_i f(\mathbf{x}_k)| \leq [\mathbf{u}_k]_i$ ) for CD, or of the gradient norms in SGD: ( $[\ell_k]_i \leq \|\nabla f_i(\mathbf{x}_k)\|_2 \leq [\mathbf{u}_k]_i$ ). We postpone the discussion of this assumption to Section 4, where we give concrete examples.

The minimization of the upper bound (3) amounts to the equivalent problem<sup>3</sup>

$$\min_{\alpha_k} \min_{\mathbf{p}_k \in \Delta^n} \left[ -\alpha_k \|\mathbf{c}_k\|_2^2 + \frac{\alpha_k^2}{2} V(\mathbf{p}_k, \mathbf{c}_k) \right] \Leftrightarrow \min_{\mathbf{p}_k \in \Delta^n} \frac{V(\mathbf{p}_k, \mathbf{c}_k)}{\|\mathbf{c}_k\|_2^2} \quad (6)$$

where  $\mathbf{c}_k \in \mathbb{R}^n$  represents the *unknown* true gradient. That is, with respect to the bounds  $\ell_k, \mathbf{u}_k$ , we can write  $\mathbf{c}_k \in C_k := \{\mathbf{x} \in \mathbb{R}^n : [\ell_k]_i \leq [\mathbf{x}]_i \leq [\mathbf{u}_k]_i, i \in [n]\}$ . In Example 2.3 we derived the optimal solution for a fixed  $\mathbf{c}_k \in C_k$ . However, this is not sufficient to find the optimal solution for an arbitrary  $\mathbf{c}_k \in C_k$ . Just computing the optimal solution for an arbitrary (but fixed)  $\mathbf{c}_k \in C_k$  is unlikely to yield a good solution. For instance both extreme cases  $\mathbf{c}_k = \ell_k$  and  $\mathbf{c}_k = \mathbf{u}_k$  (the latter choice is quite common, cf. [36, 23]) might be poor. This is demonstrated in the next example.

**Example 3.1.** Let  $\ell = (1, 2)^\top$ ,  $\mathbf{u} = (2, 3)^\top$ ,  $\mathbf{c} = (2, 2)^\top$  and  $L_1 = L_2 = 1$ . Then  $V(\frac{\ell}{\|\ell\|_1}, \mathbf{c}) = \frac{9}{4} \|\mathbf{c}\|_2^2$ ,  $V(\frac{\mathbf{u}}{\|\mathbf{u}\|_1}, \mathbf{c}) = \frac{25}{12} \|\mathbf{c}\|_2^2$ , whereas for uniform sampling  $V(\frac{\mathbf{c}}{\|\mathbf{c}\|_1}, \mathbf{c}) = 2 \|\mathbf{c}\|_2^2$ .

**The proposed sampling.** As a consequence of these observations, we propose to solve the following optimization problem to find the best sampling distribution with respect to  $C_k$ :

$$v_k := \min_{\mathbf{p} \in \Delta^n} \max_{\mathbf{c} \in C_k} \frac{V(\mathbf{p}, \mathbf{c})}{\|\mathbf{c}\|_2^2}, \quad \text{and to set} \quad (\alpha_k, \mathbf{p}_k) := \left( \frac{1}{v_k}, \hat{\mathbf{p}}_k \right), \quad (7)$$

where  $\hat{\mathbf{p}}_k$  denotes a solution of (7). The resulting algorithm for CD is summarized in Alg. 2.

In the remainder of this section we discuss the properties of the solution  $\hat{\mathbf{p}}_k$  (Theorem 3.2) and how such a solution can be efficiently be computed (Theorem 3.4, Algorithm 4).

<sup>3</sup>Although only shown here for CD, an equivalent optimization problem arises for SGD methods, cf. [36].

### 3.2 Proposed Sampling and its Properties

**Theorem 3.2.** Let  $(\hat{\mathbf{p}}, \hat{\mathbf{c}}) \in \Delta^n \times \mathbb{R}_{\geq 0}^n$  denote a solution of (7). Then  $L_{\min} \leq v_k \leq \text{Tr}[\mathbf{L}]$  and

- (i)  $\max_{\mathbf{c} \in C_k} \frac{V(\hat{\mathbf{p}}, \mathbf{c})}{\|\mathbf{c}\|_2^2} \leq \max_{\mathbf{c} \in C_k} \frac{V(\mathbf{p}, \mathbf{c})}{\|\mathbf{c}\|_2^2}, \forall \mathbf{p} \in \Delta^n; \quad (\hat{\mathbf{p}} \text{ has the best worst-case guarantee})$
- (ii)  $V(\hat{\mathbf{p}}, \mathbf{c}) \leq \text{Tr}[\mathbf{L}] \cdot \|\mathbf{c}\|_2^2, \forall \mathbf{c} \in C_k. \quad (\hat{\mathbf{p}} \text{ is always better than } L_i\text{-based sampling})$

**Remark 3.3.** In the special case  $L_i = L$  for all  $i \in [n]$ , the  $L_i$ -based sampling boils down to uniform sampling (Example 2.2) and  $\hat{\mathbf{p}}$  is better than uniform sampling:  $V(\hat{\mathbf{p}}, \mathbf{c}) \leq Ln \|\mathbf{c}\|_2^2, \forall \mathbf{c} \in C_k$ .

*Proof.* Property (i) is an immediate consequence of (7). Moreover, observe that the  $L_i$ -based sampling  $\mathbf{p}_L$  is a feasible solution in (7) with value  $\frac{V(\mathbf{p}_L, \mathbf{c})}{\|\mathbf{c}\|_2^2} \equiv \text{Tr}[\mathbf{L}]$  for all  $\mathbf{c} \in C_k$ . Hence

$$L_{\min} \leq \frac{\|\sqrt{\mathbf{L}}\mathbf{c}\|_1^2}{\|\mathbf{c}\|_2^2} \stackrel{2.4}{=} \min_{\mathbf{p} \in \Delta^n} \frac{V(\mathbf{p}, \mathbf{c})}{\|\mathbf{c}\|_2^2} \leq \frac{V(\hat{\mathbf{p}}, \mathbf{c})}{\|\mathbf{c}\|_2^2} \stackrel{(*)}{\leq} \frac{V(\hat{\mathbf{p}}, \hat{\mathbf{c}})}{\|\hat{\mathbf{c}}\|_2^2} \stackrel{(7)}{\leq} \max_{\mathbf{c} \in C_k} \frac{V(\mathbf{p}_L, \mathbf{c})}{\|\mathbf{c}\|_2^2} = \text{Tr}[\mathbf{L}], \quad (8)$$

for all  $\mathbf{c} \in C_k$ , thus  $v_k \in [L_{\min}, \text{Tr}[\mathbf{L}]]$  and (ii) follows. We prove inequality  $(*)$  in the appendix, by showing that min and max can be interchanged in (7).  $\square$

**A geometric interpretation.** We show in Appendix B that the optimization problem (7) can equivalently be written as  $\sqrt{v_k} = \max_{\mathbf{c} \in C_k} \frac{\|\sqrt{\mathbf{L}}\mathbf{c}\|_1}{\|\mathbf{c}\|_2} = \max_{\mathbf{c} \in C_k} \frac{\langle \sqrt{\mathbf{L}}\mathbf{c}, \mathbf{l} \rangle}{\|\mathbf{c}\|_2}$ , where  $[\mathbf{l}]_i = L_i$  for  $i \in [n]$ . The maximum is thus attained for vectors  $\mathbf{c} \in C_k$  that minimize the angle with the vector  $\mathbf{l}$ .

**Theorem 3.4.** Let  $\mathbf{c} \in C_k$ ,  $\mathbf{p} = \frac{\sqrt{\mathbf{L}}\mathbf{c}}{\|\sqrt{\mathbf{L}}\mathbf{c}\|_1}$  and denote  $m = \|\mathbf{c}\|_2^2 \cdot \|\sqrt{\mathbf{L}}\mathbf{c}\|_1^{-1}$ . If

$$[\mathbf{c}]_i = \begin{cases} [\mathbf{u}_k]_i & \text{if } [\mathbf{u}_k]_i \leq \sqrt{L_i}m, \\ [\mathbf{l}_k]_i & \text{if } [\mathbf{l}_k]_i \geq \sqrt{L_i}m, \\ \sqrt{L_i}m & \text{otherwise,} \end{cases} \quad \forall i \in [n], \quad (9)$$

then  $(\mathbf{p}, \mathbf{c})$  is a solution to (7). Moreover, such a solution can be computed in time  $O(n \log n)$ .

*Proof.* This can be proven by examining the optimality conditions of problem (7). This is deferred to Section B.1 of the appendix. A procedure that computes such a solution is depicted in Algorithm 4. The algorithm makes extensive use of (9). For simplicity, assume first  $\mathbf{L} = \mathbf{I}_n$  for now. In each iteration  $t$ , a potential solution vector  $\mathbf{c}_t$  is proposed, and it is verified whether this vector satisfies all optimality conditions. In Algorithm 4,  $\mathbf{c}_t$  is just implicit, with  $[\mathbf{c}_t]_i = [\mathbf{c}]_i$  for decided indices  $i \in D$  and  $[\mathbf{c}_t]_i = [\sqrt{L_i}m]_i$  for undecided indices  $i \notin D$ . After at most  $n$  iterations a valid solution is found. By sorting the components of  $\sqrt{L_i}^{-1}\mathbf{l}_k$  and  $\sqrt{L_i}^{-1}\mathbf{u}_k$  by their magnitude, at most a linear number of inequality checks in (9) have to be performed in total. Hence the running time is dominated by the  $O(n \log n)$  complexity of the sorting algorithm. A formal proof is given in the appendix.  $\square$

---

#### Algorithm 4 Computing the Safe Sampling for Gradient Information $\ell, \mathbf{u}$

---

- 1: **Input:**  $0_n \leq \ell \leq \mathbf{u}$ , **L, Initialize:**  $\mathbf{c} = \mathbf{0}_n, u = 1, \ell = n, D = \emptyset$ .
  - 2:  $\ell^{\text{sort}} := \text{sort\_asc}(\sqrt{\mathbf{L}^{-1}}\ell), \mathbf{u}^{\text{sort}} := \text{sort\_asc}(\sqrt{\mathbf{L}^{-1}}\mathbf{u}), m = \max(\ell^{\text{sort}})$
  - 3: **while**  $u \leq \ell$  **do**
  - 4:   **if**  $[\ell^{\text{sort}}]_\ell > m$  **then** (largest undecided lower bound is violated)
  - 5:     Set corresponding  $[\mathbf{c}]_{\text{index}} := [\sqrt{\mathbf{L}}\ell^{\text{sort}}]_\ell; \ell := \ell - 1; D := D \cup \{\text{index}\}$
  - 6:   **else if**  $[\mathbf{u}^{\text{sort}}]_u < m$  **then** (smallest undecided upper bound is violated)
  - 7:     Set corresponding  $[\mathbf{c}]_{\text{index}} := [\sqrt{\mathbf{L}}\mathbf{u}^{\text{sort}}]_u; u := u + 1; D := D \cup \{\text{index}\}$
  - 8:   **else**
  - 9:     **break** (no constraints are violated)
  - 10:   **end if**
  - 11:    $m := \|\mathbf{c}\|_2^2 \cdot \|\sqrt{\mathbf{L}}\mathbf{c}\|_1^{-1}$  (update  $m$  as in (9))
  - 12: **end while**
  - 13: Set  $[\mathbf{c}]_i := \sqrt{L_i}m$  for all  $i \notin D$  and **Return**  $\left(\mathbf{c}, \mathbf{p} = \frac{\sqrt{\mathbf{L}}\mathbf{c}}{\|\sqrt{\mathbf{L}}\mathbf{c}\|_1}, v = \frac{\|\sqrt{\mathbf{L}}\mathbf{c}\|_1^2}{\|\mathbf{c}\|_2^2}\right)$
-

**Competitive Ratio.** We now compare the proposed sampling distribution  $\hat{\mathbf{p}}_k$  with the optimal sampling solution in *hindsight*. We know that if the true (gradient) vector  $\tilde{\mathbf{c}} \in C_k$  would be given to us, then the corresponding optimal probability distribution would be  $\mathbf{p}^*(\tilde{\mathbf{c}}) = \frac{\sqrt{\mathbf{L}}\tilde{\mathbf{c}}}{\|\sqrt{\mathbf{L}}\tilde{\mathbf{c}}\|_1}$  (Example 2.3).

Thus, for this  $\tilde{\mathbf{c}}$  we can now analyze the ratio  $\frac{V(\hat{\mathbf{p}}_k, \tilde{\mathbf{c}})}{V(\mathbf{p}^*(\tilde{\mathbf{c}}), \tilde{\mathbf{c}})}$ . As we are interested in the worst case ratio among all possible candidates  $\tilde{\mathbf{c}} \in C_k$ , we define

$$\rho_k := \max_{\mathbf{c} \in C_k} \frac{V(\hat{\mathbf{p}}_k, \mathbf{c})}{V(\mathbf{p}^*(\mathbf{c}), \mathbf{c})} = \max_{\mathbf{c} \in C_k} \frac{V(\hat{\mathbf{p}}_k, \mathbf{c})}{\|\sqrt{\mathbf{L}}\mathbf{c}\|_1^2}. \quad (10)$$

**Lemma 3.5.** Let  $w_k := \min_{\mathbf{c} \in C_k} \frac{\|\sqrt{\mathbf{L}}\mathbf{c}\|_1^2}{\|\mathbf{c}\|_2^2}$ . Then  $L_{\min} \leq w_k \leq v_k$ , and  $\rho_k \leq \frac{v_k}{w_k} (\leq \frac{v_k}{L_{\min}})$ .

**Lemma 3.6.** Let  $\gamma \geq 1$ . If  $[C_k]_i \cap \gamma[C_k]_i = \emptyset$  and  $\gamma^{-1}[C_k]_i \cap [C_k]_i = \emptyset$  for all  $i \in [n]$  (here  $[C_k]_i$  denotes the projection on the  $i$ -th coordinate), then  $\rho_k \leq \gamma^4$ .

These two lemma provide bounds on the competitive ratio. Whilst Lemma 3.6 relies on a relative accuracy condition, Lemma 3.5 can always be applied. However, the corresponding minimization problem is non-convex. Note that knowledge of  $\rho_k$  is not needed to run the algorithm.

## 4 Example Safe Gradient Bounds

In this section, we argue that for a large class of objective functions of interest in machine learning, suitable safe upper and lower bounds  $\ell, \mathbf{u}$  on the gradient along every coordinate direction can be estimated and maintained efficiently during optimization. A similar argument can be given for the efficient approximation of component wise gradient norms in finite sum objective based stochastic gradient optimization.

As the guiding example, we will here showcase the training of generalized linear models (GLMs) as e.g. in regression, classification and feature selection. These models are formulated in terms of a given data matrix  $\mathbf{A} \in \mathbb{R}^{d \times n}$  with columns  $\mathbf{a}_i \in \mathbb{R}^d$  for  $i \in [n]$ .

**Coordinate Descent - GLMs with Arbitrary Regularizers.** Consider general objectives of the form  $f(\mathbf{x}) := h(\mathbf{A}\mathbf{x}) + \sum_{i=1}^n \psi_i([\mathbf{x}]_i)$  with an arbitrary convex separable regularizer term given by the  $\psi_i: \mathbb{R} \rightarrow \mathbb{R}$  for  $i \in [n]$ . A key example is when  $h: \mathbb{R}^d \rightarrow \mathbb{R}$  describes the *least-squares* regression objective  $h(\mathbf{A}\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$  for a  $\mathbf{b} \in \mathbb{R}^d$ . Using that this  $h$  is twice differentiable with  $\nabla^2 h(\mathbf{A}\mathbf{x}) = \mathbf{I}_n$ , it is easy to see that we can track the evolution of all gradient entries, when performing CD steps, as follows:

$$\nabla_i f(\mathbf{x}_{k+1}) - \nabla_i f(\mathbf{x}_k) = \gamma_k \langle \mathbf{a}_i, \mathbf{a}_{i_k} \rangle, \quad \forall i \neq i_k. \quad (11)$$

for  $i_k$  being the coordinate changed in step  $k$  (here we also used the separability of the regularizer).

Therefore, all gradient changes can be tracked exactly if the inner products of all datapoints are available, or approximately if those inner products can be upper and lower bounded. For computational efficiency, we in our experiments simply use Cauchy-Schwarz  $|\langle \mathbf{a}_i, \mathbf{a}_{i_k} \rangle| \leq \|\mathbf{a}_i\| \cdot \|\mathbf{a}_{i_k}\|$ . This results in safe upper and lower bounds  $[\ell_{k+1}]_i \leq \nabla_i f(\mathbf{x}_{k+1}) \leq [\mathbf{u}_{k+1}]_i$  for all inactive coordinates  $i \neq i_k$ . (For the active coordinate  $i_k$  itself one observes the true value without uncertainty). These bounds can be updated in linear time  $O(n)$  in every iteration.

For general smooth  $h$  (again with arbitrary separable regularizers  $\psi_i$ ), (11) can readily be extended to hold [32, Lemma 4.1], the inner product change term becoming  $\langle \mathbf{a}_i, \nabla^2 f(\mathbf{A}\tilde{\mathbf{x}})\mathbf{a}_{i_k} \rangle$  instead, when assuming  $h$  is twice-differentiable. Here  $\tilde{\mathbf{x}}$  will be an element of the line segment  $[\mathbf{x}_k, \mathbf{x}_{k+1}]$ .

**Stochastic Gradient Descent - GLMs.** We now present a similar result for finite sum problems (5) for the use in SGD based optimization, that is  $f(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n h_i(\mathbf{a}_i^\top \mathbf{x})$ .

**Lemma 4.1.** Consider  $f: \mathbb{R}^d \rightarrow \mathbb{R}$  as above, with twice differentiable  $h_i: \mathbb{R} \rightarrow \mathbb{R}$ . Let  $\mathbf{x}_k, \mathbf{x}_{k+1} \in \mathbb{R}^d$  denote two successive iterates of SGD, i.e.  $\mathbf{x}_{k+1} := \mathbf{x}_k - \eta_k \mathbf{a}_{i_k} \nabla h_{i_k}(\mathbf{a}_{i_k}^\top \mathbf{x}_k) = \mathbf{x}_k + \gamma_k \mathbf{a}_{i_k}$ . Then there exists  $\tilde{\mathbf{x}} \in \mathbb{R}^d$  on the line segment between  $\mathbf{x}_k$  and  $\mathbf{x}_{k+1}$ ,  $\tilde{\mathbf{x}} \in [\mathbf{x}_k, \mathbf{x}_{k+1}]$  with

$$\nabla f_i(\mathbf{x}_{k+1}) - \nabla f_i(\mathbf{x}_k) = \gamma_k \nabla^2 h_i(\mathbf{a}_i^\top \tilde{\mathbf{x}}) \langle \mathbf{a}_i, \mathbf{a}_{i_k} \rangle \mathbf{a}_i, \quad \forall i \neq i_k. \quad (12)$$

This leads to safe upper and lower bounds for the norms of the partial gradient,  $[\ell_k]_i \leq \|\nabla f_i(\mathbf{x}_k)\|_2 \leq [\mathbf{u}_k]_i$ , that can be updated in linear time  $O(n)$ , analogous to the coordinate case discussed above.<sup>4</sup>

We note that there are many other ways to track safe gradient bounds for relevant machine learning problems, including possibly more tight ones. We here only illustrate the simplest variants, highlighting the fact that our new sampling procedure works for any safe bounds  $\ell, \mathbf{u}$ .

**Computational Complexity.** In this section, we have demonstrated how safe upper and lower bounds  $\ell, \mathbf{u}$  on the gradient information can be obtained for GLMs, and argued that these bounds can be updated in time  $O(n)$  per iteration of CD and SGD. The computation of the proposed sampling takes  $O(n \log n)$  time (Theorem 3.4). Hence, the introduced overhead in Algorithm 2 compared to fixed sampling (Algorithm 3) is of the order  $O(n \log n)$  in every iteration. The computation of one coordinate of the gradient,  $\nabla_{i_k} f(\mathbf{x}_k)$ , takes time  $\Theta(d)$  for general data matrices. Hence, when  $d = \Omega(n)$ , the introduced overhead reduces to  $O(\log n)$  per iteration.

## 5 Empirical Evaluation

In this section we evaluate the empirical performance of our proposed adaptive sampling scheme on relevant machine learning tasks. In particular, we illustrate performance on generalized linear models with  $L1$  and  $L2$  regularization, as of the form (5),

$$\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n h_i(\mathbf{a}_i^\top \mathbf{x}) + \lambda \cdot r(\mathbf{x}) \quad (13)$$

We use square loss, squared hinge loss as well as logistic loss for the data fitting terms  $h_i$ , and  $\|\mathbf{x}\|_1$  and  $\|\mathbf{x}\|_2^2$  for the regularizer  $r(\mathbf{x})$ . The datasets used in the evaluation are *rcv1*, *real-sim* and *news20*.<sup>5</sup> The *rcv1* dataset consists of 20,242 samples with 47,236 features, *real-sim* contains 72,309 datapoints and 20,958 features and *news20* contains 19,996 datapoints and 1,355,191 features. For all datasets we set unnormalized features with all the non-zero entries set to 1 (bag-of-words features). By *real-sim'* and *rcv1'* we denote a subset of the data chosen by randomly selecting 10,000 features and 10,000 datapoints. By *news20'* we denote a subset of the data chose by randomly selecting 15% of the features and 15% of the datapoints. A regularization parameter  $\lambda = 0.1$  is used for all experiments.

Our results show the evolution of the optimization objective over time or number of epochs (an epoch corresponding to  $n$  individual updates). To compute safe lower and upper bounds we use the methods presented in Section 4 with no special initialization, i.e.  $\ell_0 = \mathbf{0}_n$ ,  $\mathbf{u}_0 = \infty_n$ .

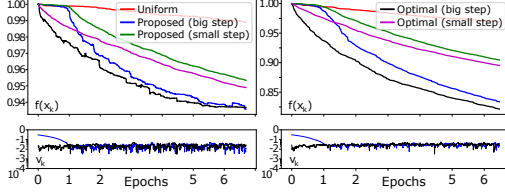
**Coordinate Descent.** In Figure 2 we compare the effect of the fixed stepsize  $\alpha_k = \frac{1}{L_n}$  (denoted as “small”) vs. the time varying optimal stepsize (denoted as “big”) as discussed in Section 2. Results are shown for optimal sampling  $\mathbf{p}_k^*$  (with optimal stepsize  $\alpha_k(\mathbf{p}_k^*)$ , cf. Example 2.3), our proposed sampling  $\hat{\mathbf{p}}_k$  (with optimal stepsize  $\alpha_k(\hat{\mathbf{p}}_k) = v_k^{-1}$ , cf. (7)) and uniform sampling (with optimal stepsize  $\alpha_k(\mathbf{p}_L) = \frac{1}{L_n}$ , as here  $\mathbf{L} = L\mathbf{I}_n$ , cf. Example 2.2). As the experiment aligns with theory—confirming the advantage of the varying “big” stepsizes—we only show the results for Algorithms 1–3 in the remaining plots.

Performance for squared hinge loss, as well as logistic regression with  $L1$  and  $L2$  regularization is presented in Figure 3 and Figure 4 respectively. In Figures 5 and 6 we report the iteration complexity vs. accuracy as well as timing vs. accuracy results on the full dataset for coordinate descent with square loss and  $L1$  (Lasso) and  $L2$  regularization (Ridge).

**Theoretical Sampling Quality.** As part of the CD performance results in Figures 2–6 we include an additional evolution plot on the bottom of each figure to illustrate the values  $v_k$  which determine the stepsize ( $\hat{\alpha}_k = v_k^{-1}$ ) for the proposed Algorithm 2 (blue) and the optimal stepsizes of Algorithm 1 (black) which rely on the full gradient information. The plots show the normalized values  $\frac{v_k}{\text{Tr}(\mathbf{L})}$ , i.e. the relative improvement over  $L_i$ -based importance sampling. The results show that despite only relying on very loose safe gradient bounds, the proposed adaptive sampling is able to strongly benefit from the additional information.

<sup>4</sup>Here we use the efficient representation  $\nabla f_i(\mathbf{x}) = \theta(\mathbf{x}) \cdot \mathbf{a}_i$  for  $\theta(\mathbf{x}) \in \mathbb{R}$ .

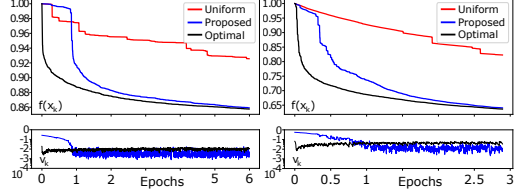
<sup>5</sup>All data are available at [www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/](http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/)



(a) *rcvI'*, L1 reg.

(b) *rcvI'*, L2 reg.

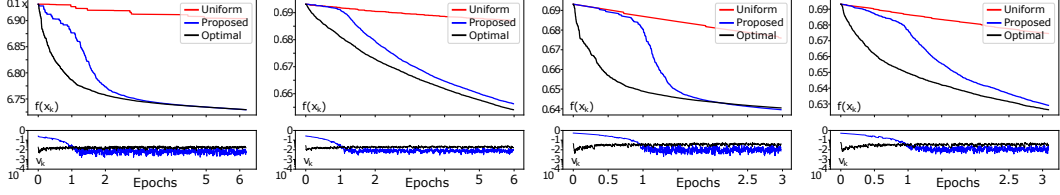
Figure 2: (CD, square loss) Fixed vs. adaptive sampling strategies, and dependence on stepsizes. With “big”  $\alpha_k = v_k^{-1}$  and “small”  $\alpha_k = \frac{1}{\text{Tr}[\mathbf{L}]}$ .



(a) *rcvI'*, L1 reg.

(b) *real-sim'*, L2 reg.

Figure 3: (CD, squared hinge loss) Function value vs. number of iterations for optimal step-size  $\alpha_k = v_k^{-1}$ .



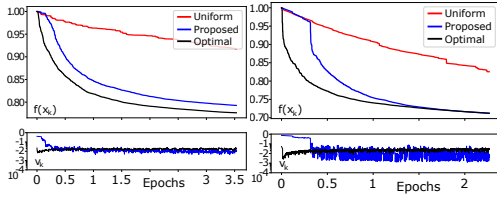
(a) *rcvI'*, L1 reg.

(b) *rcvI'*, L2 reg.

(c) *real-sim'*, L1 reg.

(d) *real-sim'*, L2 reg.

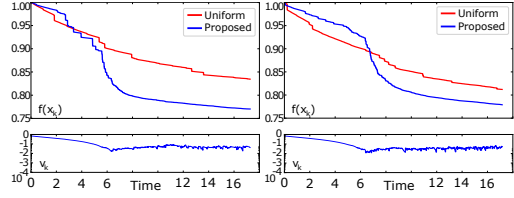
Figure 4: (CD, logistic loss) Function value vs. number of iterations for different sampling strategies. Bottom: Evolution of the value  $v_k$  which determines the optimal stepsize ( $\hat{\alpha}_k = v_k^{-1}$ ). The plots show the normalized values  $\frac{v_k}{\text{Tr}[\mathbf{L}]}$ , i.e. the relative improvement over  $L_i$ -based importance sampling.



(a) *rcvI*, L1 reg.

(b) *real-sim*, L1 reg.

Figure 5: (CD, square loss) Function value vs. number of iterations on the full datasets.



(a) *real-sim*, L1 reg.

(b) *real-sim*, L2 reg.

Figure 6: (CD, square loss) Function value vs. clock time on the full datasets. (Data for the optimal sampling omitted, as this strategy is not competitive time-wise.)

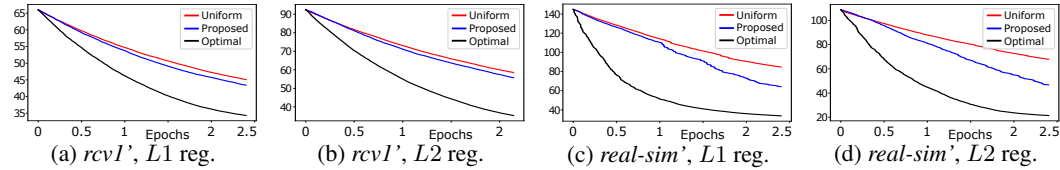
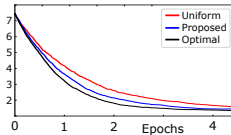
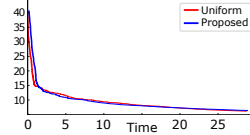


Figure 7: (SGD, square loss) Function value vs. number of iterations.



(a) *news20'*, L1 reg.

Figure 8: (SGD, square loss) Function value vs. number of iterations.



(a) *news20'*, L1 reg.

Figure 9: (SGD square loss) Function value vs. clock time.



**Stochastic Gradient Descent.** Finally, we also evaluate the performance of our approach when used within SGD with  $L1$  and  $L2$  regularization and square loss. In Figures 7–8 we report the iteration complexity vs. accuracy results and in Figure 9 the timing vs. accuracy results. The time units in Figures 6 and 9 are not directly comparable, as the experiments were conducted on different machines.

We observe that on all three datasets SGD with the optimal sampling performs only slightly better than uniform sampling. This is in contrast with the observations for CD, where the optimal sampling yields a significant improvement. Consequently, the effect of the proposed sampling is less pronounced in the three SGD experiments.

**Summary.** The main findings of our experimental study can be summarized as follows:

- **Adaptive importance sampling significantly outperforms fixed importance sampling in iterations and time.** The results show that (i) convergence in terms of iterations is almost as good as for the optimal (but not efficiently computable) gradient-based sampling and (ii) the introduced computational overhead is small enough to outperform fixed importance sampling in terms of total computation time.
- **Adaptive sampling requires adaptive stepsizes.** The adaptive stepsize strategies of Algorithms 1 and 2 allow for much faster convergence than conservative fixed-stepsize strategies. In the experiments, the measured value  $v_k$  was always significantly below the worst case estimate, in alignment with the observed convergence.
- **Very loose safe gradient bounds are sufficient.** Even the bounds derived from the the very naïve gradient information obtained by estimating scalar products resulted in significantly better sampling than using no gradient information at all. Further, no initialization of the gradient estimates is needed (at the beginning of the optimization process the proposed adaptive method performs close to the fixed sampling but accelerates after just one epoch).

## 6 Conclusion

In this paper we propose a safe adaptive importance sampling scheme for CD and SGD algorithms. We argue that optimal gradient-based sampling is theoretically well justified. To make the computation of the adaptive sampling distribution computationally tractable, we rely on safe lower and upper bounds on the gradient. However, in contrast to previous approaches, we use these bounds in a novel way: in each iteration, we formulate the problem of picking the optimal sampling distribution as a convex optimization problem and present an efficient algorithm to compute the solution. The novel sampling provably performs better than any fixed importance sampling—a guarantee which could not be established for previous samplings that were also derived from safe lower and upper bounds.

The computational cost of the proposed scheme is of the order  $O(n \log n)$  per iteration—this is on many problems comparable with the cost to evaluate a single component (coordinate, sum-structure) of the gradient, and the scheme can thus be implemented at no extra computational cost. This is verified by timing experiments on real datasets.

We discussed one simple method to track the gradient information in GLMs during optimization. However, we feel that the machine learning community could profit from further research in that direction, for instance by investigating how such safe bounds can efficiently be maintained on more complex models. Our approach can immediately be applied when the tracking of the gradient is delegated to other machines in a distributed setting, like for instance in [1].

## References

- [1] Guillaume Alain, Alex Lamb, Chinnadhurai Sankar, Aaron Courville, and Yoshua Bengio. Variance Reduction in SGD by Distributed Importance Sampling. *arXiv.org*, February 2015.
- [2] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even Faster Accelerated Coordinate Descent Using Non-Uniform Sampling. In *ICML 2017 - Proceedings of the 34th International Conference on Machine Learning*, pages 1110–1119. June 2016.
- [3] Ichiro Takeuchi Atsushi Shibagaki. Stochastic Primal Dual Coordinate Method with Non-Uniform Sampling Based on Optimality Violations. *arXiv.org*, October 2017.

- [4] Stephen P Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge University Press, 2004.
- [5] Dominik Csiba, Zheng Qu, and Peter Richtárik. Stochastic Dual Coordinate Ascent with Adaptive Probabilities. In *ICML 2015 - Proceedings of the 32th International Conference on Machine Learning*, February 2015.
- [6] Dominik Csiba and Peter Richtárik. Importance Sampling for Minibatches. *arXiv.org*, February 2016.
- [7] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, December 2007.
- [8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–22, 2010.
- [9] Wenjiang J. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998.
- [10] Xi He and Martin Takáč. Dual Free Adaptive Mini-batch SDCA for Empirical Risk Minimization. *arXiv.org*, October 2015.
- [11] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathya Keerthi, and S Sundararajan. A Dual Coordinate Descent Method for Large-scale Linear SVM. In *ICML 2008 - the 25th International Conference on Machine Learning*, pages 408–415, New York, USA, 2008. ACM Press.
- [12] Hidetoshi Komiya. Elementary proof for sion’s minimax theorem. *Kodai Math. J.*, 11(1):5–7, 1988.
- [13] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an  $O(1/t)$  convergence rate for projected stochastic subgradient descent. *arXiv.org*, December 2012.
- [14] Jun Liu, Zheng Zhao, Jie Wang, and Jieping Ye. Safe Screening with Variational Inequalities and Its Application to Lasso. In *ICML 2014 - Proceedings of the 31st International Conference on Machine Learning*, pages 289–297, 2014.
- [15] Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Gap Safe screening rules for sparsity enforcing penalties. *JMLR*, 2017.
- [16] Deanna Needell, Rachel Ward, and Nathan Srebro. Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm. In *NIPS 2014 - Advances in Neural Information Processing Systems* 27, pages 1017–1025, 2014.
- [17] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.
- [18] Yurii Nesterov. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.
- [19] Yurii Nesterov and Sebastian U. Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017.
- [20] Julie Nutini, Mark W Schmidt, Issam H Laradji, Michael P Friedlander, and Hoyt A Koepke. Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. In *ICML*, pages 1632–1641, 2015.
- [21] Anton Osokin, Jean-Baptiste Alayrac, Isabella Lukasewitz, Puneet K. Dokania, and Simon Lacoste-Julien. Minding the gaps for block frank-wolfe optimization of structured svms. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML’16*, pages 593–602. JMLR.org, 2016.
- [22] Guillaume Papa, Pascal Bianchi, and Stéphan Cléménçon. Adaptive Sampling for Incremental Optimization Using Stochastic Gradient Descent. *ALT 2015 - 26th International Conference on Algorithmic Learning Theory*, pages 317–331, 2015.
- [23] Dmytro Perekrestenko, Volkan Cevher, and Martin Jaggi. Faster Coordinate Descent via Adaptive Importance Sampling. In *AISTATS 2017 - Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 869–877. PMLR, 20–22 Apr 2017.
- [24] Zheng Qu, Peter Richtárik, and Tong Zhang. Randomized Dual Coordinate Ascent with Arbitrary Sampling. *arXiv.org*, November 2014.

- [25] Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 10(6):1233–1243, 2016.
- [26] Mark Schmidt, Reza Babanezhad, Mohamed Ahmed, Aaron Defazio, Ann Clifton, and Anoop Sarkar. Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields. In *AISTATS 2015 - Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, pages 819–828. PMLR, 09–12 May 2015.
- [27] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal Estimated Sub-Gradient Solver for SVM. *Mathematical Programming*, 127(1):3–30, October 2010.
- [28] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic Methods for  $l_1$ -regularized Loss Minimization. *JMLR*, 12:1865–1892, June 2011.
- [29] Shai Shalev-Shwartz and Tong Zhang. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *JMLR*, 14:567–599, February 2013.
- [30] Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.
- [31] S. U. Stich, C. L. Müller, and B. Gärtner. Variable metric random pursuit. *Mathematical Programming*, 156(1):549–579, Mar 2016.
- [32] Sebastian U. Stich, Anant Raj, and Martin Jaggi. Approximate steepest coordinate descent. In Doina Precup and Yee Whye Teh, editors, *ICML 2017 - Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 3251–3259. PMLR, 06–11 Aug 2017.
- [33] Thomas Strohmer and Roman Vershynin. A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262, 2008.
- [34] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009.
- [35] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.
- [36] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *ICML 2015 - Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1–9. PMLR, 07–09 Jul 2015.
- [37] Rong Zhu. Gradient-based sampling: An adaptive importance sampling for least-squares. In *NIPS - Advances in Neural Information Processing Systems 29*, pages 406–414. 2016.