Painless Stochastic Gradient: Interpolation, Line-Search, and Convergence Rates

Sharan Vaswani
Mila, Université de Montréal

Aaron Mishkin
University of British Columbia

Issam Laradji
University of British Columbia
Element AI

Mark Schmidt
University of British Columbia, 1QBit
CCAI Affiliate Chair (Amii)

Gauthier Gidel
Mila, Université de Montréal
Element AI

Simon Lacoste-Julien[†]
Mila, Université de Montréal

Abstract

Recent works have shown that stochastic gradient descent (SGD) achieves the fast convergence rates of full-batch gradient descent for over-parameterized models satisfying certain interpolation conditions. However, the step-size used in these works depends on unknown quantities and SGD's practical performance heavily relies on the choice of this step-size. We propose to use line-search techniques to automatically set the step-size when training models that can interpolate the data. In the interpolation setting, we prove that SGD with a stochastic variant of the classic Armijo line-search attains the deterministic convergence rates for both convex and strongly-convex functions. Under additional assumptions, SGD with Armijo line-search is shown to achieve fast convergence for non-convex functions. Furthermore, we show that stochastic extra-gradient with a Lipschitz line-search attains linear convergence for an important class of non-convex functions and saddle-point problems satisfying interpolation. To improve the proposed methods' practical performance, we give heuristics to use larger step-sizes and acceleration. We compare the proposed algorithms against numerous optimization methods on standard classification tasks using both kernel methods and deep networks. The proposed methods result in competitive performance across all models and datasets, while being robust to the precise choices of hyper-parameters. For multi-class classification using deep networks, SGD with Armijo line-search results in both faster convergence and better generalization.

1 Introduction

Stochastic gradient descent (SGD) and its variants [18, 21, 35, 39, 72, 82, 87] are the preferred optimization methods in modern machine learning. They only require the gradient for one training example (or a small "mini-batch" of examples) in each iteration and thus can be used with large datasets. These first-order methods have been particularly successful for training highly-expressive, over-parameterized models such as non-parametric regression [7, 45] and deep neural networks [9, 88]. However, the practical efficiency of stochastic gradient methods is adversely affected by two challenges: (i) their performance heavily relies on the choice of the step-size ("learning rate") [9, 70] and (ii) their slow convergence compared to methods that compute the full gradient (over all training examples) in each iteration [58].

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada
[†] Canada CIFAR AI Chair..

Variance-reduction (VR) methods [18, 35, 72] are relatively new variants of SGD that improve its slow convergence rate. These methods exploit the finite-sum structure of typical loss functions arising in machine learning, achieving both the low iteration cost of SGD and the fast convergence rate of deterministic methods that compute the full-gradient in each iteration. Moreover, VR makes setting the learning rate easier and there has been work exploring the use of line-search techniques for automatically setting the step-size for these methods [71, 72, 76, 81]. These methods have resulted in impressive performance on a variety of problems. However, the improved performance comes at the cost of additional memory [72] or computational [18, 35] overheads, making these methods less appealing when training high-dimensional models on large datasets. Moreover, in practice VR methods do not tend to converge faster than SGD on over-parameterized models [19].

Indeed, recent works [5, 13, 33, 47, 52, 73, 83] have shown that when training over-parameterized models, classic SGD with a constant step-size and *without VR* can achieve the convergence rates of full-batch gradient descent. These works assume that the model is expressive enough to *interpolate* the data. The interpolation condition is satisfied for models such as non-parametric regression [7, 45], over-parametrized deep neural networks [88], boosting [69], and for linear classifiers on separable data. However, the good performance of SGD in this setting relies on using the proposed constant step-size, which depends on problem-specific quantities not known in practice. On the other hand, there has been a long line of research on techniques to automatically set the step-size for classic SGD. These techniques include using meta-learning procedures to modify the main stochastic algorithm [2, 6, 63, 75, 77, 86, 86], heuristics to adjust the learning rate on the fly [20, 43, 70, 74], and recent adaptive methods inspired by online learning [21, 39, 51, 60, 67, 68, 87]. However, none of these techniques have been proved to achieve the fast convergence rates that we now know are possible in the over-parametrized setting.

In this work, we use classical line-search methods [59] to automatically set the step-size for SGD when training over-parametrized models. Line-search is a standard technique to adaptively set the step-size for deterministic methods that evaluate the full gradient in each iteration. These methods make use of additional function/gradient evaluations to characterize the function around the current iterate and adjust the magnitude of the descent step. The additional noise in SGD complicates the use of line-searches in the general stochastic setting and there have only been a few attempts to address this. Mahsereci et al. [53] define a Gaussian process model over probabilistic Wolfe conditions and use it to derive a termination criterion for the line-search. The convergence rate of this procedure is not known, and experimentally we found that our proposed line-search technique is simpler to implement and more robust. Other authors [12, 17, 22, 42, 62] use a line-search termination criteria that requires function/gradient evaluations averaged over multiple samples. However, in order to achieve convergence, the number of samples required per iteration (the "batch-size") increases progressively, losing the low per iteration cost of SGD. Other work [11, 26] exploring trust-region methods assume that the model is sufficiently accurate, which is not guaranteed in the general stochastic setting. In contrast to these works, our line-search procedure does not consider the general stochastic setting and is designed for models that satisfy interpolation; it achieves fast rates in the over-parameterized regime without the need to manually choose a step-size or increase the batch size.

We make the following contributions: in Section 3 we prove that, under interpolation, SGD with a stochastic variant of the Armijo line-search attains the convergence rates of full-batch gradient descent in both the convex and strongly-convex settings. We achieve these rates under weaker assumptions than the prior work [83] and *without* the explicit knowledge of problem specific constants. We then consider minimizing non-convex functions satisfying interpolation [5, 83]. Previous work [5] proves that constant step-size SGD achieves a linear rate for non-convex functions satisfying the PL inequality [37, 65]. SGD is further known to achieve deterministic rates for general non-convex functions under a stronger assumption on the growth of the stochastic gradients [73, 83]. Under this assumption and an upper bound (that requires knowledge of the "Lipschitz" constant) on the maximum step size, we prove that SGD with Armijo line-search can achieve the deterministic rate for general non-convex functions (Section 4). Note that these are the first convergence rates for SGD with line-search in the interpolation setting for both convex and non-convex functions.

Moving beyond SGD, in Section 5 we consider the stochastic extra-gradient (SEG) method [24, 31, 36, 41, 55] used to solve general variational inequalities [27]. These problems encompass both convex minimization and saddle point problems arising in robust supervised learning [8, 84] and learning with non-separable losses or regularizers [4, 34]. In the interpolation setting, we show that a variant of SEG [24] with a "Lipschitz" line-search convergences linearly when minimizing an

important class of non-convex functions [16, 40, 44, 78, 79] satisfying the restricted secant inequality (RSI). Moreover, in Appendix E, we prove that the same algorithm results in linear convergence for both strongly convex-concave and bilinear saddle point problems satisfying interpolation.

In Section 6, we give heuristics to use large step-sizes and integrate acceleration with our line-search techniques, which improves practical performance of the proposed methods. We compare our algorithms against numerous optimizers [21, 39, 51, 53, 60, 68] on a synthetic matrix factorization problem (Section 7.2), convex binary-classification problems using radial basis function (RBF) kernels (Section 7.3), and non-convex multi-class classification problems with deep neural networks (Section 7.4). We observe that when interpolation is (approximately) satisfied, the proposed methods are robust and have competitive performance across models and datasets. Moreover, SGD with Armijo line-search results in both faster convergence and better generalization performance for classification using deep networks. Finally, in Appendix G.2, we evaluate SEG with line-search for synthetic bilinear saddle point problems. The code to reproduce our results can be found at `https://github.com/IssamLaradji/sls`.

We note that in concurrent work to ours, Berrada et al. [10] propose adaptive step-sizes for SGD on convex, finite-sum loss functions under an ϵ-interpolation condition. Unlike our approach, ϵ-interpolation requires knowledge of a lower bound on the global minimum and only guarantees approximate convergence to a stationary point. Moreover, in order to obtain linear convergence rates, they assume μ-strong-convexity of *each* individual function. This assumption with ϵ-interpolation reduces the finite-sum optimization to minimization of *any single* function in the finite sum.

2 Assumptions

We aim to minimize a differentiable function f assuming access to noisy stochastic gradients of the function. We focus on the common machine learning setting where the function f has a *finite-sum structure* meaning that $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$. Here n is equal to the number of points in the training set and the function f_i is the loss function for the training point i. Depending on the model, f can either be strongly-convex, convex, or non-convex. We assume that f is lower-bounded by some value f^* and that f is L-smooth [56] implying that the gradient ∇f is L-Lipschitz continuous.

We assume that the model is able to interpolate the data and use this property to derive convergence rates. Formally, interpolation requires that the gradient with respect to *each* point converges to zero at the optimum, implying that if the function f is minimized at w^* and thus $\nabla f(w^*) = 0$, then for all functions f_i we have that $\nabla f_i(w^*) = 0$. For example, interpolation is exactly satisfied when using a linear model with the squared hinge loss for binary classification on linearly separable data.

3 Stochastic Gradient Descent for Convex Functions

Stochastic gradient descent (SGD) computes the gradient of the loss function corresponding to one or a mini-batch of randomly (typically uniformly) chosen training examples i_k in iteration k. It then performs a descent step as $w_{k+1} = w_k - \eta_k \nabla f_{ik}(w_k)$, where w_{k+1} and w_k are the SGD iterates, η_k is the step-size and $\nabla f_{ik}(\cdot)$ is the (average) gradient of the loss function(s) chosen at iteration k. Each stochastic gradient $\nabla f_{ik}(w)$ is assumed to be unbiased, implying that $\mathbb{E}_i\left[\nabla f_i(w)\right] = \nabla f(w)$ for all w. We now describe the Armijo line-search method to set the step-size in each iteration.

3.1 Armijo line-search

Armijo line-search [3] is a standard method for setting the step-size for gradient descent in the deterministic setting [59]. We adapt it to the stochastic case as follows: at iteration k, the Armijo line-search selects a step-size satisfying the following condition:

$$f_{ik}\left(w_k - \eta_k \nabla f_{ik}(w_k)\right) \leq f_{ik}(w_k) - c \cdot \eta_k \left\|\nabla f_{ik}(w_k)\right\|^2. \tag{1}$$

Here, $c > 0$ is a hyper-parameter. Note that the above line-search condition uses the function and gradient values *of the mini-batch* at the current iterate w_k. Thus, compared to SGD, checking this condition only makes use of additional mini-batch function (and not gradient) evaluations. In the context of deep neural networks, this corresponds to extra forward passes on the mini-batch.

In our theoretical results, we assume that there is a maximum step-size $\eta_{\max}$ from which the line-search starts in *each* iteration k and that we choose the largest step-size η_k (less than or equal to $\eta_{\max}$) satisfying (1). In practice, backtracking line-search is a common way to ensure that Equation 1 is satisfied. Starting from $\eta_{\max}$, backtracking iteratively decreases the step-size by a constant factor

β until the line-search succeeds (see Algorithm 1). Suitable strategies for *resetting* the step-size can avoid backtracking in the majority of iterations and make the step-size selection procedure efficient. We describe such strategies in Section 6. With resetting, we required (on average) only one additional forward pass on the mini-batch per iteration when training a standard deep network model (Section 7.4). Empirically, we observe that the algorithm is robust to the choice of both c and $\eta_{\max}$; setting c to a small constant and $\eta_{\max}$ to a large value consistently results in good performance.

We bound the chosen step-size in terms of the properties of the function(s) selected in iteration k.

Lemma 1. *The step-size η_k returned by the Armijo line-search and constrained to lie in the $(0, \eta_{max}]$ range satisfies the following inequality,*

$$\eta_k \geq \min\left\{\frac{2\,(1-c)}{L_{ik}}, \eta_{max}\right\}, \tag{2}$$

where L_{ik} is the Lipschitz constant of ∇f_{i_k}.

The proof is in Appendix A and follows the deterministic case [59]. Note that Equation (1) holds for all smooth functions (for small-enough η_k), does not require convexity, and guarantees backtracking line-search will terminate at a non-zero step-size. The parameter c controls the "aggressiveness" of the algorithm; small c values encourage a larger step-size. For a sufficiently large $\eta_{\max}$ and $c \leq 1/2$, the step-size is at least as large as $1/L_{ik}$, which is the constant step-size used in the interpolation setting [73, 83]. In practice, we expect these larger step-sizes to result in improved performance. In Appendix A, we also give upper bounds on η_k if the function f_{i_k} satisfies the Polyak-Lojasiewicz (PL) inequality [37, 65] with constant μ_{ik}. PL is a weaker condition than strong-convexity and does not require convexity. In this case, η_k is upper-bounded by the minimum of $\eta_{\max}$ and $1/(2c \cdot \mu_{ik})$. If we use a backtracking line-search that multiplies the step-size by β until (1) holds, the step-size will be smaller by at most a factor of β (we do not include this dependence in our results).

3.2 Convergence rates

In this section, we characterize the convergence rate of SGD with Armijo line-search in the strongly-convex and convex cases. The theorems below are proved in Appendix B and Appendix C respectively.

Theorem 1 (Strongly-Convex)**.** *Assuming (a) interpolation, (b) L_i-smoothness, (c) convexity of f_i's, and (d) μ strong-convexity of f, SGD with Armijo line-search with $c = 1/2$ in Eq. 1 achieves the rate:*

$$\mathbb{E}\left[\|w_T - w^*\|^2\right] \leq \max\left\{\left(1 - \frac{\bar{\mu}}{L_{max}}\right), (1 - \bar{\mu}\,\eta_{max})\right\}^T \|w_0 - w^*\|^2.$$

Here $\bar{\mu} = \sum_{i=1}^n \mu_i/n$ is the average strong-convexity of the finite sum and $L_{max} = \max_i L_i$ is the maximum smoothness constant in the f_i's.

In contrast to the previous results [52, 73, 83] that depend on μ, the above linear rate depends on $\bar{\mu} \leq \mu$. Note that unlike Berrada et al. [10], we do not require that *each* f_i is strongly convex, but for $\bar{\mu}$ to be non-zero we still require that *at least one* of the f_i's is strongly-convex.

Theorem 2 (Convex)**.** *Assuming (a) interpolation, (b) L_i-smoothness and (c) convexity of f_i's, SGD with Armijo line-search for all $c > 1/2$ in Equation 1 and iterate averaging achieves the rate:*

$$\mathbb{E}\left[f(\bar{w}_T) - f(w^*)\right] \leq \frac{c \cdot \max\left\{\frac{L_{max}}{2\,(1-c)}, \frac{1}{\eta_{max}}\right\}}{(2c-1)\,T} \|w_0 - w^*\|^2.$$

Here, $\bar{w}_T = \frac{\left[\sum_{i=1}^T w_i\right]}{T}$ is the averaged iterate after T iterations and $L_{max} = \max_i L_i$.

In particular, setting $c = 2/3$ implies that $\mathbb{E}\left[f(\bar{w}_T) - f(w^*)\right] \leq \frac{\max\left\{3\,L_{\max}, \frac{2}{\eta_{\max}}\right\}}{T} \|w_0 - w^*\|^2$. These are the first rates for SGD with line-search in the interpolation setting and match the corresponding rates for full-batch gradient descent on strongly-convex and convex functions. This shows SGD attains fast convergence under interpolation *without* explicit knowledge of the Lipschitz constant. Next, we use the above line-search to derive convergence rates of SGD for non-convex functions.

4 Stochastic Gradient Descent for Non-convex Functions

To prove convergence results in the non-convex case, we additionally require the strong growth condition (SGC) [73, 83] to hold. The function f satisfies the SGC with constant ρ, if $\mathbb{E}_i \|\nabla f_i(w)\|^2 \leq$

$\rho \|\nabla f(w)\|^2$ holds for any point w. This implies that if $\nabla f(w) = 0$, then $\nabla f_i(w) = 0$ for *all* i. Thus, functions satisfying the SGC necessarily satisfy the interpolation property. The SGC holds for all smooth functions satisfying a PL condition [83]. Under the SGC, we show that by upper-bounding the maximum step-size η_{max}, SGD with Armijo line-search achieves an $O(1/T)$ convergence rate.

Theorem 3 (Non-convex). *Assuming (a) the SGC with constant ρ and (b) L_i-smoothness of f_i's, SGD with Armijo line-search in Equation 1 with $c = 1/2$ and setting $\eta_{max} = 3/2\rho L$ achieves the rate:*

$$\min_{k=0,\ldots,T-1} \mathbb{E} \|\nabla f(w_k)\|^2 \leq \frac{4\ L_{max}}{T} \left(\frac{2\rho}{3} + 1\right) (f(w_0) - f(w^*)) .$$

We prove Theorem 3 in Appendix D. The result requires knowledge of $\rho\ L_{\max}$ to bound the maximum step-size, which is less practically appealing. It is not immediately clear how to relax this condition and we leave it for future work. However, in the next section, we show that if the non-convex function satisfies a specific curvature condition, a modified stochastic extra-gradient algorithm can achieve a linear rate under interpolation without additional assumptions or knowledge of the Lipschitz constant.

5 Stochastic Extra-Gradient Method

In this section, we use a modified stochastic extra-gradient (SEG) method for convex and non-convex minimization. For finite-sum minimization, stochastic extra-gradient (SEG) has the following update:

$$w'_k = w_k - \eta_k \nabla f_{ik}(w_k)\ ,\ w_{k+1} = w_k - \eta_k \nabla f_{ik}(w'_k). \tag{3}$$

It computes the gradient at an extrapolated point w'_k and uses it in the update from the current iterate w_k. Note that using the same sample i_k and step-size η_k for both steps [24] is important for the subsequent theoretical results. We now describe a "Lipschitz" line-search strategy [31, 32, 38] in order to automatically set the step-size for SEG.

5.1 Lipschitz line-search

The "Lipschitz" line-search has been used by previous work in the deterministic [32, 38] and the variance reduced settings [30]. It selects a step-size η_k that satisfies the following condition:

$$\|\nabla f_{ik}(w_k - \eta_k \nabla f_{ik}(w_k)) - \nabla f_{ik}(w_k)\| \leq c\ \|\nabla f_{ik}(w_k)\| . \tag{4}$$

As before, we use backtracking line-search starting from the maximum value of $\eta_{\max}$ to ensure that the chosen step-size satisfies the above condition. If the function f_{ik} is L_{ik}-smooth, the step-size returned by the Lipschitz line-search satisfies $\eta_k \geq \min\left\{c/L_{ik}, \eta_{\max}\right\}$. Like the Armijo line-search in Section 3, the Lipschitz line-search does not require knowledge of the Lipschitz constant. Unlike the line-search strategy in the previous sections, checking condition (4) requires computing the gradient at a prospective extrapolation point. We now prove convergence rates for SEG with Lipschitz line-search for both convex and a special class of non-convex problems.

5.2 Convergence rates for minimization

For the next result, we assume that each function $f_i(\cdot)$ satisfies the restricted secant inequality (RSI) with constant μ_i, implying that for all w, $\langle \nabla f_i(w), w - w^* \rangle \geq \mu_i \|w - w^*\|^2$. RSI is a weaker condition than strong-convexity. With additional assumptions, RSI is satisfied by important non-convex models such as single hidden-layer neural networks [40, 44, 78], matrix completion [79] and phase retrieval [16]. Under interpolation, we show SEG results in linear convergence for functions satisfying RSI. In particular, we obtain the following guarantee:

Theorem 4 (Non-convex + RSI). *Assuming (a) interpolation, (b) L_i-smoothness, and (c) μ_i-RSI of f_i's, SEG with Lipschitz line-search in Eq. 4 with $c = 1/4$ and $\eta_{max} \leq \min_i 1/4\mu_i$ achieves the rate:*

$$\mathbb{E}\left[\|w_T - \mathcal{P}_{\mathcal{X}^*}[w_T]\|^2\right] \leq \max\left\{\left(1 - \frac{\bar{\mu}}{4\ L_{max}}\right), (1 - \eta_{max}\ \bar{\mu})\right\}^T \|w_0 - \mathcal{P}_{\mathcal{X}^*}[w_0]\|^2 ,$$

where $\bar{\mu} = \frac{\sum_{i=1}^n \mu_i}{n}$ is the average RSI constant of the finite sum and $\mathcal{X}^*$ is the non-empty set of optimal solutions. The operation $\mathcal{P}_{\mathcal{X}^*}[w]$ denotes the projection of w onto $\mathcal{X}^*$.

See Appendix E.2 for proof. Similar to the result of Theorem 1, the rate depends on the average RSI constant. Note that we do not require explicit knowledge of the Lipschitz constant to achieve the above rate. The constraint on the maximum step-size is mild since the minimum μ_i is typically small, thus allowing for large step-sizes. Moreover, Theorem 4 improves upon the $\left(1 - \mu^2/L^2\right)$ rate

Algorithm 1 SGD+Armijo(f, w_0, $\eta_{\max}$, b, c, β, γ, opt)

for $k = 0, \ldots, T$ **do**
  $i_k \leftarrow$ sample mini-batch of size b
  $\eta \leftarrow \texttt{reset}(\eta, \eta_{\max}, \gamma, b, k, \texttt{opt})/\beta$
  repeat
    $\eta \leftarrow \beta \cdot \eta$
    $\tilde{w_k} \leftarrow w_k - \eta \nabla f_{ik}(w_k)$
  until $f_{ik}(\tilde{w_k}) \leq f_{ik}(w_k) - c \cdot \eta \left\|\nabla f_{ik}(w_k)\right\|^2$
  $w_{k+1} \leftarrow \tilde{w_k}$
end for
return w_{k+1}

Algorithm 2 reset(η, $\eta_{\max}$, γ, b, k, opt)

if k = 1 **then**
  return $\eta_{\max}$
else if opt $= 0$ **then**
  $\eta \leftarrow \eta$
else if opt $= 1$ **then**
  $\eta \leftarrow \eta_{\max}$
else if opt $= 2$ **then**
  $\eta \leftarrow \eta \cdot \gamma^{b/n}$
end if
return η

Figure 1: Algorithm 1 gives pseudo-code for SGD with Armijo line-search. Algorithm 2 implements several heuristics (by setting opt) for resetting the step-size at each iteration.

obtained using constant step-size SGD [5, 83]. In Appendix E.2, we show that the same rate can be attained by SEG with a constant step-size. In Appendix E.3, we show that under interpolation, SEG with Lipschitz line-search also achieves the desired $O(1/T)$ rate for convex functions.

5.3 Convergence rates for saddle point problems

In Appendix E.4, we use SEG with Lipschitz line-search for a class of saddle point problems of the form $\min_{u \in U} \max_{v \in \mathcal{V}} \phi(u, v)$. Here $\mathcal{U}$ and $\mathcal{V}$ are the constraint sets for the variables u and v respectively. In Theorem 6 in Appendix E.4, we show that under interpolation, SEG with Lipschitz line-search results in linear convergence for functions $\phi(u, v)$ that are strongly-convex in u and strongly-concave in v. The required conditions are satisfied for robust optimization [84] with expressive models capable of interpolating the data. Furthermore, the interpolation property can be used to improve the convergence for a bilinear saddle-point problem [24, 25, 54, 85]. In Theorem 7 in Appendix E.5, we show that SEG with Lipschitz line-search results in linear convergence under interpolation. We empirically validate this claim with simple synthetic experiments in Appendix G.2.

6 Practical Considerations

In this section, we give heuristics to use larger step-sizes across iterations and discuss ways to use common acceleration schemes with our line-search techniques.

6.1 Using larger step-sizes

Recall that our theoretical analysis assumes that the line-search in *each* iteration starts from a global maximum step-size $\eta_{\max}$. However, in practice, this strategy increases the amount of backtracking and consequently the algorithm's runtime. A simple alternative is to initialize the line-search in each iteration to the step-size selected in the previous iteration η_{k-1}. With this strategy, the step-size can not increase and convergence is slowed in practice (it takes smaller steps than necessary). To alleviate these problems, we consider increasing the step-size across iterations by initializing the backtracking at iteration k with $\eta_{k-1} \cdot \gamma^{b/n}$ [71, 72], where b is the size of the mini-batch and $\gamma > 1$ is a tunable parameter. These heuristics correspond to the options used in Algorithm 2.

We also consider the Goldstein line-search that uses additional function evaluations to check the curvature condition $f_{ik}\left(w_k - \eta_k \nabla f_{ik}(w_k)\right) \geq f_{ik}(w_k) - (1-c) \cdot \eta_k \left\|\nabla f_{ik}(w_k)\right\|^2$ and increases the step-size if it is not satisfied. Here, c is the constant in Equation 1. The resulting method decreases the step-size if the Armijo condition is not satisfied and increases it if the curvature condition does not hold. Algorithm 3 in Appendix H gives pseudo-code for SGD with the Goldstein line-search.

6.2 Acceleration

In practice, augmenting stochastic methods with some form of momentum or acceleration [57, 64] often results in faster convergence [80]. Related work in this context includes algorithms specifically designed to achieve an accelerated rate of convergence in the stochastic setting [1, 23, 46]. Unlike these works, we propose simple ways of using either Polyak [64] or Nesterov [57] acceleration with the proposed line-search techniques. In both cases, similar to adaptive methods using momentum [80], we use SGD with Armijo line-search to determine η_k and then use it directly within the acceleration scheme. When using Polyak momentum, the effective update can be given as: $w_{k+1} = w_k -$

$\eta_k \nabla f_{ik}(w_k) + \alpha(w_k - w_{k-1})$, where α is the momentum factor. This update rule has been used with a constant step-size and proven to obtain linear convergence rates on the *generalization error* for quadratic functions under an interpolation condition [48, 49]. For Nesterov acceleration, we use the variant for the convex case [57] (which has no additional hyper-parameters) with our line-search. The pseudo-code for using these methods with the Armijo line-search is given in Appendix H.

7 Experiments

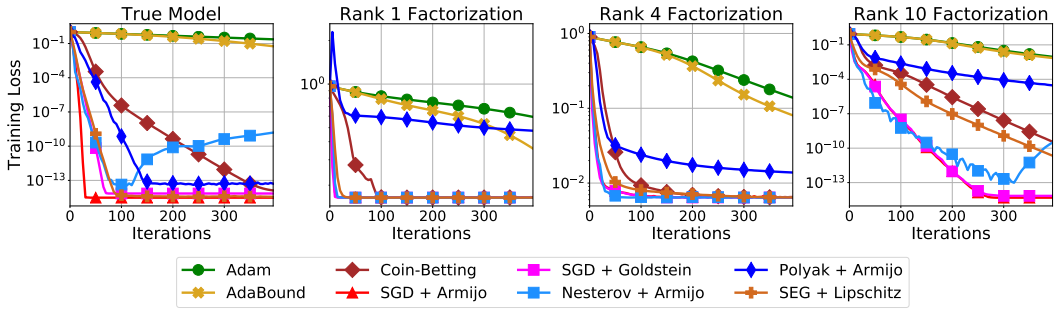

Figure 2: Matrix factorization using the true model and rank 1, 4, 10 factorizations. Rank 1 factorization is under-parametrized, while ranks 4 and 10 are over-parametrized. Rank 10 and the true model satisfy interpolation.

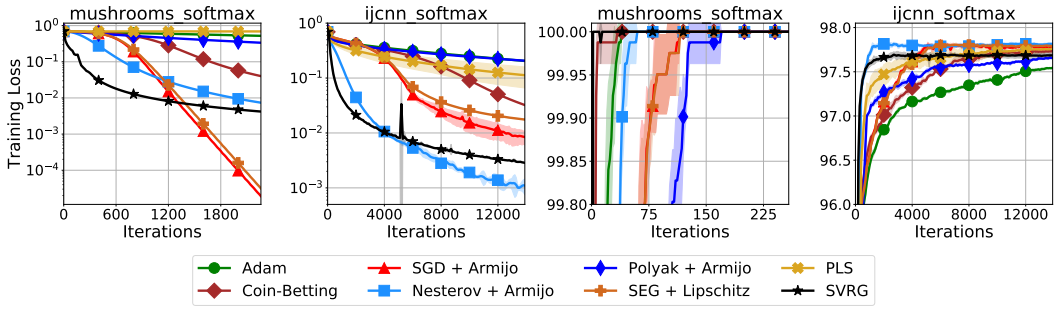

Figure 3: Binary classification using a softmax loss and RBF kernels for the mushrooms and ijcnn datasets. Mushrooms is linear separable in kernel-space with the selected kernel bandwidths while ijcnn is *not*. Overall, we observe fast convergence of SGD + Armijo, Nesterov + Armijo, and SEG + Lipschitz for both datasets.

We describe the experimental setup in Section 7.1. In Section 7.2, we present synthetic experiments to show the benefits of over-parametrization. In Sections 7.3 and 7.4, we showcase the convergence and generalization performance of our methods for kernel experiments and deep networks, respectively.

7.1 Experimental setup

We benchmark five configurations of the proposed line-search methods: SGD with (1) Armijo line-search with resetting the initial step-size (Algorithm 1 using option 2 in Algorithm 2), (2) Goldstein line-search (Algorithm 3), (3) Polyak momentum (Algorithm 5), (4) Nesterov acceleration (Algorithm 6), and (5) SEG with Lipschitz line-search (Algorithm 4) with option 2 to reset the step-size. Appendix F gives additional details on our experimental setup and the default hyper-parameters used for the proposed line-search methods. We compare our methods against Adam [39], which is the most common adaptive method, and other methods that report better performance than Adam: coin-betting [60], L4[1] [68], and Adabound [51]. We use the default learning rates for the competing methods. Unless stated otherwise, our results are averaged across 5 independent runs.

[1]L4 applied to momentum SGD (L4 Mom) in `https://github.com/iovdin/l4-pytorch` was unstable in our experiments and we omit it from the main paper.

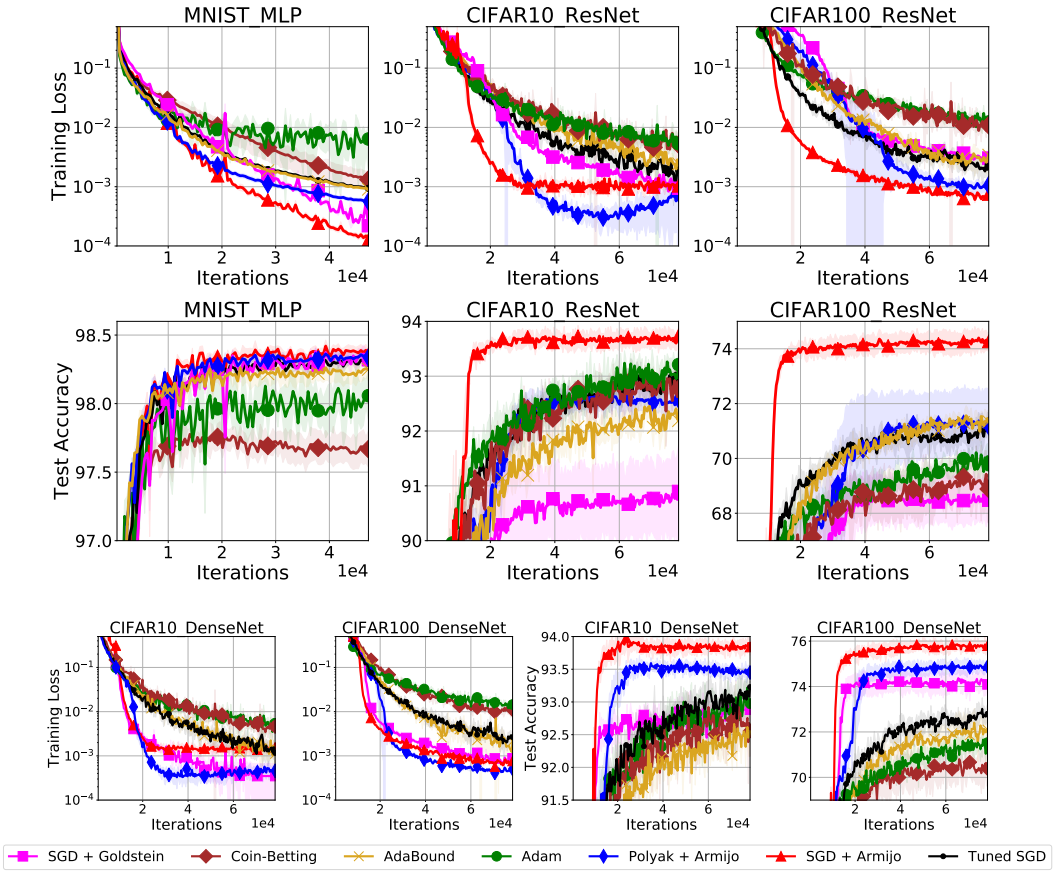

Figure 4: Multi-class classification using softmax loss and (top) an MLP model for MNIST; ResNet model for CIFAR-10 and CIFAR-100 (bottom) DenseNet model for CIFAR-10 and CIFAR-100.

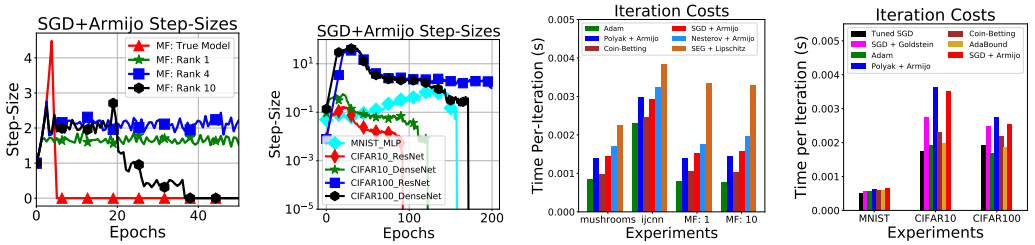

Figure 5: (Left) Variation in step-sizes for SGD+Armijo for the matrix factorization problem and classification with deep neural networks. (Right) Average time per iteration.

7.2 Synthetic experiment

We examine the effect of over-parametrization on convergence rates for the non-convex regression problem: $\min_{W_1,W_2} \mathbb{E}_{x\sim N(0,I)} \|W_2W_1x - Ax\|^2$. This is equivalent to a matrix factorization problem satisfying RSI [79] and has been proposed as a challenging benchmark for gradient descent methods [66]. Following Rolínek et al. [68], we choose $A \in \mathbb{R}^{10\times 6}$ with condition number $\kappa(A) = 10^{10}$ and generate a fixed dataset of 1000 samples. Unlike the previous work, we consider stochastic optimization and control the model's expressivity via the rank k of the matrix factors $W_1 \in \mathbb{R}^{k\times 6}$ and $W_2 \in \mathbb{R}^{10\times k}$. Figure 2 shows plots of training loss (averaged across 20 runs) for the true data-generating model, and using factors with rank $k \in \{1, 4, 10\}$.

We make the following observations: (i) for $k = 4$ (where interpolation *does not hold*) the proposed methods converge quicker than other optimizers but all methods reach an artificial optimization floor, (ii) using $k = 10$ yields an over-parametrized model where SGD with both Armijo and Goldstein

line-search converge linearly to machine precision, (iii) SEG with Lipschitz line-search obtains fast convergence according to Theorem 4, and (iv) adaptive-gradient methods stagnate in all cases. These observations validate our theoretical results and show that over-parameterization and line-search can allow for fast, "painless" optimization using SGD and SEG.

7.3 Binary classification with kernels

We consider convex binary classification using RBF kernels without regularization. We experiment with four standard datasets: mushrooms, rcv1, ijcnn, and w8a from LIBSVM [14]. The mushrooms dataset satisfies the interpolation condition with the selected kernel bandwidths, while ijcnn, rcv1, and w8a do not. For these experiments we also compare against a standard VR method (SVRG) [35] and probabilistic line-search (PLS) [53].[2] Figure 3 shows the training loss and test accuracy on mushrooms and ijcnn for the different optimizers with softmax loss. Results for rcv1 and w8a are given in Appendix G.3. We make the following observations: (i) SGD + Armijo, Nesterov + Armijo, and SEG + Lipschitz perform the best and are comparable to hand-tuned SVRG. (ii) The proposed line-search methods perform well on ijcnn even though it is not separable in kernel space. This demonstrates some robustness to violations of the interpolation condition.

7.4 Multi-class classification using deep networks

We benchmark the convergence rate and generalization performance of our line-search methods on standard deep learning experiments. We consider non-convex minimization for multi-class classification using deep network models on the MNIST, CIFAR10, and CIFAR100 datasets. Our experimental choices follow the setup in Luo et al. [51]. For MNIST, we use a 1 hidden-layer multi-layer perceptron (MLP) of width 1000. For CIFAR10 and CIFAR100, we experiment with the standard image-classification architectures: ResNet-34 [28] and DenseNet-121 [29]. We also compare to the best performing constant step-size SGD with the step-size selected by grid search.

From Figure 4, we observe that: (i) SGD with Armijo line-search consistently leads to the best performance in terms of both the training loss and test accuracy. It also converges to a good solution *much* faster when compared to the other methods. (ii) The performance of SGD with line-search and Polyak momentum is always better than "tuned" constant step-size SGD and Adam, whereas that of SGD with Goldstein line-search is competitive across datasets. We omit Nesterov + Armijo as it unstable and diverges and omit SEG since it resulted in slower convergence and worse performance.

We also verify that our line-search methods do not lead to excessive backtracking and function evaluations. Figure 5 (right) shows the cost per iteration for the above experiments. Our line-searches methods are only marginally slower than Adam and converge much faster. In practice, we observed SGD+Armijo uses only one additional function evaluation on average. Figure 5 (left) shows the evolution of step-sizes for SGD+Armijo in our experiments. For deep neural networks, SGD+Armijo automatically finds a step-size schedule resembling cosine-annealing [50]. In Appendix G.1, we evaluate and compare the hyper-parameter sensitivity of Adam, constant step-size SGD, and SGD with Armijo line-search on CIFAR10 with ResNet-34. While SGD is sensitive to the choice of the step-size, the performance of SGD with Armijo line-search is robust to the value of c in the $[0.1, 0.5]$ range. There is virtually no effect of $\eta_{\max}$, since the correct range of step-sizes is found in early iterations.

8 Conclusion

We showed that under the interpolation condition satisfied by modern over-parametrized models, simple line-search techniques for classic SGD and SEG lead to fast convergence in both theory and practice. For future work, we hope to strengthen our results for non-convex minimization using SGD with line-search and study stochastic momentum techniques under interpolation. More generally, we hope to utilize the rich literature on line-search and trust-region methods to improve stochastic optimization for machine learning.

[2] PLS is impractical for deep networks since it requires the second moment of the mini-batch gradients and needs GP model inference for every line-search evaluation.

Acknowledgments

We would like to thank Yifan Sun and Nicolas Le Roux for insightful discussions. AM is supported by the NSERC CGS M award. IL is funded by the UBC Four-Year Doctoral Fellowships (4YF), This research was also partially supported by the Canada CIFAR AI Chair Program, the CIFAR LMB Program, by a Google Focused Research award, by an IVADO postdoctoral scholarship (for SV), by a Borealis AI fellowship (for GG), by the Canada Excellence Research Chair in "Data Science for Realtime Decision-making" and by the NSERC Discovery Grants RGPIN-2017-06936 and 2015-06068.

References

[1] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *ACM SIGACT Symposium on Theory of Computing*, 2017.

[2] Luís Almeida. Parameter adaptation in stochastic optimization. *On-line learning in neural networks*, 1998.

[3] Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 1966.

[4] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 2012.

[5] Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of SGD in non-convex over-parametrized learning. *arXiv preprint arXiv:1811.02564*, 2018.

[6] Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. In *ICLR*, 2017.

[7] Mikhail Belkin, Alexander Rakhlin, and Alexandre B. Tsybakov. Does data interpolation contradict statistical optimality? In *AISTATS*, 2019.

[8] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[9] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*. Springer, 2012.

[10] Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Training neural networks for and by interpolation. *arXiv preprint arXiv:1906.05661*, 2019.

[11] Jose Blanchet, Coralia Cartis, Matt Menickelly, and Katya Scheinberg. Convergence rate analysis of a stochastic trust region method via supermartingales. *Informs Journal on Optimization*, 2019.

[12] Richard H Byrd, Gillian M Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical programming*, 2012.

[13] Volkan Cevher and Bang Công Vũ. On the linear convergence of the stochastic gradient method with constant step-size. *Optimization Letters*, 2018.

[14] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[15] Tatjana Chavdarova, Gauthier Gidel, François Fleuret, and Simon Lacoste-Julien. Reducing noise in GAN training with variance reduced extragradient. In *NeurIPS*, pages 391–401, 2019.

[16] Yuxin Chen and Emmanuel Candes. Solving random quadratic systems of equations is nearly as easy as solving linear systems. In *NeurIPS*, 2015.

[17] Soham De, Abhay Yadav, David Jacobs, and Tom Goldstein. Big batch SGD: Automated inference using adaptive batch sizes. *arXiv preprint arXiv:1610.05792*, 2016.

[18] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NeurIPS*, 2014.

[19] Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *NeurIPS*, pages 1753–1763, 2019.

[20] Bernard Delyon and Anatoli Juditsky. Accelerated stochastic approximation. *SIAM Journal on Optimization*, 1993.

[21] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 2011.

[22] Michael P Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 2012.

[23] Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*, 2015.

[24] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. In *ICLR*, 2019.

[25] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[26] Serge Gratton, Clément W Royer, Luís N Vicente, and Zaikun Zhang. Complexity and global rates of trust-region methods based on probabilistic models. *IMA Journal of Numerical Analysis*, 38(3):1579–1597, 2017.

[27] Patrick T Harker and Jong-Shi Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. *Mathematical programming*, 1990.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[30] Alfredo N Iusem, Alejandro Jofré, Roberto I Oliveira, and Philip Thompson. Variance-based extragradient methods with line search for stochastic variational inequalities. *SIAM Journal on Optimization*, 2019.

[31] AN Iusem, Alejandro Jofré, Roberto I Oliveira, and Philip Thompson. Extragradient method with variance reduction for stochastic variational inequalities. *SIAM Journal on Optimization*, 2017.

[32] AN Iusem and BF Svaiter. A variant of korpelevich's method for variational inequalities with a new search strategy. *Optimization*, 1997.

[33] Prateek Jain, Sham Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent for least squares regression. In *COLT*, 2018.

[34] Thorsten Joachims. A support vector method for multivariate performance measures. In *ICML*, 2005.

[35] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NeurIPS*, 2013.

[36] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 2011.

[37] 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*, 2016.

[38] Evgenii Nikolaevich Khobotov. Modification of the extra-gradient method for solving variational inequalities and certain optimization problems. *USSR Computational Mathematics and Mathematical Physics*, 1987.

[39] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[40] Robert Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does SGD escape local minima? In *ICML*, 2018.

[41] GM Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 1976.

[42] Nataša Krejić and Nataša Krklec. Line search methods with variable sample size for unconstrained optimization. *Journal of Computational and Applied Mathematics*, 2013.

[43] Harold J Kushner and Jichuan Yang. Stochastic approximation with averaging and feedback: Rapidly convergent" on-line" algorithms. *IEEE Transactions on Automatic Control*, 1995.

[44] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with ReLU activation. In *NeurIPS*, 2017.

[45] Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel" ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387*, 2018.

[46] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *NeurIPS*, 2015.

[47] Chaoyue Liu and Mikhail Belkin. Accelerating stochastic training for over-parametrized learning. *arXiv preprint arXiv:1810.13395*, 2019.

[48] Nicolas Loizou and Peter Richtárik. Linearly convergent stochastic heavy ball method for minimizing generalization error. *arXiv preprint arXiv:1710.10737*, 2017.

[49] Nicolas Loizou and Peter Richtárik. Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods. *arXiv preprint arXiv:1712.09677*, 2017.

[50] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[51] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *ICLR*, 2019.

[52] Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *ICML*, 2018.

[53] Maren Mahsereci and Philipp Hennig. Probabilistic line searches for stochastic optimization. *JMLR*, 2017.

[54] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of GANs. In *NeurIPS*, 2017.

[55] Arkadi Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 2004.

[56] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 2009.

[57] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 2013.

[58] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2004.

[59] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[60] Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. In *NeurIPS*, 2017.

[61] Balamurugan Palaniappan and Francis Bach. Stochastic variance reduction methods for saddle-point problems. In *NeurIPS*, 2016.

[62] Courtney Paquette and Katya Scheinberg. A stochastic line search method with convergence rate analysis. *arXiv preprint arXiv:1807.07994*, 2018.

[63] VP Plagianakos, GD Magoulas, and MN Vrahatis. Learning rate adaptation in stochastic gradient descent. In *Advances in convex analysis and global optimization*. Springer, 2001.

[64] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 1964.

[65] Boris Teodorovich Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 1963.

[66] Ali Rahimi and Ben Recht. Reflections on random kitchen sinks, 2017.

[67] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *ICLR*, 2019.

[68] Michal Rolinek and Georg Martius. L4: practical loss-based stepsize adaptation for deep learning. In *NeurIPS*, 2018.

[69] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 1998.

[70] Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *ICML*, 2013.

[71] 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.

[72] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 2017.

[73] Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

[74] Alice Schoenauer-Sebag, Marc Schoenauer, and Michèle Sebag. Stochastic gradient descent: Going as fast as possible but not faster. *arXiv preprint arXiv:1709.01427*, 2017.

[75] Nicol Schraudolph. Local gain adaptation in stochastic gradient descent. 1999.

[76] Fanhua Shang, Yuanyuan Liu, Kaiwen Zhou, James Cheng, Kelvin Ng, and Yuichi Yoshida. Guaranteed sufficient decrease for stochastic variance reduced gradient optimization. In *AISTATS*, 2018.

[77] S Shao and Percy Yip. Rates of convergence of adaptive step-size of stochastic approximation algorithms. *Journal of mathematical analysis and applications*, 2000.

[78] Mahdi Soltanolkotabi, Adel Javanmard, and Jason Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 2018.

[79] Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 2016.

[80] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[81] Conghui Tan, Shiqian Ma, Yu-Hong Dai, and Yuqiu Qian. Barzilai-Borwein step size for stochastic gradient descent. In *NeurIPS*, 2016.

[82] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera: Neural networks for machine learning*, 2012.

[83] Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *AISTATS*, 2019.

[84] Junfeng Wen, Chun-Nam Yu, and Russell Greiner. Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In *ICML*, 2014.

[85] Abhay Yadav, Sohil Shah, Zheng Xu, David Jacobs, and Tom Goldstein. Stabilizing adversarial nets with prediction methods. *arXiv preprint arXiv:1705.07364*, 2017.

[86] Jin Yu, Douglas Aberdeen, and Nicol Schraudolph. Fast online policy gradient learning with smd gain vector adaptation. In *NeurIPS*, 2006.

[87] Matthew Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[88] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

[89] Jian Zhang and Ioannis Mitliagkas. Yellowfin and the art of momentum tuning. *arXiv preprint arXiv:1706.03471*, 2017.

A Proof of Lemma 1

Proof.

From the smoothness of f_{ik} and the update rule, the following inequality holds for all values of η_k.

$$f_{ik}(w_{k+1}) \leq f_{ik}(w_k) - \left(\eta_k - \frac{L_{ik}\eta_k^2}{2}\right) \|\nabla f_{ik}(w_k)\|^2$$

The step-size returned by the line-search satisfies Equation 1, implying that,

$$f_{ik}(w_{k+1}) \leq f_{ik}(w_k) - c\,\eta_k \, \|\nabla f_{ik}(w_k)\|^2$$

Using the above relations, the step-size returned by the line-search satisfies the following inequality,

$$\begin{aligned} c\,\eta_k &\geq \left(\eta_k - \frac{L_{ik}\eta_k^2}{2}\right) \\ \implies \eta_k &\geq \frac{2\,(1-c)}{L_{ik}} \end{aligned}$$

This gives us a lower bound on η_k.

Let us now upper-bound η_k. Using Equation 1,

$$\begin{aligned} \implies \eta_k &\leq \frac{[f_{ik}(w_k) - f_{ik}(w_{k+1})]}{c\,\|\nabla f_{ik}(x_k)\|^2} \\ \eta_k &\leq \frac{[f_{ik}(w_k) - f_{ik}(w^*) + f_{ik}(w^*) - f_{ik}(w_{k+1})]}{c\,\|\nabla f_{ik}(w_k)\|^2} \end{aligned}$$

By the interpolation condition, $f_{ik}(w^*) \leq f_{ik}(w)$ for all functions i_k and points w, $\implies f_{ik}(w^*) - f_{ik}(w_{k+1}) \leq 0$

$$\implies \eta_k \leq \frac{[f_{ik}(w_k) - f_{ik}(w^*)]}{c\,\|\nabla f_{ik}(w_k)\|^2}$$

By definition, $\eta_k \leq \eta_{\max}$. Furthermore, if we each $f_{ik}(\cdot)$ satisfies the either strong-convexity or the Polyak-Lojasiewicz (PL) inequality [37, 65] (which is weaker than strong-convexity and does not require convexity), then,

$$\begin{aligned} f_{ik}(w_k) - f_{ik}(w_k^*) &\leq \frac{1}{2\mu_{ik}} \|\nabla f_{ik}(w_k)\|^2 \\ \implies f_{ik}(w_k) - f_{ik}(w_{k+1}) &\leq \frac{1}{2\mu_{ik}} \|\nabla f_{ik}(w_k)\|^2 \\ f_{ik}(w_k) - f_{ik}(w^*) &\leq \frac{1}{2\mu_{ik}} \|\nabla f_{ik}(w_k)\|^2 && \text{(Using the interpolation condition)} \\ \implies f_{ik}(w_k) - f_{ik}(w_{k+1}) &\leq \frac{1}{2\mu_{ik}} \|\nabla f_{ik}(w_k)\|^2 && \text{(Since, } f_{ik}(w^*) \leq f_{ik}(w_{k+1})\text{.)} \\ \implies c\cdot\eta_k &\leq \frac{\|\nabla f_{ik}(w_k)\|^2}{2\mu_{ik}\,\|\nabla f_{ik}(w_k)\|^2} && \text{(From the above relation on } \eta_k\text{.)} \\ \implies c\cdot\eta_k &\leq \frac{1}{2\mu_{ik}} \end{aligned}$$

Thus, the step-size returned by the line-search satisfies the relation $\eta_k \leq \min\{\frac{1}{2c\cdot\mu_{ik}}, \eta_{\max}\}$.

From the above relations,

$$\eta_k \in \left[\min\left\{\frac{2\,(1-c)}{L_{ik}}, \eta_{\max}\right\}, \min\left\{\frac{1}{2c\cdot\mu_{ik}}, \eta_{\max}\right\}\right]$$

□

B Proof for Theorem 1

Proof.

$$\|w_{k+1} - w^*\|^2 = \|w_k - \eta_k \nabla f_{ik}(w_k) - w^*\|^2$$
$$\|w_{k+1} - w^*\|^2 = \|w_k - w^*\|^2 - 2\eta_k \langle \nabla f_{ik}(w_k), w_k - w^* \rangle + \eta_k^2 \|\nabla f_{ik}(w_k)\|^2$$

Using strong-convexity of $f_{ik}(\cdot)$ (and setting $\mu_{ik} = 0$ if the f_{i_k} is not strongly-convex),

$$-\langle \nabla f_{ik}(w_k), w_k - w^* \rangle \leq f_{ik}(w^*) - f_{ik}(w_k) - \frac{\mu_{ik}}{2} \|w_k - w^*\|^2$$
$$\implies \|w_{k+1} - w^*\|^2 \leq \|w_k - w^*\|^2 + 2\eta_k \left[f_{ik}(w^*) - f_{ik}(w_k) - \frac{\mu_{ik}}{2} \|w_k - w^*\|^2 \right] + \eta_k^2 \|\nabla f_{ik}(w_k)\|^2$$
$$= \|w_k - w^*\|^2 + 2\eta_k \left[f_{ik}(w^*) - f_{ik}(w_k) \right] - \mu_{ik}\eta_k \|w_k - w^*\|^2 + \eta_k^2 \|\nabla f_{ik}(w_k)\|^2$$
$$\implies \|w_{k+1} - w^*\|^2 \leq (1 - \mu_{ik}\eta_k) \|w_k - w^*\|^2 + 2\eta_k \left[f_{ik}(w^*) - f_{ik}(w_k) \right] + \eta_k^2 \|\nabla f_{ik}(w_k)\|^2$$

Using Equation 1,

$$\eta_k^2 \|\nabla f_{ik}(w_k)\|^2 \leq \frac{\eta_k}{c} \left[f_{ik}(w_k) - f_{ik}(w_{k+1}) \right]$$
$$\implies \|w_{k+1} - w^*\|^2 \leq (1 - \mu_{ik}\eta_k) \|w_k - w^*\|^2 + 2\eta_k \left[f_{ik}(w^*) - f_{ik}(w_k) \right] + \frac{\eta_k}{c} \left[f_{ik}(w_k) - f_{ik}(w_{k+1}) \right]$$

The interpolation condition implies that w^* is the minimum for all functions f_i, implying that for all i, $f_i(w^*) \leq f_i(w_{k+1})$.

$$\|w_{k+1} - w^*\|^2 \leq (1 - \mu_{ik}\eta_k) \|w_k - w^*\|^2 + 2\eta_k \left[f_{ik}(w^*) - f_{ik}(w_k) \right] + \frac{\eta_k}{c} \left[f_{ik}(w_k) - f_{ik}(w^*) \right]$$
$$= (1 - \mu_{ik}\eta_k) \|w_k - w^*\|^2 + \left(2\eta_k - \frac{\eta_k}{c} \right) \left[f_{ik}(w^*) - f_{ik}(w_k) \right]$$

The term $[f_{ik}(w^*) - f_{ik}(w_k)]$ is negative. Let $c \geq \frac{1}{2} \implies \left(2\eta_k - \frac{\eta_k}{c}\right) \geq 0$ for all η_k.

$$\implies \|w_{k+1} - w^*\|^2 \leq (1 - \mu_{ik}\eta_k) \|w_k - w^*\|^2$$

Taking expectation wrt to i_k,

$$\implies \mathbb{E}\left[\|w_{k+1} - w^*\|^2 \right] \leq \mathbb{E}_{ik}\left[(1 - \mu_{ik}\eta_k) \|w_k - w^*\|^2 \right]$$
$$= (1 - \mathbb{E}_{ik}\left[\mu_{ik}\eta_k\right]) \|w_k - w^*\|^2$$
$$\leq \left(1 - \mathbb{E}_{ik}\left[\mu_{ik} \ \min\left\{ \frac{2(1-c)}{L_{ik}}, \eta_{\max} \right\} \right] \right) \|w_k - w^*\|^2 \qquad \text{(Using Equation 2)}$$

Setting $c = 1/2$,

$$\implies \mathbb{E}\left[\|w_{k+1} - w^*\|^2 \right] \leq \left(1 - \mathbb{E}_{ik}\left[\mu_{ik} \ \min\left\{ \frac{1}{L_{ik}}, \eta_{\max} \right\} \right] \right) \|w_k - w^*\|^2$$

We consider the following two cases: $\eta_{\max} < 1/L_{\max}$ and $\eta_{\max} \geq 1/L_{\max}$. When $\eta_{\max} < 1/L_{\max}$, we have $\eta_{\max} < 1/L_{ik}$ and,

$$\mathbb{E}\left[\|w_{k+1} - w^*\|^2 \right] \leq (1 - \mathbb{E}_{ik}\left[\mu_{ik} \ \eta_{\max}\right]) \|w_k - w^*\|^2$$
$$= (1 - \mathbb{E}_{ik}\left[\mu_{ik}\right] \ \eta_{\max}) \|w_k - w^*\|^2 = (1 - \bar{\mu} \ \eta_{\max}) \|w_k - w^*\|^2$$

By recursion through iterations $k = 1$ to T,

$$\mathbb{E}\left[\|w_T - w^*\|^2 \right] \leq (1 - \bar{\mu} \ \eta_{\max})^T \|w_0 - w^*\|^2 .$$

When $\eta_{\max} \geq 1/L_{\max}$, we use $\min\left\{\frac{1}{L_{ik}}, \eta_{\max}\right\} \geq \min\left\{\frac{1}{L_{\max}}, \eta_{\max}\right\}$ to obtain

$$\begin{aligned}
\mathbb{E}\left[\|w_{k+1} - w^*\|^2\right] &\leq \left(1 - \mathbb{E}_{ik}\left[\mu_{ik}\ \min\left\{\frac{1}{L_{\max}}, \eta_{\max}\right\}\right]\right)\|w_k - w^*\|^2 \\
&= \left(1 - \mathbb{E}_{ik}\left[\mu_{ik}\ \frac{1}{L_{\max}}\right]\right)\|w_k - w^*\|^2 \\
&= \left(1 - \frac{\mathbb{E}_{ik}\left[\mu_{ik}\right]}{L_{\max}}\right)\|w_k - w^*\|^2 = \left(1 - \frac{\bar{\mu}}{L_{\max}}\right)\|w_k - w^*\|^2
\end{aligned}$$

By recursion through iterations $k = 1$ to T,

$$\mathbb{E}\left[\|w_T - w^*\|^2\right] \leq \left(1 - \frac{\bar{\mu}}{L_{\max}}\right)^T \|w_0 - w^*\|^2 .$$

Putting the two cases together,

$$\mathbb{E}\left[\|w_T - w^*\|^2\right] \leq \max\left\{\left(1 - \frac{\bar{\mu}}{L_{\max}}\right), (1 - \bar{\mu}\ \eta_{\max})\right\}^T \|w_0 - w^*\|^2$$

□

C Proof for Theorem 2

Proof.

$$\begin{aligned}
\|w_{k+1} - w^*\|^2 &= \|w_k - \eta_k \nabla f_{ik}(w_k) - w^*\|^2 \\
\|w_{k+1} - w^*\|^2 &= \|w_k - w^*\|^2 - 2\eta_k \langle \nabla f_{ik}(w_k), w_k - w^* \rangle + \eta_k^2 \|\nabla f_{ik}(w_k)\|^2 \\
2\eta_k \langle \nabla f_{ik}(w_k), w_k - w^* \rangle &= \|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2 + \eta_k^2 \|\nabla f_{ik}(w_k)\|^2 \\
\langle \nabla f_{ik}(w_k), w_k - w^* \rangle &= \frac{1}{2\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right] + \frac{\eta_k}{2}\|\nabla f_{ik}(w_k)\|^2 \\
&\leq \frac{1}{2\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right] + \frac{f_{ik}(w_k) - f_{ik}(w_{k+1})}{2c} \qquad \text{(Using Equation 1)}
\end{aligned}$$

The interpolation condition implies that w^* is the minimum for all functions f_i, implying that for all i, $f_i(w^*) \leq f_i(w_{k+1})$.

$$\implies \langle \nabla f_{ik}(w_k), w_k - w^* \rangle \leq \frac{1}{2\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right] + \frac{f_{ik}(w_k) - f_{ik}(w^*)}{2c}$$

Taking expectation wrt i_k,

$$\begin{aligned}
\mathbb{E}\left[\langle \nabla f_{ik}(w_k), w_k - w^* \rangle\right] &\leq \mathbb{E}\left[\frac{1}{2\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right]\right] + \mathbb{E}\left[\frac{f_{ik}(w_k) - f_{ik}(w^*)}{2c}\right] \\
&= \mathbb{E}\left[\frac{1}{2\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right]\right] + \left[\frac{f(w_k) - f(w^*)}{2c}\right] \\
\implies \left\langle \mathbb{E}\left[\nabla f_{ik}(w_k)\right], w_k - w^* \right\rangle &\leq \mathbb{E}\left[\frac{1}{2\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right]\right] + \left[\frac{f(w_k) - f(w^*)}{2c}\right] \\
\implies \langle \nabla f(w_k), w_k - w^* \rangle &\leq \mathbb{E}\left[\frac{1}{2\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right]\right] + \left[\frac{f(w_k) - f(w^*)}{2c}\right]
\end{aligned}$$

By convexity,

$$\begin{aligned}
f(w_k) - f(w^*) &\leq \langle \nabla f(w_k), w_k - w^* \rangle \\
\implies f(w_k) - f(w^*) &\leq \mathbb{E}\left[\frac{1}{2\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right]\right] + \left[\frac{f(w_k) - f(w^*)}{2c}\right]
\end{aligned}$$

If $1 - \frac{1}{2c} \geq 0 \implies$ if $c \geq \frac{1}{2}$, then

$$\implies f(w_k) - f(w^*) \leq \mathbb{E}\left[\frac{c}{(2c-1)\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right]\right]$$

Taking expectation and summing from $k = 0$ to $k = T - 1$

$$\implies \mathbb{E}\left[\sum_{k=0}^{T-1}\left[f(w_k) - f(w^*)\right]\right] \leq \mathbb{E}\left[\sum_{k=0}^{T-1}\frac{c}{(2c-1)\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right]\right]$$

By Jensen's inequality,

$$\mathbb{E}\left[f(\bar{w}_T) - f(w^*)\right] \leq \mathbb{E}\left[\sum_{k=0}^{T-1}\left[\frac{f(w_k) - f(w^*)}{T}\right]\right]$$

$$\implies \mathbb{E}\left[f(\bar{w}_T) - f(w^*)\right] \leq \frac{1}{T}\mathbb{E}\left[\sum_{k=0}^{T-1}\frac{c}{(2c-1)\eta_k}\left[\|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right]\right]$$

If $\Delta_k = \|w_k - w^*\|^2$, then

$$\mathbb{E}\left[f(\bar{w}_T) - f(w^*)\right] \leq \frac{c}{T\ (2c-1)}\mathbb{E}\left[\sum_{k=0}^{T-1}\frac{1}{\eta_k}\left[\Delta_k - \Delta_{k+1}\right]\right]$$

Using Equation 2,

$$\frac{1}{\eta_k} \leq \max\left\{\frac{L_{ik}}{2\ (1-c)}, \frac{1}{\eta_{\max}}\right\} \leq \max\left\{\frac{L_{\max}}{2\ (1-c)}, \frac{1}{\eta_{\max}}\right\}$$

$$\implies \mathbb{E}\left[f(\bar{w}_T) - f(w^*)\right] \leq \frac{c \cdot \max\left\{\frac{L_{\max}}{2\ (1-c)}, \frac{1}{\eta_{\max}}\right\}}{(2c-1)\ T}\mathbb{E}\sum_{k=0}^{T-1}\left[\Delta_k - \Delta_{k+1}\right]$$

$$= \frac{c \cdot \max\left\{\frac{L_{\max}}{2\ (1-c)}, \frac{1}{\eta_{\max}}\right\}}{(2c-1)\ T}\mathbb{E}\left[\Delta_0 - \Delta_T\right]$$

$$\mathbb{E}\left[f(\bar{w}_T) - f(w^*)\right] \leq \frac{c \cdot \max\left\{\frac{L_{\max}}{2\ (1-c)}, \frac{1}{\eta_{\max}}\right\}}{(2c-1)\ T}\|w_0 - w^*\|^2$$

□

D Proof for Theorem 3

Proof.

By the smoothness assumption,

$$f(w_{k+1}) \leq f(w_k) - \langle \nabla f(w_k), \eta_k \nabla f_{ik}(w_k)\rangle + \frac{L\eta_k^2}{2}\|\nabla f_{ik}(w_k)\|^2$$

$$\frac{f(w_{k+1}) - f(w_k)}{\eta_k} \leq -\langle \nabla f(w_k), \nabla f_{ik}(w_k)\rangle + \frac{L\eta_k}{2}\|\nabla f_{ik}(w_k)\|^2$$

Taking expectation,

$$\begin{aligned}
\mathbb{E}\left[\frac{f(w_{k+1})-f(w_k)}{\eta_k}\right] &\leq -\left\|\nabla f(w_k)\right\|^2 + \mathbb{E}\left[\frac{L\eta_k}{2}\left\|\nabla f_{ik}(w_k)\right\|^2\right] \\
&\leq -\left\|\nabla f(w_k)\right\|^2 + \frac{L\eta_{\max}}{2}\mathbb{E}\left[\left\|\nabla f_{ik}(w_k)\right\|^2\right] \\
\implies \mathbb{E}\left[\frac{f(w_{k+1})-f(w_k)}{\eta_k}\right] &\leq -\left\|\nabla f(w_k)\right\|^2 + \frac{L\eta_{\max}\rho}{2}\left\|\nabla f(w_k)\right\|^2 \qquad \text{(By the SGC)} \\
\implies \left(1-\frac{L\eta_{\max}\rho}{2}\right)\left\|\nabla f(w_k)\right\|^2 &\leq \mathbb{E}\left[\frac{f(w_k)-f(w_{k+1})}{\eta_k}\right]
\end{aligned}$$

If $\eta_{\max} \leq \frac{2}{L\rho}$,

$$\left\|\nabla f(w_k)\right\|^2 \leq \frac{1}{1-\frac{L\eta_{\max}\rho}{2}}\mathbb{E}\left[\frac{f(w_k)-f(w_{k+1})}{\eta_k}\right]$$

If $f(w_k)-f(w_{k+1}) \leq 0$,

$$\implies \left\|\nabla f(w_k)\right\|^2 \leq 0 \implies \left\|\nabla f(w_k)\right\|^2 = 0$$

If $f(w_k)-f(w_{k+1}) \geq 0$,

$$\implies \left\|\nabla f(w_k)\right\|^2 \leq \frac{1}{1-\frac{L\eta_{\max}\rho}{2}}\,\mathbb{E}\left[\frac{1}{\eta_k}\left(f(w_k)-f(w_{k+1})\right)\right]$$

From the line-search we know that,

$$\begin{aligned}
\eta_k &\geq \min\left\{\eta_{\max}, \frac{2(1-c)}{L_{ik}}\right\} \\
\implies \frac{1}{\eta_k} &\leq \max\left\{\frac{1}{\eta_{\max}}, \frac{L_{ik}}{2(1-c)}\right\} \\
&\leq \frac{1}{\eta_{\max}} + \frac{L_{ik}}{2(1-c)} \\
\implies \frac{1}{\eta_k} &\leq \frac{1}{\eta_{\max}} + \frac{L_{max}}{2(1-c)}
\end{aligned}$$

From the above relations,

$$\begin{aligned}
\implies \left\|\nabla f(w_k)\right\|^2 &\leq \left(\frac{1}{1-\frac{L\eta_{\max}\rho}{2}}\right)\left(\frac{1}{\eta_{\max}}+\frac{L_{max}}{2(1-c)}\right)\mathbb{E}\left[f(w_k)-f(w_{k+1})\right] \\
\implies \min_{k\in[T]}\mathbb{E}\left\|\nabla f(w_k)\right\|^2 &\leq \frac{1}{T}\left(\frac{1}{1-\frac{L\eta_{\max}\rho}{2}}\right)\left(\frac{1}{\eta_{\max}}+\frac{L_{max}}{2(1-c)}\right)\mathbb{E}\left[f(w_0)-f(w_T)\right] \\
&\leq \frac{1}{T}\left(\frac{1}{1-\frac{L\eta_{\max}\rho}{2}}\right)\left(\frac{1}{\eta_{\max}}+\frac{L_{max}}{2(1-c)}\right)\left(f(w_0)-f(w^*)\right)
\end{aligned}$$

Let $c = 1/2$ and $\eta_{\max} = \frac{\gamma}{\rho L}$, for $\gamma < 2$,

$$\min_{k\in[T]}\mathbb{E}\left\|\nabla f(w_k)\right\|^2 \leq \frac{1}{T}\left(\frac{1}{1-\frac{\gamma}{2}}\right)\left(\frac{\rho L}{\gamma}+L_{max}\right)\left(f(w_0)-f(w^*)\right).$$

Noting $L \leq L_{max}$,

$$\min_{k\in[T]}\mathbb{E}\left\|\nabla f(w_k)\right\|^2 \leq \frac{2\,L_{max}}{T}\left(\frac{1}{2-\gamma}\right)\left(\frac{\rho}{\gamma}+1\right)\left(f(w_0)-f(w^*)\right)$$

Setting $\gamma = 3/2$,

$$\min_{k\in[T]} \mathbb{E}\left\|\nabla f(w_k)\right\|^2 \leq \frac{4\ L_{max}}{T}\ \left(\frac{2\rho}{3}+1\right)\ (f(w_0)-f(w^*))$$

□

E Proofs for SEG

E.1 Common lemmas

We denote $\|u-v\|^2$ as $\Delta(u,v) = \Delta(v,u)$. We first prove the following lemma that will be useful in the subsequent analysis.

Lemma 2. *For any set of vectors a, b, c, d, if $a = b + c$, then,*

$$\Delta(a,d) = \Delta(b,d) - \Delta(a,b) + 2\langle c, a-d\rangle$$

Proof.

$$\begin{aligned}\Delta(a,d) &= \|a-d\|^2 = \|b+c-d\|^2\\ &= \|b-d\|^2 + 2\langle c, b-d\rangle + \|c\|^2\end{aligned}$$

Since $c = a - b$,

$$\begin{aligned}\Delta(a,d) &= \|b-d\|^2 + 2\langle a-b, b-d\rangle + \|a-b\|^2\\ &= \|b-d\|^2 + 2\langle a-b, b-a+a-d\rangle + \|a-b\|^2\\ &= \|b-d\|^2 + 2\langle a-b, b-a\rangle + 2\langle a-b, a-d\rangle + \|a-b\|^2\\ &= \|b-d\|^2 - 2\|a-b\|^2 + 2\langle a-b, a-d\rangle + \|a-b\|^2\\ &= \|b-d\|^2 - \|a-b\|^2 + 2\langle c, a-d\rangle\\ \Delta(a,d) &= \Delta(b,d) - \Delta(a,b) + 2\langle c, a-d\rangle.\end{aligned}$$

□

E.2 Proof for Theorem 4

We start from Lemma 2 with $a = w_{k+1} = w_k - \eta_k \nabla f_{ik}(w'_k)$ and $d = w^*$:

$$\begin{aligned}\Delta(w_{k+1}, w^*) &= \Delta(w_k, w^*) - \Delta(w_{k+1}, w_k) - 2\eta_k\left[\langle \nabla f_{ik}(w'_k), w_{k+1} - w^*\rangle\right].\\ &= \Delta(w_k, w^*) - \eta_k^2\left\|\nabla f_{ik}(w'_k)\right\|^2 - 2\eta_k\left[\langle \nabla f_{ik}(w'_k), w_{k+1} - w^*\rangle\right].\end{aligned}$$

Using $w_{k+1} = w'_k + \eta_k \nabla f_{ik}(w_k) - \eta_k \nabla f_{ik}(w'_k)$ and completing the square,

$$\begin{aligned}\Delta(w_{k+1}, w^*) &= \Delta(w_k, w^*) - \eta_k^2\left\|\nabla f_{ik}(w'_k)\right\|^2 - 2\eta_k\left[\langle \nabla f_{ik}(w'_k), w'_k + \eta_k\nabla f_{ik}(w_k) - \eta_k\nabla f_{ik}(w'_k) - w^*\rangle\right]\\ &= \Delta(w_k, w^*) + \eta_k^2\left\|\nabla f_{ik}(w'_k)\right\|^2 - 2\eta_k\left[\langle \nabla f_{ik}(w'_k), w'_k + \eta_k\nabla f_{ik}(w_k) - w^*\rangle\right]\\ &= \Delta(w_k, w^*) + \eta_k^2\left\|\nabla f_{ik}(w'_k) - \nabla f_{ik}(w_k)\right\|^2 - \eta_k^2\left\|\nabla f_{ik}(w_k)\right\|^2 - 2\eta_k\left[\langle \nabla f_{ik}(w'_k), w'_k - w^*\rangle\right]\end{aligned}$$

Noting $\Delta(w'_k, w_k) = \eta_k^2\left\|\nabla f_{ik}(w_k)\right\|^2$ gives

$$\begin{aligned}&\Delta(w_{k+1}, w^*) = \Delta(w_k, w^*) - \Delta(w'_k, w_k) + \eta_k^2\left\|\nabla f_{ik}(w'_k) - \nabla f_{ik}(w_k)\right\|^2 - 2\eta_k\left[\langle \nabla f_{ik}(w'_k), w'_k - w^*\rangle\right]\\ \implies\ &2\eta_k\left[\langle \nabla f_{ik}(w'_k), w'_k - w^*\rangle\right] = \Delta(w_k, w^*) - \Delta(w'_k, w_k) + \eta_k^2\left\|\nabla f_{ik}(w'_k) - \nabla f_{ik}(w_k)\right\|^2 - \Delta(w_{k+1}, w^*). \qquad (5)\end{aligned}$$

By RSI, which states that for all w, $\langle \nabla f_i(w), w - w^*\rangle \geq \mu_i \left\|w^* - w\right\|^2$, we have

$$\langle \nabla f_{ik}(w'_k), w'_k - w^*\rangle \geq \mu_{ik}\Delta(w'_k, w^*)$$

By Young's inequality,

$$\begin{aligned}
&\Delta(w_k, w^*) \leq 2\Delta(w_k, w_k') + 2\Delta(w_k', w^*) \\
\implies &2\Delta(w_k', w^*) \geq \Delta(w_k, w^*) - 2\Delta(w_k, w_k') \\
\implies &\langle 2\eta_k \nabla f_{ik}(w_k'), w_k' - w^* \rangle \geq \mu_{ik}\eta_k \left[\Delta(w_k, w^*) - 2\Delta(w_k, w_k')\right]
\end{aligned}$$

Rearranging Equation (5),

$$\begin{aligned}
&\Delta(w_{k+1}, w^*) = \Delta(w_k, w^*) - \Delta(w_k', w_k) + \eta_k^2 \left\|\nabla f_{ik}(w_k') - \nabla f_{ik}(w_k)\right\|^2 - 2\eta_k \left[\langle \nabla f_{ik}(w_k'), w_k' - w^* \rangle\right] \\
\implies &\Delta(w_{k+1}, w^*) \leq \Delta(w_k, w^*) - \Delta(w_k', w_k) + \eta_k^2 \left\|\nabla f_{ik}(w_k') - \nabla f_{ik}(w_k)\right\|^2 - \mu_{ik}\eta_k \left[\Delta(w_k, w^*) - 2\Delta(w_k, w_k')\right] \\
&\Delta(w_{k+1}, w^*) \leq (1 - \eta_k\mu_{ik})\,\Delta(w_k, w^*) - \Delta(w_k', w_k) + \eta_k^2 \left\|\nabla f_{ik}(w_k') - \nabla f_{ik}(w_k)\right\|^2 + 2\mu_{ik}\eta_k\Delta(w_k, w_k')
\end{aligned}$$

Now we consider using a constant step-size as well as the Lipschitz line-search.

E.2.1 Using a constant step-size

Proof.

Using smoothness of $f_{ik}(\cdot)$,

$$\begin{aligned}
&\Delta(w_{k+1}, w^*) \leq (1 - \eta_k\mu_{ik})\,\Delta(w_k, w^*) - \Delta(w_k', w_k) + \eta_k^2 L_{ik}^2 \Delta(w_k', w_k) + 2\mu_{ik}\eta_k\Delta(w_k, w_k') \\
\implies &\Delta(w_{k+1}, w^*) \leq (1 - \eta_k\mu_{ik})\,\Delta(w_k, w^*) + \left(\eta_k^2 L_{ik}^2 - 1 + 2\mu_{ik}\eta_k\right)\Delta(w_k', w_k)
\end{aligned}$$

Taking expectation with respect to i_k,

$$\mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \mathbb{E}\left[(1 - \eta_k\mu_{ik})\,\Delta(w_k, w^*)\right] + \mathbb{E}\left[\left(\eta_k^2 L_{ik}^2 - 1 + 2\mu_{ik}\eta_k\right)\Delta(w_k', w_k)\right]$$

Note that w_k doesn't depend on i_k. Furthermore, neither does w^* because of the interpolation property.

$$\implies \mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \mathbb{E}\left[1 - \eta_k\mu_{ik}\right]\Delta(w_k, w^*) + \mathbb{E}\left[\left(\eta_k^2 L_{ik}^2 - 1 + 2\mu_{ik}\eta_k\right)\Delta(w_k', w_k)\right]$$

If $\eta_k \leq \frac{1}{4 \cdot L_{\max}}$, then $\left(\eta_k^2 L_{ik}^2 - 1 + 2\mu_{ik}\eta_k\right) \leq 0$ and

$$\begin{aligned}
&\implies \mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \mathbb{E}\left[1 - \frac{\mu_{ik}}{4L_{\max}}\right]\Delta(w_k, w^*) \\
&\implies \mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \left(1 - \frac{\bar{\mu}}{4L_{\max}}\right)\Delta(w_k, w^*) \\
&\implies \mathbb{E}\left[\Delta(w_k, w^*)\right] \leq \left(1 - \frac{\bar{\mu}}{4L_{\max}}\right)^T \Delta(w_0, w^*)
\end{aligned}$$

□

E.2.2 Using the line-search

Proof.

Using Equation (4) to control the difference in gradients,

$$\begin{aligned}
&\Delta(w_{k+1}, w^*) \leq (1 - \eta_k\mu_{ik})\,\Delta(w_k, w^*) - \Delta(w_k', w_k) + c^2\Delta(w_k', w_k) + 2\mu_{ik}\eta_k\Delta(w_k, w_k') \\
\implies &\Delta(w_{k+1}, w^*) \leq (1 - \eta_k\mu_{ik})\,\Delta(w_k, w^*) + \left(c^2 + 2\mu_{ik}\eta_k - 1\right)\Delta(w_k', w_k)
\end{aligned}$$

Taking expectation with respect to i_k,

$$\mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \mathbb{E}\left[1 - \eta_k\mu_{ik}\Delta(w_k, w^*)\right] + \mathbb{E}\left[\left(c^2 - 1 + 2\eta_k\mu_i\right)\Delta(w_k', w_k)\right]$$

Note that w_k doesn't depend on i_k. Furthermore, neither does w^* because of the interpolation property.

$$\implies \mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \mathbb{E}\left[1 - \eta_k\mu_{ik}\right]\Delta(w_k, w^*) + \mathbb{E}\left[\left(c^2 - 1 + 2\eta_k\mu_{ik}\right)\Delta(w_k', w_k)\right]$$

Using smoothness, the line-search in Equation 4 is satisfied if $\eta_k \leq \frac{c}{L_{ik}}$, implying that the step-size returned by the line-search always satisfies $\eta_k \geq \min\left\{\frac{c}{L_{ik}}, \eta_{\max}\right\}$.

$$\implies \mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \mathbb{E}\left(1 - \mu_{ik} \min\left\{\frac{c}{L_{ik}}, \eta_{\max}\right\}\right) \Delta(w_k, w^*) + \mathbb{E}\left[\left(c^2 - 1 + 2\eta_k \mu_{ik}\right) \Delta(w'_k, w_k)\right]$$

If we ensure that $\eta_k \leq \frac{c}{\mu_{ik}}$, then $c^2 - 1 + 2\eta_k\mu_{ik} \leq 0$. In other words, we need to ensure that $\eta_{\max} \leq \min_i \frac{c}{\mu_i}$. Choosing $c = 1/4$, we obtain the following:

$$\mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \mathbb{E}\left(1 - \mu_{ik} \min\left\{\frac{1}{4\ L_{ik}}, \eta_{\max}\right\}\right) \Delta(w_k, w^*)$$

We consider the following cases: $\eta_{\max} < \frac{1}{4\ L_{\max}}$ and $\eta_{\max} \geq \frac{1}{4\ L_{\max}}$. When $\eta_{\max} < \frac{1}{4\ L_{\max}}$,

$$\begin{aligned}
\mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] &\leq \mathbb{E}\left(1 - \mu_{ik}\ \eta_{\max}\right) \Delta(w_k, w^*) \\
&= \left(1 - \bar{\mu}\ \eta_{\max}\right) \Delta(w_k, w^*) \\
\implies \mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] &\leq \left(1 - \bar{\mu}\ \eta_{\max}\right)^T \Delta(w_0, w^*)
\end{aligned}$$

When $\eta_{\max} \geq 1/(4\ L_{\max})$, we use $\min\left\{\frac{1}{4\ L_{ik}}, \eta_{\max}\right\} \geq \min\left\{\frac{1}{4\ L_{\max}}, \eta_{\max}\right\}$ to obtain

$$\begin{aligned}
\mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] &\leq \mathbb{E}\left(1 - \mu_{ik} \min\left\{\frac{1}{4\ L_{\max}}, \eta_{\max}\right\}\right) \Delta(w_k, w^*) \\
&= \mathbb{E}\left(1 - \mu_{ik} \frac{1}{4\ L_{\max}}\right) \Delta(w_k, w^*) \\
&= \left(1 - \frac{\bar{\mu}}{4\ L_{\max}}\right) \Delta(w_k, w^*) \\
\implies \mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] &\leq \left(1 - \frac{\bar{\mu}}{4\ L_{\max}}\right)^T \Delta(w_0, w^*).
\end{aligned}$$

Putting the two cases together, we obtain

$$\mathbb{E}\left[\Delta(w_{k+1}, w^*)\right] \leq \max\left\{\left(1 - \frac{\bar{\mu}}{4\ L_{\max}}\right), \left(1 - \bar{\mu}\ \eta_{\max}\right)\right\}^T \Delta(w_0, w^*).$$

□

E.3 Proof of SEG for convex minimization

Theorem 5. *Assuming the interpolation property and under L-smoothness and convexity of f, SEG with Lipschitz line-search with $c = 1/\sqrt{2}$ in Equation 4 and iterate averaging achieves the following rate:*

$$\mathbb{E}\left[f(\bar{w}_T) - f(w^*)\right] \leq \frac{2\ \max\left\{\sqrt{2}\ L_{max}, \frac{1}{\eta_{max}}\right\}}{T} \left\|w_0 - w^*\right\|^2.$$

Here, $\bar{w}_T = \frac{\left[\sum_{i=1}^T w_i\right]}{T}$ is the averaged iterate after T iterations.

Proof. Starting from Equation (5),

$$2\eta_k \left[\langle \nabla f_{ik}(w'_k), w'_k - w^* \rangle\right] = \Delta(w_k, w^*) - \Delta(w'_k, w_k) + \eta_k^2 \left\|\nabla f_{ik}(w'_k) - \nabla f_{ik}(w_k)\right\|^2 - \Delta(w_{k+1}, w^*)$$

and using the standard convexity inequality,

$$\begin{aligned}\langle \nabla f_{ik}(w'_k), w'_k - w^* \rangle &\geq f_{i_k}(w'_k) - f_{i_k}(w^*) \\ &\geq \tfrac{1}{4}(f_{i_k}(w'_k) - f_{i_k}(w^*)) \\ &\geq \tfrac{1}{4}(f_{i_k}(w_k) - \eta_k \|\nabla f_{ik}(w_k)\|^2 - f_{i_k}(w^*)) \\ &= \tfrac{1}{4}(f_{i_k}(w_k) - \frac{1}{\eta_k}\Delta(w_k, w'_k) - f_{i_k}(w^*)) \\ \implies 2\eta_k \left[\langle \nabla f_{ik}(w'_k), w'_k - w^* \rangle\right] &\geq \frac{\eta_k}{2}\left[f_{i_k}(w_k) - f_{i_k}(w^*)\right] - \frac{1}{2}\Delta(w_k, w'_k)\end{aligned}$$

where we used the interpolation hypothesis to say that w^* is a minimizer of f_{i_k} and thus $f_{i_k}(w'_k) \geq f_{i_k}(w^*)$. Combining this with (5) and (4) leads to,

$$\begin{aligned}\frac{\eta_k}{2}(f_{i_k}(w_k) - f_{i_k}(w^*)) &\leq \Delta(w_k, w^*) - \Delta(w_{k+1}, w^*) - \tfrac{1}{2}\Delta(w'_k, w_k) + \eta_k^2 \left\|\nabla f_{ik}(w'_k) - \nabla f_{ik}(w_k)\right\|^2 \\ &\leq \Delta(w_k, w^*) - \Delta(w_{k+1}, w^*) - (\tfrac{1}{2} - c^2)\Delta(w'_k, w_k) \\ &\leq \Delta(w_k, w^*) - \Delta(w_{k+1}, w^*), \\ \implies f_{i_k}(w_k) - f_{i_k}(w^*) &\leq \frac{2}{\eta_k}\left[\Delta(w_k, w^*) - \Delta(w_{k+1}, w^*)\right]\end{aligned}$$

where for the last inequality we used Equation 4 and the fact that $c^2 \leq 1/2$. By definition of the Lipschitz line-search, $\eta_k \in [\min\{c/L_{\max}, \eta_{\max}\}, \eta_{\max}]$, implying

$$\frac{1}{\eta_k} \leq \max\left\{\frac{L_{\max}}{c}, \frac{1}{\eta_{\max}}\right\}$$

Setting $c = \frac{1}{\sqrt{2}}$,

$$\begin{aligned}\frac{1}{\eta_k} &\leq \max\left\{\sqrt{2}L_{\max}, \frac{1}{\eta_{\max}}\right\} \\ f_{i_k}(w_k) - f_{i_k}(w^*) &\leq 2\ \max\left\{\sqrt{2}L_{\max}, \frac{1}{\eta_{\max}}\right\}(\Delta(w_k, w^*) - \Delta(w_{k+1}, w^*))\end{aligned}$$

Taking expectation with respect to i_k,

$$f(w_k) - f(w^*) \leq 2\ \max\left\{\sqrt{2}L_{\max}, \frac{1}{\eta_{\max}}\right\}(\Delta(w_k, w^*) - \mathbb{E}\Delta(w_{k+1}, w^*))$$

Finally, taking the expectation respect to w_k and summing for $k = 1, \ldots, T$, we get,

$$\mathbb{E}\left[f(\bar{w}_k) - f(w^*)\right] \leq \frac{2\ \max\left\{\sqrt{2}L_{\max}, \frac{1}{\eta_{\max}}\right\}\ \Delta(w_0, w^*)}{T}$$

□

E.4 SEG for general strongly monotone operators

Let $F(\cdot)$ be a Lipschitz (strongly)-monotone operator. F satisfies the following inequalities for all u, v,

$$\|F(u) - F(v)\| \leq L\|u - v\| \qquad \text{(Lipschitz continuity)}$$

$$\langle F(u) - F(v), u - v\rangle \geq \mu\|u - v\|^2 \qquad \text{(Strong monotonicity)}$$

Here, μ is the strong-monotonicity constant and L is the Lipschitz constant. Note that $\mu = 0$ for monotone operators. We seek the solution w^* to the following optimization problem: $\sup_w \langle F(w^*), w^* - w\rangle \leq 0$.

Note that for strongly-convex minimization where $w^* = \arg\min f(w)$, F is equal to the gradient operator and μ and L are the strong-convexity and smoothness constants in the previous sections.

SEG [36] is a common method for optimizing stochastic variational inequalities and results in an $O(1/\sqrt{T})$ rate for monotone operators and an $O(1/T)$ rate for strongly-monotone operators [24]. For strongly-monotone operators, the convergence can be improved to obtain a linear rate by using variance-reduction methods [15, 61] exploiting the finite-sum structure in F. In this setting,

$F(w) = \frac{1}{n}\sum_{i=1}^{n} F_i(w)$. To the best of our knowledge, the interpolation condition has not been studied in the context of general strongly monontone operators. In this case, the interpolation condition implies that $F_i(w^*) = 0$ for all operators F_i in the finite sum.

Theorem 6 (Strongly-monotone)**.** *Assuming (a) interpolation, (b) L-smoothness and (c) μ-strong monotonocity of F, SEG using Lipschitz line-search with $c = 1/4$ in Equation 4 and setting $\eta_{max} \leq \min_i \frac{1}{4\mu_i}$ has the rate:*

$$\mathbb{E}\left[\|w_k - w^*\|^2\right] \leq \left(\max\left\{\left(1 - \frac{\bar{\mu}}{4\, L_{max}}\right), (1 - \eta_{max}\, \bar{\mu})\right\}\right)^T \|w_0 - w^*\|^2 .$$

Proof.

For each $F_{ik}(\cdot)$, we use the strong-monotonicity condition with constant μ_{ik},

$$\langle F_{ik}(u) - F_{ik}(v), u - v\rangle \geq \mu_{ik} \|u - v\|^2$$

Set $u = w$, $v = w^*$,

$$\implies \langle F_{ik}(w) - F_{ik}(w^*), w - w^*\rangle \geq \mu_{ik} \|w - w^*\|^2$$

By the interpolation condition,

$$F_{ik}(w^*) = 0$$
$$\implies \langle F_{ik}(w), w - w^*\rangle \geq \mu_{ik} \|w - w^*\|^2$$

This is equivalent to an RSI-like condition, but with the gradient operator $\nabla f_{ik}(\cdot)$ replaced with a general operator $F_{ik}(()\cdot)$.

From here on, the theorem follows the same proof as that for Theorem 4 above with the $F_{ik}()$ instead of $\nabla f_{ik}()$ and the strong-convexity constant being replaced with the constant for strong-monotonicity. □

Like in the RSI case, the above result can also be obtained using a constant step-size $\eta \leq \frac{1}{4\, L_{\max}}$.

E.5 SEG for bilinear saddle point problems

Let us consider the bilinear saddle-point problem of the form $\min_x \max_y x^\top A y - x^\top b - y^\top c$, where A is the "coupling" matrix and where both b and c are vectors [15, 24]. In this case, the (monotone) operator $F(x, y) = \left[Ax - b,\ -A^\top y + c\right]$ and we assume the finite sum formulation as:

$$x^\top A y - x^\top b - y^\top c = \frac{1}{n}\sum_{i=0}^{n} x^\top A_i y - x^\top b_i - y^\top c_i \tag{6}$$

We show that the interpolation condition enables SEG with Lipschitz line-search achieve a linear rate of convergence. In every iteration, the SEG algorithm samples rows A_i (resp. columns A_j) of the matrix A and the respective coefficient b_i (resp. c_j). If x_k and y_k correspond to the iterates for the minimization and maximization problem respectively, then the update rules for SEG can be written as:

$$\begin{cases} x_{k+1} = x_k - \eta_k(A_{i_k} y_{k+1/2} - b_{i_k}) \\ y_{k+1} = y_k + \eta_k(A_{i_k}^\top x_{k+1/2} - c_{i_k}) \end{cases} \quad \text{and} \quad \begin{cases} x_{k+1/2} = x_k - \eta_k(A_{i_k} y_k - b_{i_k}) \\ y_{k+1/2} = y_k + \eta_k(A_{i_k}^\top x_k - c_{i_k}) \end{cases} \tag{7}$$

which can be more compactly written as,

$$\begin{aligned} x_{k+1} &= x_k - \eta_k(A_{i_k}(y_k + \eta_k(A_{i_k}^\top x_k - c_{i_k}) - b_{i_k}) \\ y_{k+1} &= y_k + \eta_k(A_{i_k}^\top(x_k - \eta_k(A_{i_k} y_k - b_{i_k}) - c_{i_k}) . \end{aligned} \tag{8}$$

We now prove that SEG attains the following linear rate of convergence.

Theorem 7 (Bilinear)**.** *Assuming the (a) interpolation property and for the (b) bilinear saddle point problem, SEG with Lipschitz line-search with $c = 1/\sqrt{2}$ in Equation 4 achieves the following rate:*

$$\mathbb{E}\left[\|w_k - w^*\|^2\right] \leq \left(\max\left\{\left(1 - \frac{\sigma_{\min}(\mathbb{E}[A_{i_k} A_{i_k}^\top]}{4 \max_i \sigma_{max}(A_i A_i^\top)}\right), \left(1 - \frac{\eta_{max}}{2}\, \sigma_{\min}(\mathbb{E}[A_{i_k} A_{i_k}^\top]\right)\right\}\right)^T (\|x_k\|^2 + \|y_k\|^2)$$

Proof. If (x^*, y^*) is the solution to the above saddle point problem, then interpolation hypothesis implies that

$$A_{i_k} y^* = b_{i_k} \quad \text{and} \quad A_i^\top x^* = c_i$$

We note that the problem can be reduced to the case $b = c = 0$ by using the change of variable $\tilde{x}_k := x_k - x^*$ and $\tilde{y}_k := y_k - y^*$.

$$\tilde{x}_{k+1} = x_{k+1} - x^* = x_k - x^* - \eta_k(A_{i_k}(y_k - y^* + \eta_k A_{i_k}^\top(x_k - x^*)) = \tilde{x}_k - \eta_k A_{i_k}(\tilde{y}_k + \eta_k A_{i_k}^\top \tilde{x}_k)$$
$$\tilde{y}_{k+1} = y_{k+1} - y^* = y_k - y^* + \eta_k(A_{i_k}^\top(x_k - x^* - \eta_k A_{i_k}(y_k - y^*)) = \tilde{y}_k + \eta_k A_{i_k}^\top(\tilde{x}_k - \eta_k A_{i_k}\tilde{y}_k)$$

Thus, $(\tilde{x}_{k+1}, \tilde{y}_{k+1})$ correspond to the update rule Eq.(8) with $b = c = 0$. Note that the interpolation hypothesis is key for this problem reduction.

In the following, without loss of generality, we will assume that $b = c = 0$.

Using the update rule, we get,

$$\|x_{k+1}\|^2 + \|y_{k+1}\|^2 = \|x_k\|^2 + \|y_k\|^2 - \eta_k^2(x_k^\top A_{i_k} A_{i_k}^\top x_k + y_k^\top A_{i_k}^\top A_{i_k} y_k) + \eta_k^4(x_k^\top (A_{i_k} A_{i_k}^\top)^2 x_k + y_k^\top (A_{i_k}^\top A_{i_k})^2 y_k)$$

The line-search hypothesis can be simplified as,

$$\eta_k^2(x_k^\top (A_{i_k} A_{i_k}^\top)^2 x_k + y_k^\top (A_{i_k}^\top A_{i_k})^2 y_k) \leq c^2(x_k^\top A_{i_k} A_{i_k}^\top x_k + y_k^\top A_{i_k}^\top A_{i_k} y_k) \tag{9}$$

leading to,

$$\|x_{k+1}\|^2 + \|y_{k+1}\|^2 \leq \|x_k\|^2 + \|y_k\|^2 - \eta_k^2(1 - c^2)(x_k^\top A_{i_k} A_{i_k}^\top x_k + y_k^\top A_{i_k}^\top A_{i_k} y_k)$$

Noting that $L_{\max} = \left[\max_i \sigma_{\max}(A_i A_i^\top)\right]^{1/2}$, we obtain $\eta_k \geq \min\left\{\left[2\max_i \sigma_{\max}(A_i A_i^\top)\right]^{-1/2}, \eta_{\max}\right\}$ from the Lipschitz line-search. Taking the expectation with respect to i_k gives,

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1}\|^2 + \|y_{k+1}\|^2\right] &\leq (1 - \eta_k^2 \sigma_{\min}(\mathbb{E}[A_{i_k} A_{i_k}^\top])(1 - c^2))(\|x_k\|^2 + \|y_k\|^2) \\ &\leq \max\left\{\left(1 - \frac{\sigma_{\min}(\mathbb{E}[A_{i_k} A_{i_k}^\top])}{4\max_i \sigma_{\max}(A_i A_i^\top)}\right), \left(1 - \frac{\eta_{\max}}{2}\sigma_{\min}(\mathbb{E}[A_{i_k} A_{i_k}^\top])\right)\right\}(\|x_k\|^2 + \|y_k\|^2).\end{aligned}$$

Applying this inequality recursively and taking expectations yields the final result. □

Observe that the rate depends on the minimum and maximum singular values of the matrix formed using the mini-batch of examples selected in the SEG iterations. Note that these are the first results for bilinear min-max problems in the stochastic, interpolation setting.

F Additional Experimental Details

In this section we give details for all experiments in the main paper and the additional results given in Appendix G. In all experiments, we used the default learning rates provided in the implementation for the methods we compare against. For the proposed line-search methods and for *all* experiments in this paper, we set the initial step-size $\eta_{\max} = 1$ and use back-tracking line-search where we reduce the step-size by a factor of 0.9 if the line-search is not satisfied. We used $c = 0.1$ for all our experiments with both Armijo and Goldstein line-search procedures, $c = 0.9$ for SEG with Lipschitz line-search, and $c = 0.5$ when using Nesterov acceleration [3]. For Polyak acceleration, we use $c = 0.1$ in our experiments with deep neural networks and $c = 0.5$ otherwise. For our non-convex experiments, we always constrain the step-size to be less than 10 to prevent it from becoming unbounded. Note that we conduct a robustness study to quantify the influence of the c and $\eta_{\max}$ parameter in Section G.1. For the heuristic in [71, 72], we set the step-size increase factor to $\gamma = 1.5$ for convex minimization and use $\gamma = 2$ for non-convex minimization. Similarly, when using Polyak momentum we set the momentum factor to the highest value that does not lead to divergence. It is set to $\beta = 0.8$ in the convex case and $\beta = 0.6$ in the non-convex case [4].

F.1 Synthetic Matrix Factorization Experiment

In the following we give additional details for synthetic matrix factorization experiment in Section 7.2. As stated in the main text, we set $A \in \mathbb{R}^{10^3 \times 6}$ with condition number $\kappa(A) = 10^{10}$ and generated a fixed dataset of 1000 samples using the code released by Ben Recht [5]. We withheld 200 of these examples as a test set. All optimizers used mini-batches of 100 examples and were run for 50 epochs. We averaged over 20 runs with different random seeds to control for variance in the training loss, which approached machine precision for several optimizers.

F.2 Binary Classification using Kernel Methods

We give additional details for the experiments on binary classification with RBF kernels in Section 7.3. For all datasets, we used only the training sets available in the `LIBSVM` [14] library and used an 80:20 split of it. The 80 percent split of the data was used

[3] Note that these choices are inspired by the theory

[4] We hope to use method such as [89] to automatically set the momentum parameter in the future.

[5] This code is available at `https://github.com/benjamin-recht/shallow-linear-net`

Dataset	Dimension (d)	Training Set Size	Test Set Size	Kernel Bandwidth	SVRG Step-Size
mushrooms	112	6499	1625	0.5	500
ijcnn	22	39992	9998	0.05	500
rcv1	47236	16194	4048	0.25	500
w8a	300	39799	9950	20.0	0.0025

Table 1: Additional details for binary classification datasets used in convex minimization experiments. Kernel bandwidths were selected by 10-fold cross validation on the training set. SVRG step-sizes were selected by 3-fold CV on the training set. See text for more details.

as a training set and 20 percent split as the test set. The bandwidth parameters for the RBF kernel were selected by grid search using 10-fold cross-validation on the training splits. The grid of kernel bandwidth parameters that were considered is `[0.05, 0.1, 0.25, 0.5, 1, 2.5, 5, 10, 15, 20]`. For the cross-validation, we used batch L-BFGS to minimize both objectives on the rcv1 and mushrooms datasets, while we used the Coin-Betting algorithm on the larger w8a and ijcnn datasets with mini-batches of 100 examples. In both cases, we ran the optimizers for 100 epochs on each fold. The bandwidth parameters that maximized cross-validated accuracy were selected for our final experiments. The final kernel parameters are given in Table 1, along with additional details for each dataset.

We used the default hyper-parameters for all baseline optimizers used in our other experiments. For PLS, we used the exponential exploration strategy and its default hyper-parameters. Fixed step-size SVRG requires that the step-size parameter to be well-tuned in order to obtain a fair comparison with adaptive methods. To do so, we selected step-sizes by grid search. For each step-size, a 3-fold cross-validation experiment was run on each dataset's training set. On each fold, SVRG was run with mini-batches of size 100 for 50 epochs. Final step-sizes were selected by maximizing convergence rate on the cross-validated loss. The grid of possible step-sizes was expanded whenever the best step-size found was the largest or smallest step-size in the considered grid. We found that the mushrooms, ijcnn, and rcv1 datasets admitted very large step-sizes; in this case, we terminated our grid-search when increasing the step-size further gave only marginal improvement. The final step-sizes selected by this procedure are given in Table 1.

Each optimizer was run with five different random seeds in the final experiment. All optimizers used mini-batches of 100 examples and were run for 35 epochs. Experiment figures display shaded error bars of one standard-deviation from the mean. Note that we did not use a bias parameter in these experiments.

F.3 Multi-class Classification using Deep Networks

For mutliclass-classification with deep networks, we considered the MNIST and CIFAR10 datasets, each with 10 classes. For MNIST, we used the standard training set consisting of 60k examples and a test set of 10k examples; whereas for CIFAR10, this split was 50k training examples and 10k examples in the test set. As in the kernel experiments, we evaluated the optimizers using the softmax. All optimizers were used with their default learning rates and without any weight decay. We used the experimental setup proposed in [51] and used a batch-size of 128 for all methods and datasets. As before, each optimizer was run with five different random seeds in the final experiment. The optimizers were run until the performance of most methods saturated; 100 epochs for MNIST and 200 epochs for the models on the CIFAR10 dataset. We compare against a tuned SGD method, that uses a constant step-size selected according to a search on the $[1e-1, 1e-5]$ grid and picking the variant that led to the best convergence in the training loss. This procedure resulted in choosing a step-size of 0.01 for the MLP on MNIST and 0.1 for both models on CIFAR10.

G Additional Results

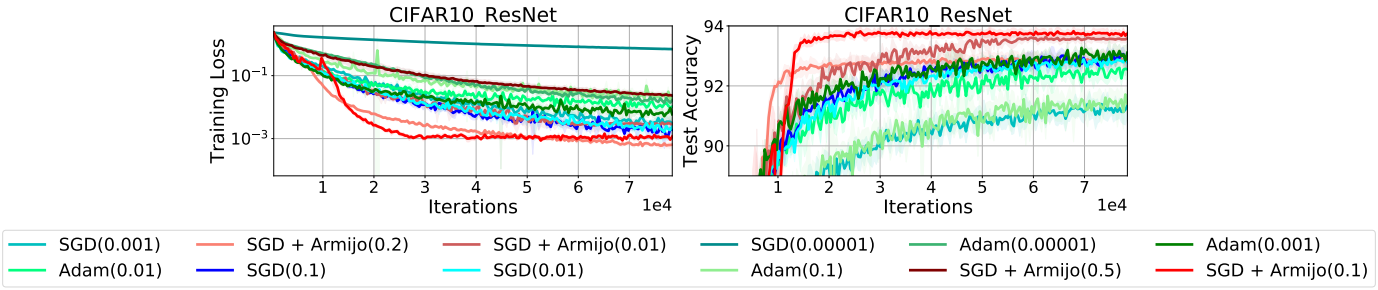

Figure 6: Testing the robustness of Adam, SGD and SGD with Armijo line-search for training ResNet on CIFAR10. SGD is highly sensitive to it's fixed step-size; selecting too small a step-size results in very slow convergence. In contrast, SGD + Armijo has similar performance with $c = 0.1$ and $c = 0.01$ and all c values obtain reasonable performance. We note that Adam is similarly robust to its initial learning-rate parameter.

G.1 Evaluating robustness and computation

In this experiment, we compare the robustness and computational complexity between the three best performing methods across datasets: Adam, constant step-size and SGD with Armijo line-search. For both Adam and constant step-size SGD, we vary the step-size in the $[10^{-1}, 10^{-5}]$ range; whereas for the SGD with line-search, we vary the parameter c in the $[0.1, 0.5]$ range and vary $\eta_{max} \in [1, 10^3]$ range. We observe that although the performance of constant step-size SGD is sensitive to its step-size; SGD with

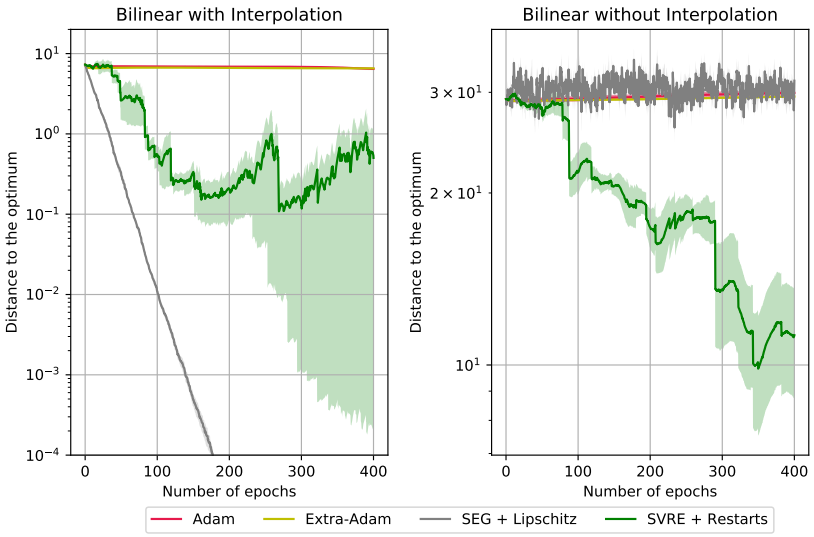

Figure 7: Min-max optimization on synthetic bilinear example (left) with interpolation (right) without interpolation. SEG with Lipschitz line-search converges linearly when interpolation is satisfied – in agreement with in Theorem 7 – although it fails to converge when interpolation is violated.

Armijo line-search is robust with respect to the c parameter. Similarly, we find that Adam is quite robust with respect to its initial learning rate.

G.2 Min-max optimization for bilinear games

Chavdarova et al. [15] propose a challenging stochastic bilinear game as follows:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\max_{\boldsymbol{\varphi}\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{\theta}^\top\boldsymbol{b}_i+\boldsymbol{\theta}^\top\boldsymbol{A}_i\boldsymbol{\varphi}+\boldsymbol{c}_i^\top\boldsymbol{\varphi}\right),\quad [\boldsymbol{A}_i]_{kl}=\delta_{kli}\,,\ [\boldsymbol{b}_i]_k\,,[\boldsymbol{c}_i]_k\sim\mathcal{N}(0,\tfrac{1}{d}),1\leq k,l\leq d$$

Standard methods such as stochastic extragradient fail to converge on this example. We compare Adam, ExtraAdam [24], SEG with backtracking line-search using Equation 4 with $c=1/\sqrt{2}$ and p-SVRE [15]. The latter combines restart, extrapolation and variance reduction for finite sum. It exhibits linear convergence rate but requires the tuning of the restart parameter p and do not have any convergence guarantees on such bilinear problem. ExtraAdam [24] combines extrapolation and Adam has good performances on GANs although it fails to converge on this simple stochastic bilinear example.

In our synthetic experiment, we consider two variants of this bilinear game; one where interpolation condition is satisfied, and the other when it is not. As predicted by the theory, SEG + Lipschitz results in linear convergence where interpolation is satisfied and does not converge to the solution when it is not. When interpolation is satisfied, empirical convergence rate is faster than SVRE, the best variance reduced method. Note that SVRE does well even in the absence of interpolation, and the both variants of Adam fail to converge on either example.

G.3 Synthetic Experiment and Binary Classification with Kernels

We provide additional results for binary classification with RBF kernels on the rcv1 and w8a datasets. As before, we do not use regularization. We compare against L4 Mom [68] as well as the original baselines introduced in Section 7.1. For fairness, we reproduce the results for mushrooms and ijcnn with L4 Mom included. Figure 8 shows the training loss and test accuracy for the methods considered, while Figure 10 shows the evolution of step-sizes for SGD+Armijo on all four kernel datasets.

The proposed line-search methods perform well on both rcv1 and w8a although neither dataset satisfies the interpolation condition with the selected kernel bandwidths. Furthermore, all of the proposed line-search methods converge quickly and remain at the global minimum for the w8a dataset, which is particularly ill-conditioned. In contrast, adaptive optimizers, such as Adam, fail to converge. Unlike other methods, PLS uses a separate mini-batch for each step of the line-search procedure. Accordingly, we plot the number of iterations *accepted* by the probabilistic Wolfe conditions, which may correspond to several mini-batches of information. Despite this, PLS converges slowly. In practice, we observed that the initial step-size was accepted at most iterations of the PLS line-search.

Figure 10 provides an additional comparison against L4 Mom on the synthetic matrix factorization problem from Section 7.2. We observe that L4 Mom is unstable when used for *stochastic* optimization of this problem, especially when interpolation is not satisfied. The method converges slowly when interpolation is satisfied.

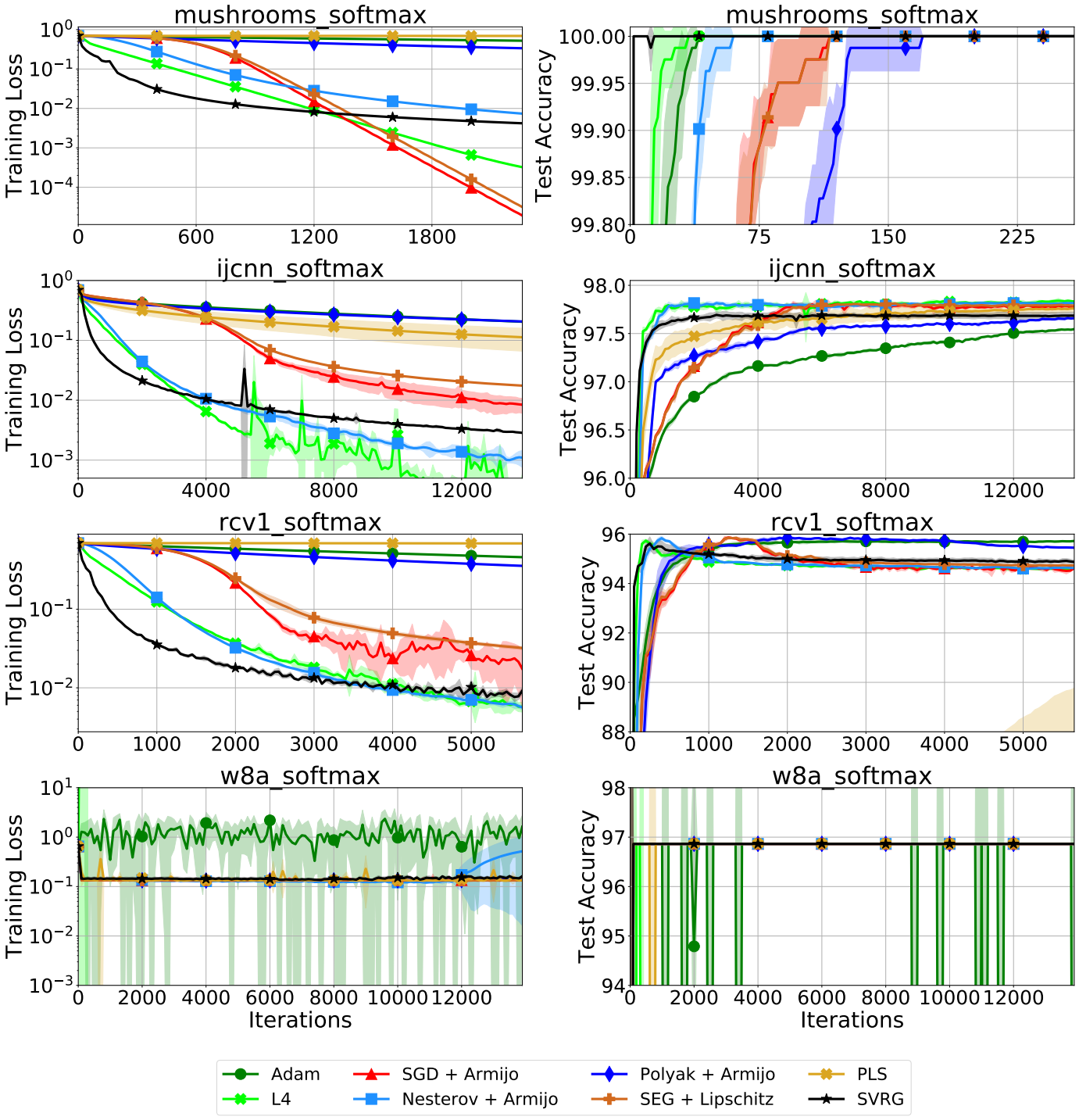

Figure 8: Binary classification using a softmax loss and RBF kernels on the mushrooms, ijcnn, rcv1, and w8a datasets. *Only* the mushrooms dataset satisfies interpolation with the selected kernel bandwidths. We compare against L4 Mom in addition to the other baseline methods; L4 Mom converges quickly on all datasets, but is unstable on ijcnn. Note that w8a dataset is particularly challenging for Adam, which shows large, periodic drops test accuracy. Our line-search methods quickly and stably converge to the global minimum despite the ill-conditioning.

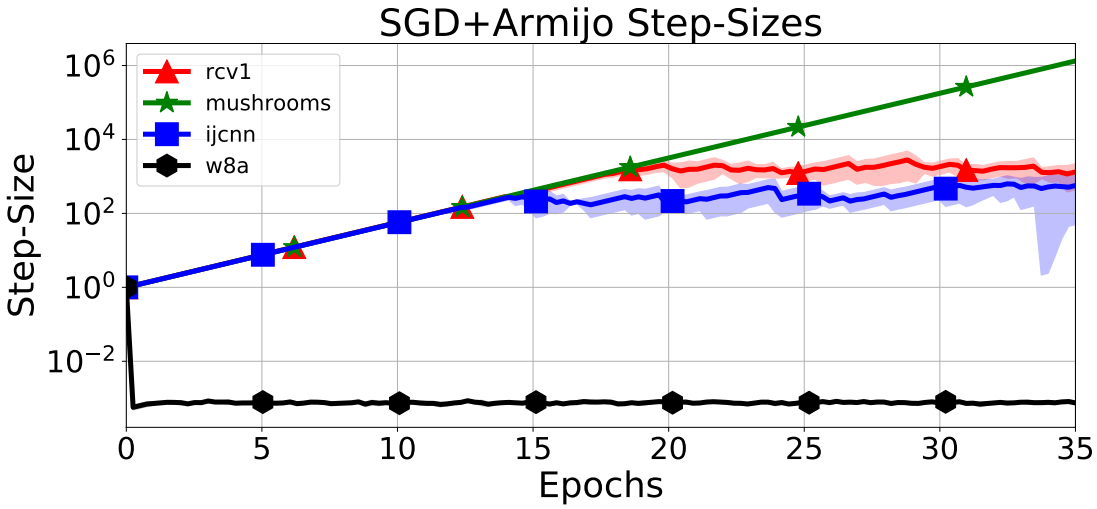

Figure 9: Variation in the step-sizes for SGD + Armijo for binary classification with softmax loss and RBF kernels on the mushrooms, ijcnn, rcv1 and w8a datasets. Recall that we use reset option 2 in Algorithm 2. The step-size grows exponentially on mushrooms, which satisfies interpolation. In contrast, the step-sizes for rcv1, ijcnn, and w8a increase or decrease to match the smoothness of the problem.

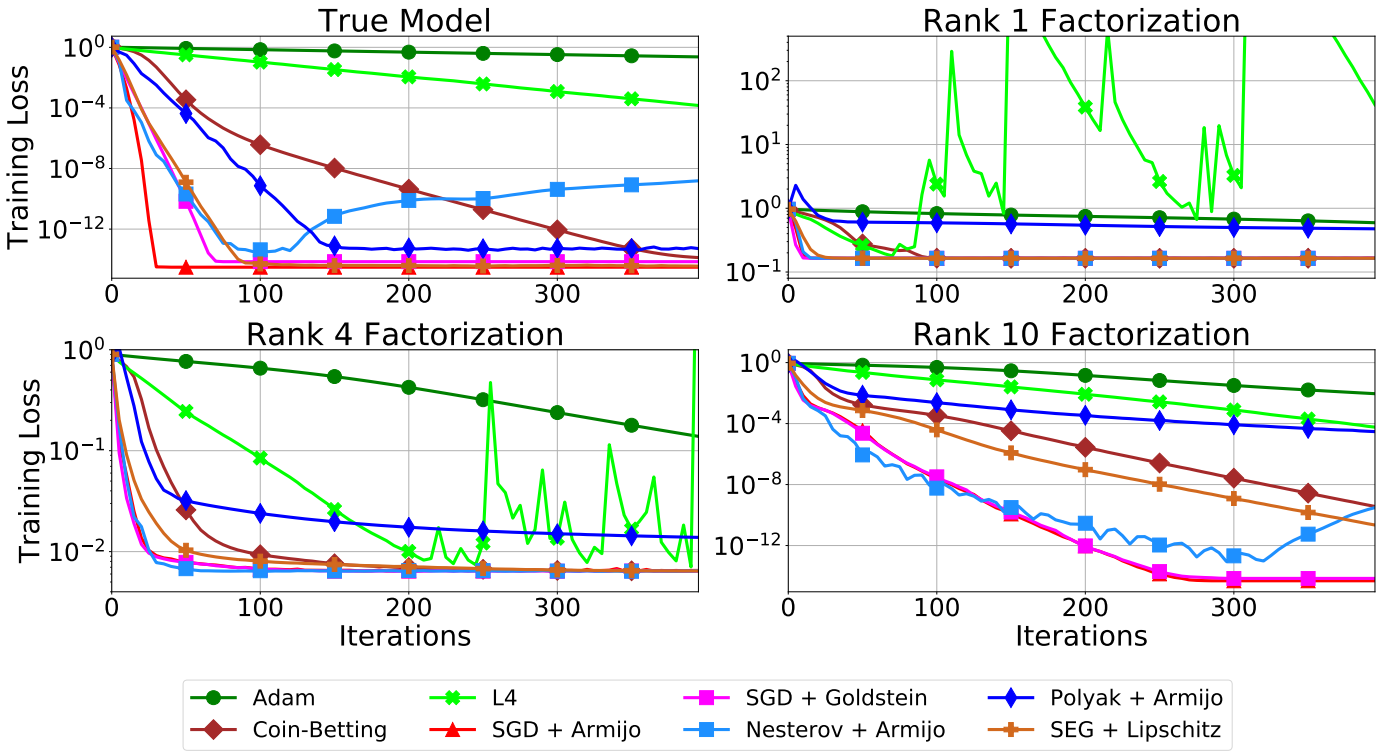

Figure 10: Matrix factorization using the true model and rank 1, 4, 10 factorizations. Rank 1 factorization is under-parametrized, while ranks 4 and 10 are over-parametrized. Only rank 10 factorization and the true model satisfy interpolation. We include L4 Mom as additional baseline optimizer. L4 Mom is unstable and does not converge on rank 1 and 4 factorization, where interpolation is not satisfied; it exhibits slow convergence on the true model and rank 10 factorization.

H Algorithm Pseudo-Code

Algorithm 3 `SGD+Goldstein`(f, w_0, $\eta_{\max}$, b, c, β, γ)

$\eta \leftarrow \eta_{\max}$
for $k = 0, \dots, T$ **do**
  $i_k \leftarrow$ sample a minibatch of size b with replacement
  while 1 **do**
    if $f_{ik}\left(w_k - \eta \nabla f_{ik}(w_k)\right) > f_{ik}(w_k) - c \cdot \eta \left\|\nabla f_{ik}(w_k)\right\|^2$ **then** ▷ check Equation (1)
      $\eta \leftarrow \beta \cdot \eta$
    else if $f_{ik}\left(w_k - \eta \nabla f_{ik}(w_k)\right) < f_{ik}(w_k) - (1-c) \cdot \eta \left\|\nabla f_{ik}(w_k)\right\|^2$ **then** ▷ check curvature condition
      $\eta \leftarrow \min\left\{\gamma \cdot \eta, \eta_{\max}\right\}$
    else
      break ▷ accept step-size η
    end if
  end while
  $w_{k+1} \leftarrow w_k - \eta \nabla f_{ik}(w_k)$ ▷ take SGD step with η
end for

return w_{k+1}

Algorithm 4 `SEG+Lipschitz`(f, w_0, $\eta_{\max}$, b, c, β, γ, `opt`)

$\eta \leftarrow \eta_{\max}$
for $k = 0, \dots, T$ **do**
  $i_k \leftarrow$ sample a minibatch of size b with replacement
  $\eta \leftarrow \texttt{reset}(\eta, \eta_{\max}, \gamma, b, k, \texttt{opt})$
  while $\left\|\nabla f_{ik}(w_k - \eta \nabla f_{ik}(w_k)) - \nabla f_{ik}(w_k)\right\| > c \ \left\|\nabla f_{ik}(w_k)\right\|$ **do** ▷ check Equation (4)
    $\eta \leftarrow \beta \cdot \eta$ ▷ backtrack by a multiple of β
  end while
  $w'_k \leftarrow w_k - \eta \nabla f_{ik}(w_k)$ ▷ take SEG step with η
  $w_{k+1} \leftarrow w_k - \eta \nabla f_{ik}(w'_k)$
end for

return w_{k+1}

Figure 11: Pseudo-code for two back-tracking line-searches used in our experiments. `SGD+Goldstein` implements SGD with the Goldstein line search described in Section 6.1 and `SEG+Lipschitz` implements SEG with the Lipschitz line-search described in Section 5. For both line-searches, we use a simple back-tracking approach that multiplies the step-size by $\beta < 1$ when the line-search is not satisified. We implement the forward search for Goldstein line-search in similar manner and multiply the step-size by $\gamma > 1$. See Algorithm 2 for the implementation of the `reset` procedure.

Algorithm 5 `Polyak+Armijo`(f, w_0, $\eta_{\max}$, b, c, β, γ, α, `opt`)

$\eta \leftarrow \eta_{\max}$
for $k = 0, \ldots, T$ **do**
  $i_k \leftarrow$ sample a minibatch of size b with replacement
  $\eta \leftarrow \texttt{reset}(\eta, \eta_{\max}, \gamma, b, k, \texttt{opt})$
  while $f_{ik}\left(w_k - \eta \nabla f_{ik}(w_k)\right) > f_{ik}(w_k) - c \cdot \eta \left\|\nabla f_{ik}(w_k)\right\|^2$ **do** ▷ check Equation (1)
    $\eta \leftarrow \beta \cdot \eta$ ▷ backtrack by a multiple of β
  end while
  $w_{k+1} \leftarrow w_k - \eta \nabla f_{ik}(w_k) + \alpha(w_k - w_{k-1})$ ▷ take SGD step with η and Polyak momentum
end for

return w_{k+1}

Algorithm 6 `Nesterov+Armijo`(f, w_0, $\eta_{\max}$, b, c, β, γ, `opt`)

$\tau \leftarrow 1$ ▷ bookkeeping for Nesterov acceleration
$\lambda \leftarrow 1$
$\lambda_{\text{prev}} \leftarrow 0$

$\eta \leftarrow \eta_{\max}$
for $k = 0, \ldots, T$ **do**
  $i_k \leftarrow$ sample a minibatch of size b with replacement
  $\eta \leftarrow \texttt{reset}(\eta, \eta_{\max}, \gamma, b, k, \texttt{opt})$
  while $f_{ik}\left(w_k - \eta \nabla f_{ik}(w_k)\right) > f_{ik}(w_k) - c \cdot \eta \left\|\nabla f_{ik}(w_k)\right\|^2$ **do** ▷ check Equation (1)
    $\eta \leftarrow \beta \cdot \eta$ ▷ backtrack by a multiple of β
  end while
  $w'_k \leftarrow w_k - \eta \nabla f_{ik}(w_k)$
  $w_{k+1} \leftarrow (1 - \tau) \cdot w'_k + \tau \cdot w_k$ ▷ Nesterov accelerated update with η

  $\texttt{temp} \leftarrow \lambda$ ▷ bookkeeping for Nesterov acceleration
  $\lambda \leftarrow \left(1 + \sqrt{1 + 4\lambda_{\text{prev}}^2}\right)/2$
  $\lambda_{\text{prev}} \leftarrow \texttt{temp}$
  $\tau \leftarrow \left(1 - \lambda_{\text{prev}}\right)/\lambda$
end for

return w_{k+1}

Figure 12: Pseudo-code for using Polyak momentum and Nesterov acceleration with our proposed line-search techniques. `Polyak+Armijo` implements SGD with Polyak momentum and Armijo line-search and `Nesterov+Armijo` implements SGD with Nesterov acceleration and Armijo line-search. Both methods are described in 6.2. See Algorithm 2 for the implementation of the `reset` procedure.