Convergence Properties of Stochastic Hypergradients

Riccardo Grazzi
Istituto Italiano di Tecnologia
University College London

Massimiliano Pontil
Istituto Italiano di Tecnologia
University College London

Saverio Salzo
Istituto Italiano di Tecnologia

Abstract

Bilevel optimization problems are receiving increasing attention in machine learning as they provide a natural framework for hyperparameter optimization and meta-learning. A key step to tackle these problems is the efficient computation of the gradient of the upper-level objective (hypergradient). In this work, we study stochastic approximation schemes for the hypergradient, which are important when the lower-level problem is empirical risk minimization on a large dataset. The method that we propose is a stochastic variant of the approximate implicit differentiation approach in (Pedregosa, 2016). We provide bounds for the mean square error of the hypergradient approximation, under the assumption that the lower-level problem is accessible only through a stochastic mapping which is a contraction in expectation. In particular, our main bound is agnostic to the choice of the two stochastic solvers employed by the procedure. We provide numerical experiments to support our theoretical analysis and to show the advantage of using stochastic hypergradients in practice.

1 Introduction

In this paper we study the following bilevel problem

$$\begin{aligned} &\min_{\lambda \in \Lambda} f(\lambda) := E(w(\lambda), \lambda) \\ &\text{subject to} \quad w(\lambda) = \Phi(w(\lambda), \lambda), \end{aligned} \tag{1}$$

which at the lower-level incorporates a (parametric) fixed-point equation. This problem is paramount in many applications, especially in machine learning and statistics, including hyperparameter optimization (Maclaurin et al., 2015; Franceschi et al., 2017; Liu et al., 2018; Lorraine et al., 2019; Elsken et al., 2019), meta-learning (Andrychowicz et al., 2016; Finn et al., 2017; Franceschi et al., 2018), and graph and recurrent neural networks (Almeida, 1987; Pineda, 1987; Scarselli et al., 2008).

In dealing with problem (1), one critical issue is to devise efficient algorithms to compute the (hyper) gradient of the function f, so as to allow using gradient based approaches to find a solution. The computation of the hypergradient via approximate implicit differentiation (AID) (Pedregosa, 2016) requires one to solve two subproblems: (i) the lower-level problem in (1) and (ii) a linear system which arises from the implicit expression for $\nabla f(\lambda)$. However, especially in large scale scenarios, solving those subproblems exactly might either be impossible or too expensive, hence, iterative approximation methods are often used. In (Grazzi et al., 2020), under the assumption that, for every $\lambda \in \Lambda$, the mapping $\Phi(\cdot, \lambda)$ in (1) is a contraction, a comprehensive analysis of the iteration complexity of the hypergradient computation for several popular deterministic algorithms was provided. Here, instead, we address such iteration complexity for stochastic methods. This study is of fundamental importance since in many practical scenarios $\Phi(w, \lambda)$ is expensive to compute, e.g., when it has a sum structure with a large number of terms. In this situation stochastic approaches become the method of choice. For example, in large scale hyperparameter optimization and neural architecture search (Maclaurin et al., 2015; Lorraine et al., 2019; Liu et al., 2018), solving the lower-level problem requires minimizing a training objective over a large dataset, which is usually done approximately through SGD and its extensions. Our contributions can be summarized as follows.

- We devise a stochastic estimator $\hat{\nabla} f(\lambda)$ of the true gradient, based on the AID technique, together with an explicit bound for the related mean square error. The bound is agnostic with respect to the

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

stochastic methods solving the related subproblems, so that can be applied to several algorithmic solutions; see Theorem 3.4.

- We study the convergence of a general stochastic fixed-point iteration method which extends and improves previous analysis of SGD for strongly convex functions and can be applied to solve both subproblems associated to the AID approach. These results, which are interesting in their own right, are given in Theorems 4.1 and 4.2.

Proofs of the results presented in the paper can be found in the supplementary material.

Related Work Pedregosa (2016) introduced an efficient class of deterministic methods to compute the hypergradient through AID together with asymptotic convergence results. Rajeswaran et al. (2019); Grazzi et al. (2020) extended this analysis providing iteration complexity bounds. AID methods require to iteratively evaluate Φ and its derivatives. In this work, we extend these methods by replacing those exact evaluations with unbiased stochastic approximations and provide iteration complexity bounds in this scenario.
Another class of methods (ITD) computes the hypergradient by differentiating through the inner optimization scheme (Maclaurin et al., 2015; Franceschi et al., 2017, 2018). Iteration complexity results for the deterministic case are given in (Grazzi et al., 2020), while we are not aware of any convergence results in the stochastic setting. Here, we focus entirely on AID methods, leaving the investigation of stochastic ITD methods for future work.
An interesting special case of the bilevel problem (1) is when $f(\lambda) = \min_w E(w, \lambda)$. This scenario occurs for example in regularized meta-learning, where the properties of a simple stochastic hypergradient estimator have been studied extensively (Denevi et al., 2019a,b; Zhou et al., 2019). In this setting, Ablin et al. (2020) analyze, among others, implicit differentiation techniques for approximating the gradient of f, including stochastic approaches. However, the proposed estimator assumes to solve the related linear system exactly, which is often impractical. In this work, we focus on the more general setting of bilevel problem (1), devising algorithmic solutions that are fully stochastic, in the sense that also the subproblem involving the linear system is solved by a stochastic method.
Finally, stochastic algorithms for hypergradient computation in bilevel optimization problems have been studied in (Couellan and Wang, 2016; Ghadimi and Wang, 2018). There, the authors provide convergence rates for a whole bilevel optimization procedure using stochastic oracles both from the upper-level and the lower-level objectives. In particular, the method used by Ghadimi and Wang (2018) to approximate the hypergradient can be seen as a special case of our method with two particular choices of the stochastic solvers.[1]

Notation We denote by $\|\cdot\|$ either the Euclidean norm or the spectral norm (when applied to matrices). The transpose and the inverse of a given matrix A, is denoted by $A^\top$ and A^{-1} respectively. For a real-valued function $g\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, we denote by $\nabla_1 g(x,y) \in \mathbb{R}^n$ and $\nabla_2 g(x,y) \in \mathbb{R}^m$, the partial derivatives w.r.t. the first and second variable respectively. For a vector-valued function $h\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ we denote by $\partial_1 h(x,y) \in \mathbb{R}^{k\times n}$ and $\partial_2 h(x,y) \in \mathbb{R}^{k\times m}$ the partial Jacobians w.r.t. the first and second variables respectively. For a random variable X we denote by $\mathbb{E}[X]$ and $\mathbb{V}[X]$ its expectation and variance respectively. Finally, given two random variables X and Y, the conditional variance of X given Y is $\mathbb{V}[X \mid Y] := \mathbb{E}[\|X - \mathbb{E}[X \mid Y]\|^2 \mid Y]$. In the following, for the reader's convenience, we provide a list of the main functions and constants used in the subsequent analysis.

Table 1: Table of Notation

Symbol(s)	Description
E	Upper-level objective
Φ	Fixed-point map
$\hat{\Phi}$	Unbiased estimator of Φ
$\hat{\ell}$	Estimator of the lower-level objective
q_λ	Contraction constant of $\Phi(\cdot, \lambda)$
$L_{E,\lambda}$	Lipschitz constant of $E(\cdot, \lambda)$
$\nu_{1,\lambda}, \nu_{2,\lambda}$	Lipschitz const. of $\partial_1\Phi(\cdot,\lambda), \partial_2\Phi(\cdot,\lambda)$
$\mu_{1,\lambda}, \mu_{2,\lambda}$	Lipschitz const. of $\nabla_1 E(\cdot,\lambda), \nabla_2 E(\cdot,\lambda)$
$L_{\hat{\Phi},\lambda}$	Lipschitz const. of $\hat{\Phi}(\cdot,\lambda,\zeta)$
$m_{2,\lambda}$	Bound on the variance of $\partial_2\hat{\Phi}(w,\lambda,\zeta)$
$\rho_\lambda(t), \sigma_\lambda(k)$	Convergence rates for the two subproblems: t, k are the number of iterations of the solvers.

2 Stochastic Hypergradient Approximation

In this section we describe a general method for generating a stochastic approximation of the (hyper) gradient of f in (1). We assume that Φ is defined by an expectation of a given function $\hat{\Phi}$, that is, we consider bilevel problems of type (1) with

$$\Phi(w, \lambda) = \mathbb{E}[\hat{\Phi}(w(\lambda), \lambda, \zeta)], \tag{2}$$

[1] Specifically they use SGD with decreasing step sizes for the lower-level problem (which is a minimization problem) and, for the linear system, a stochastic routine derived from the Neumann series approximation of the matrix inverse.

where ζ is a random variable taking values in a suitable measurable space. A special case of (1)-(2), which occurs often in machine learning, is

$$\begin{aligned} &\min_{\lambda\in\Lambda} f(\lambda) := E(w(\lambda),\lambda) \\ &\text{subject to } \; w(\lambda) = \operatorname{argmin}_w \mathbb{E}[\hat{\ell}(w,\lambda,\zeta)], \end{aligned} \tag{3}$$

where $w \mapsto \mathbb{E}[\hat{\ell}(w,\lambda,\zeta)]$ is strongly convex and Lipschitz smooth, for every $\lambda \in \Lambda$. Indeed, (3) follows from (1) and (2) by choosing $\hat{\Phi}(w,\lambda,\zeta) = w - \alpha_\lambda \nabla \hat{\ell}(w,\lambda,\zeta)$, for any $\alpha_\lambda > 0$.

In the rest of the paper we will consider the following assumptions[2].

Assumption A. *The set $\Lambda \subseteq \mathbb{R}^m$ is closed and convex and the mappings $\Phi\colon \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ and $E\colon \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ are differentiable. For every $\lambda \in \Lambda$, we assume*

(i) *$\Phi(\cdot,\lambda)$ is a contraction, i.e., $\|\partial_1\Phi(w,\lambda)\| \le q_\lambda$ for some $q_\lambda < 1$ and for all $w \in \mathbb{R}^d$.*

(ii) *$\partial_1\Phi(\cdot,\lambda)$ and $\partial_2\Phi(\cdot,\lambda)$ are Lipschitz continuous with constants $\nu_{1,\lambda}$ and $\nu_{2,\lambda}$ respectively.*

(iii) *$\nabla_1 E(\cdot,\lambda)$ and $\nabla_2 E(\cdot,\lambda)$ are Lipschitz continuous with constants $\mu_{1,\lambda}$ and $\mu_{2,\lambda}$ respectively.*

(iv) *$E(\cdot,\lambda)$ is Lipschitz continuous with constant $L_{E,\lambda}$.*

Under Assumption A, $\Phi(\cdot,\lambda)$ has a unique fixed point $w(\lambda)$ and the hypergradient is given by

$$\begin{aligned} \nabla f(\lambda) = \nabla_2 E(w(\lambda),\lambda) \\ + \partial_2\Phi(w(\lambda),\lambda)^\top v(w(\lambda),\lambda), \end{aligned} \tag{4}$$

where,

$$v(w,\lambda) := \left(I - \partial_1\Phi(w,\lambda)^\top\right)^{-1} \nabla_1 E(w,\lambda). \tag{5}$$

This formula follows by differentiating the fixed point conditions for the lower-level problem and noting that, because of Assumption A(i), $I - \partial_1\Phi(w,\lambda)^\top$ is invertible (see Lemma B.6).

We also consider the following properties for $\hat{\Phi}$.

Assumption B. *The random variable ζ takes values in measurable space $\mathcal{Z}$ and $\hat{\Phi} : \mathbb{R}^d \times \mathbb{R}^m \times \mathcal{Z} \mapsto \mathbb{R}^d$ is a measurable function, differentiable w.r.t. the first two arguments, and such that, for all $w \in \mathbb{R}^d$ and $\lambda \in \Lambda$*

(i) *$\mathbb{E}[\hat{\Phi}(w,\lambda,\zeta)] = \Phi(w,\lambda)$ and $\mathbb{E}[\|\hat{\Phi}(w,\lambda,\zeta)\|^2] < \infty$.*

(ii) *For $j \in \{1,2\}$, $\mathbb{E}[\partial_j\hat{\Phi}(w,\lambda,\zeta)] = \partial_j \mathbb{E}[\hat{\Phi}(w,\lambda,\zeta)]$ and $\mathbb{E}[\|\partial_j\hat{\Phi}(w,\lambda,\zeta)\|^2] < +\infty$.*

[2]Similar assumptions, except for Assumption A(iv), are also considered in Grazzi et al. (2020).

Algorithm 1: Stochastic Implicit Differentiation (SID)

1. Let $t \in \mathbb{N}$ and compute $w_t(\lambda)$ by t steps of a stochastic algorithm that approximates $w(\lambda)$.
2. Let $k \in \mathbb{N}$ and Compute $v_k(w_t(\lambda),\lambda)$ by k steps of a stochastic solver for the linear system
$$(I - \partial_1\Phi(w_t(\lambda),\lambda)^\top)v = \nabla_1 E(w_t(\lambda),\lambda). \tag{6}$$
3. Compute the approximate gradient as
$$\begin{aligned} \hat{\nabla} f(\lambda) :=& \nabla_2 E(w_t(\lambda),\lambda) \\ &+ \partial_2\hat{\Phi}(w_t(\lambda),\lambda,\zeta)^\top v_k(w_t(\lambda),\lambda). \end{aligned}$$

(iii) *For every $z \in \mathcal{Z}$, $\|\partial_1\hat{\Phi}(w,\lambda,z)\| \le L_{\tilde{\Phi},\lambda}$ for some constant $L_{\tilde{\Phi},\lambda} \ge 0$ (which does not depend on w).*

(iv) *$\mathbb{V}[\partial_2\hat{\Phi}(w,\lambda,\zeta)] \le m_{2,\lambda}$, for some $m_{2,\lambda} \ge 0$ (which does not depend on w).*

Motivated by (4)-(5), we consider to have at our disposal two stochastic solvers which exploit $\hat{\Phi}$: one for the lower-level problem in (1) which generates a stochastic process $w_t(\lambda)$ estimating $w(\lambda)$ and another for the linear system

$$(I - \partial_1\Phi(w,\lambda)^\top)v = \nabla_1 E(w,\lambda), \quad \text{with } w \in \mathbb{R}^d, \tag{7}$$

generating a stochastic process $v_k(w,\lambda)$ approximating the solution $v(w,\lambda)$ of (7). Then, the stochastic approximation to the hypergradient is defined as

$$\begin{aligned} \hat{\nabla} f(\lambda) :=& \nabla_2 E(w_t(\lambda),\lambda) \\ &+ \partial_2\hat{\Phi}(w_t(\lambda),\lambda,\zeta)^\top v_k(w_t(\lambda),\lambda). \end{aligned} \tag{8}$$

We also suppose that, for every $w \in \mathbb{R}^d$, $w_t(\lambda)$, $v_k(w,\lambda)$, and ζ are mutually independent. The procedure, which we call SID, is summarized in Algorithm 1. In Section 5 we will give a way to generate the stochastic processes $(w_t(\lambda))_{t\in\mathbb{N}}$ and $(v_k(w,\lambda)\lambda)_{k\in\mathbb{N}}$.

3 Mean Square Error Bound for SID

In this section, we derive a bound for the mean square error of the SID estimator, i.e.,

$$\mathrm{MSE}_{\hat{\nabla} f} := \mathbb{E}[\|\hat{\nabla} f(\lambda) - \nabla f(\lambda)\|^2]. \tag{9}$$

To that purpose, we require the stochastic procedures at point 1 and 2 of Algorithm 1 to have non-asymptotic convergence rates in mean square. This is the content of the following assumption.

Assumption C. *For every $\lambda \in \Lambda$, $t, k \geq 1$ and $w \in \mathbb{R}^d$, the random variables $v_k(w, \lambda)$, $w_t(\lambda)$ and ζ are mutually independent and*

$$\mathbb{E}[\|w_t(\lambda) - w(\lambda)\|^2] \leq \rho_\lambda(t)$$
$$\mathbb{E}[\|v_k(w, \lambda) - v(w, \lambda)\|^2] \leq \sigma_\lambda(k),$$

where $\rho_\lambda : \mathbb{N} \mapsto \mathbb{R}_+$ and $\sigma_\lambda : \mathbb{N} \mapsto \mathbb{R}_+$.

This assumption is often satisfied in applications, e.g., in problems of type (3), when the lower-level objective is strongly convex and Lipschitz smooth. In Section 4 we describe a general stochastic fixed-point method from which, in Section 5, we will derive a stochastic implicit differentiation method featuring the rates required in Assumption C.

In order to analyze the quantity in (9), we start with the standard bias-variance decomposition (see Lemma B.2) as follows

$$\mathrm{MSE}_{\hat{\nabla} f} = \underbrace{\|\mathbb{E}[\hat{\nabla} f(\lambda)] - \nabla f(\lambda)\|^2}_{\text{bias}} + \underbrace{\mathbb{V}[\hat{\nabla} f(\lambda)]}_{\text{variance}}. \tag{10}$$

Then, using the law of total variance (see Lemma B.4), we write the mean square error as below

$$\begin{aligned}\mathrm{MSE}_{\hat{\nabla} f} = &\underbrace{\|\mathbb{E}[\hat{\nabla} f(\lambda)] - \nabla f(\lambda)\|^2}_{\text{bias}} \\ &+ \underbrace{\mathbb{E}[\mathbb{V}[\hat{\nabla} f(\lambda) \mid w_t(\lambda)]] + \mathbb{V}[\mathbb{E}[\hat{\nabla} f(\lambda) \mid w_t(\lambda)]]}_{\text{variance}}.\end{aligned} \tag{11}$$

In the following we will bound each term on the right-hand side of (11) individually. The next result serves to control the bias term.

Theorem 3.1. *Suppose that Assumptions A,B, and C are satisfied. Let $\lambda \in \Lambda$, $t, k \in \mathbb{N}$ and set*

$$\hat{\Delta}_w := \|w_t(\lambda) - w(\lambda)\|, \;\; L_{\Phi,\lambda} := \|\partial_2 \Phi(w(\lambda), \lambda)\|,$$

$$c_{1,\lambda} = \mu_{2,\lambda} + \frac{\mu_{1,\lambda} L_{\Phi,\lambda} + \nu_{2,\lambda} L_{E,\lambda}}{1 - q_\lambda} + \frac{\nu_{1,\lambda} L_{E,\lambda} L_{\Phi,\lambda}}{(1 - q_\lambda)^2}.$$

Then the following hold.

(i) $\big\|\mathbb{E}[\hat{\nabla} f(\lambda) \mid w_t(\lambda)] - \nabla f(\lambda)\big\|$
$\leq c_{1,\lambda} \hat{\Delta}_w + L_{\Phi,\lambda} \sqrt{\sigma_\lambda(k)} + \nu_{2,\lambda} \hat{\Delta}_w \sqrt{\sigma_\lambda(k)}.$

(ii) $\|\mathbb{E}[\hat{\nabla} f(\lambda)] - \nabla f(\lambda)\|$
$\leq c_{1,\lambda} \sqrt{\rho_\lambda(t)} + L_{\Phi,\lambda} \sqrt{\sigma_\lambda(k)} + \nu_{2,\lambda} \sqrt{\rho_\lambda(t)} \sqrt{\sigma_\lambda(k)}.$

The following two theorems provide bounds for the two components of the variance in (11).

Theorem 3.2. *Suppose that Assumptions A,B, and C are satisfied. Let $\lambda \in \Lambda$, $t, k \in \mathbb{N}$ and set $L_{\Phi,\lambda} := \|\partial_2 \Phi(w(\lambda), \lambda)\|$. Then*

$$\begin{aligned}\mathbb{E}\big[\mathbb{V}[\hat{\nabla} f(\lambda) | w_t(\lambda)]\big] \leq\; & 2 \frac{m_{2,\lambda} L_{E,\lambda}^2}{(1 - q_\lambda)^2} \\ &+ 2(L_{\Phi,\lambda}^2 + m_{2,\lambda}) \sigma_\lambda(k) \\ &+ 2 \nu_{2,\lambda}^2 \rho_\lambda(t) \sigma_\lambda(k).\end{aligned} \tag{12}$$

Theorem 3.3. *Suppose that Assumptions A,B, and C are satisfied. Let $\lambda \in \Lambda$, and $t, k \in \mathbb{N}$. Then*

$$\begin{aligned}\mathbb{V}[\mathbb{E}[\hat{\nabla} f(\lambda) \mid w_t(\lambda)]] \leq\; & 3\big(c_{1,\lambda}^2 \rho_\lambda(t) + L_{\Phi,\lambda}^2 \sigma_\lambda(k) \\ &+ \nu_{2,\lambda}^2 \rho_\lambda(t) \sigma_\lambda(k)\big),\end{aligned}$$

where $c_{1,\lambda}$ and $L_{\Phi,\lambda}$ are defined as in Theorem 3.1.

Finally, combining the above three results, we give the promised bound on the mean square error for the estimator of the hypergradient.

Theorem 3.4 (MSE bound for SID). *Suppose that Assumptions A,B, and C are satisfied. Let $\lambda \in \Lambda$, and $t, k \in \mathbb{N}$. Then*

$$\begin{aligned}MSE_{\hat{\nabla} f} \leq\; & 2 \frac{m_{2,\lambda} L_{E,\lambda}^2}{(1 - q_\lambda)^2} + 6 c_{1,\lambda}^2 \rho_\lambda(t) \\ &+ 2(4 L_{\Phi,\lambda}^2 + m_{2,\lambda}) \sigma_\lambda(k) \\ &+ 8 \nu_{2,\lambda}^2 \rho_\lambda(t) \sigma_\lambda(k).\end{aligned} \tag{13}$$

where $c_{1,\lambda}$ is defined as in Theorem 3.1. In particular, if $\lim_{t \to \infty} \rho_\lambda(t) = 0$ and $\lim_{k \to \infty} \sigma_\lambda(k) = 0$, then

$$\lim_{t,k \to \infty} MSE_{\hat{\nabla} f} \leq 2 \frac{m_{2,\lambda} L_{E,\lambda}^2}{(1 - q_\lambda)^2}. \tag{14}$$

Proof. The statement follows from the decomposition (11) and Theorems 3.1, 3.2, and 3.3. □

In the following we make few comments related to the above results. First, it follows from (ii) in Theorem 3.1 that, if $\lim_{t \to \infty} \rho_\lambda(t) = 0$ and $\lim_{k \to \infty} \sigma_\lambda(k) = 0$, the estimator $\hat{\nabla} f(\lambda)$ is asymptotically unbiased as $t, k \to +\infty$. Next, the bound (13) in Theorem 3.4 provides the iteration complexity of the SID method (Algorithm 1). This result is a stochastic version of what was obtained by Grazzi et al. (2020) concerning approximate implicit differentiation methods. Finally, note that it follows from (14) that the mean square error cannot be made arbitrarily small unless the variance term $\mathbb{V}[\partial_2 \hat{\Phi}(w, \lambda, \zeta)]$ (controlled by $m_{2,\lambda}$) is zero. This may seem a limitation of the method. However, since SID uses $\partial_2 \hat{\Phi}(w, \lambda, \zeta)$ only once at the end of the procedure, one could modify the algorithm by sampling ζ several times so to reduce the variance of $\partial_2 \hat{\Phi}(w, \lambda, \zeta)$ or, when possible, even compute $\partial_2 \Phi(w, \lambda)$ exactly, with little increase on the overall

cost. Additionally, we stress that in several applications that variance term is zero. Indeed, this occurs each time $\hat{\Phi}$ is of the form

$$\hat{\Phi}(w, \lambda, \zeta) = \hat{\Phi}_1(w, \zeta) + \hat{\Phi}_2(w, \lambda), \tag{15}$$

meaning that, $\hat{\Phi}$ depends on the random variable ζ and on the hyperparameter λ in a separate manner. For instance, this is the case when we want to optimize the regularization hyperparameters in regularized empirical risk minimization problems, where usually, the random variable ζ affects only the data term.

4 Stochastic fixed-point iterations

In this section we address the convergence of stochastic fixed-point iteration methods which can be applied in a similar manner to solve both subproblems in Algorithm 1 (see Section 5). We consider the general situation of computing the fixed point of a contraction mapping which is accessible only through a stochastic oracle. The results are inspired by the analysis of the SGD algorithm for strongly convex and Lipschitz smooth functions given in (Bottou et al., 2018), but extended to our more general setting. Indeed, by a more accurate computation of the contraction constant of the gradient descent mapping, we are able to improve the convergence rates and increase the stepsizes given in the above cited paper. See Corollary 4.1 and the subsequent remark. We stress that the significance of the results presented in this section goes beyond the bilevel setting (1)-(2) and may be of interest per se.

We start with the assumption below.

Assumption D. *Let ζ be a random variable with values in a measurable space $\mathcal{Z}$. Let $T : \mathbb{R}^d \mapsto \mathbb{R}^d$ and $\hat{T} : \mathbb{R}^d \times \mathcal{Z} \mapsto \mathbb{R}^d$ be such that*

(i) $\forall w_1, w_2 \in \mathbb{R}^d$, $\|T(w_1) - T(w_2)\| \leq q\|w_1 - w_2\|$, *with $q < 1$.*

(ii) $\forall w \in \mathbb{R}^d$, $\mathbb{E}[\hat{T}(w, \zeta)] = T(w)$

(iii) $\forall w \in \mathbb{R}^d$, $\mathbb{V}[\hat{T}(w, \zeta)] \leq \sigma_1 + \sigma_2\|T(w) - w\|^2$.

The above assumptions are in line with those made by Bottou et al. (2018) for the case of stochastic minimization of a strongly convex and Lipschitz smooth function.

Since T is a contraction, there exists a unique $w^* \in \mathbb{R}^d$ such that

$$w^* = T(w^*). \tag{16}$$

We consider the following random process which corresponds to a stochastic version of the Krasnoselskii-Mann iteration for contractive operators. Let $(\zeta_t)_{t\in\mathbb{N}}$ be a sequence of independent copies of ζ. Then, starting from $w_0 \in \mathbb{R}^d$ we set

$$(\forall\, t \in \mathbb{N}) \quad w_{t+1} = w_t + \eta_t(\hat{T}(w_t, \zeta_t) - w_t). \tag{17}$$

The following two results provide non-asymptotic convergence rates for the procedure (17) for two different strategies about the step-sizes η_t.

Theorem 4.1 (Constant step-size). *Let Assumption D hold and suppose that $\eta_t = \eta \in \mathbb{R}_{++}$, for every $t \in \mathbb{N}$, and that*

$$\eta \leq \frac{1}{1 + \sigma_2}.$$

Let $(w_t)_{t\in\mathbb{N}}$ be generated according to algorithm (17) and set $MSE_{w_t} := \mathbb{E}[\|w_t - w^*\|^2]$. Then, for all $t \in \mathbb{N}$,

$$MSE_{w_t} \leq (1 - \eta(1 - q^2))^t \left(MSE_{w_0} - \frac{\eta\sigma_1}{1 - q^2} \right) + \frac{\eta\sigma_1}{1 - q^2}. \tag{18}$$

In particular, $\lim_{t\to\infty} MSE_{w_t} \leq \eta\sigma_1/(1 - q^2)$.

Theorem 4.2 (Decreasing step-sizes). *Let Assumption D hold and suppose that for every $t \in \mathbb{N}$*

$$\eta_t \leq \frac{1}{1 + \sigma_2}, \quad \sum_{t=1}^{\infty} \eta_t = \infty, \quad \sum_{t=1}^{\infty} \eta_t^2 < \infty. \tag{19}$$

Let $(w_t)_{t\in\mathbb{N}}$ be generated according to Algorithm (17). Then

$$w_t \to w^* \qquad \mathbb{P}\text{-a.s.}$$

Moreover, if $\eta_t = \beta/(\gamma + t)$, with $\beta > 1/(1 - q^2)$ and $\gamma \geq \beta(1 + \sigma_2)$, then we have

$$\mathbb{E}[\|w_t - w^*\|^2] \leq \frac{c}{\gamma + t}, \tag{20}$$

where

$$c := \max\left\{ \gamma\mathbb{E}[\|w_0 - w^*\|^2], \frac{\beta^2\sigma_1}{\beta(1 - q^2) - 1} \right\}.$$

We will now comment on the choice of the stepsizes in algorithm (17). Theorem 4.1 and 4.2 suggest that it may be convenient to start the algorithm with a constant stepsize. Then, once reached a mean square error approximately less than $\eta\sigma_1/(1 - q^2)$, the stepsizes should change regime and start decreasing according to Theorem 4.2. More precisely, in the first phase it is recommended to set $\eta = 1/(1 + \sigma_2)$ in order to maximize the stepsize. Then, the second phase should be initialized with w_0 such that $\mathrm{MSE}_{w_0} \leq \sigma_1/[(1 + \sigma_2)(1 - q^2)]$ and $\gamma = \beta(1 + \sigma_2)$ so that

$$\gamma\mathbb{E}[\|w_0 - w^*\|^2] \leq \frac{\beta^2\sigma_1}{\beta(1 - q^2) - 1}.$$

In this situation, c will be dominated by its second term, which is minimized when $\beta = 2/(1-q^2)$. Similar suggestions are made in (Bottou et al., 2018).

In the following, partly inspired by the analysis of Nguyen et al. (2019), we show that, with an additional Lipschitz assumption on $\hat{T}$, which is commonly verified in practice, Assumption D(iii) on the variance of the estimator is satisfied. The following Assumption E is an extension of Assumption 2 in (Nguyen et al., 2019).

Assumption E. *There exists $L_{\hat{T}} \geq 0$ such that, for every $w_1, w_2 \in \mathbb{R}^d$ and for every $z \in \mathcal{Z}$*

$$\|\hat{T}(w_1, z) - \hat{T}(w_2, z)\| \leq L_{\hat{T}} \|w_1 - w_2\|.$$

Theorem 4.3. *Suppose that Assumption E and Assumption D*(i)(ii) *hold. Then Assumption D*(iii) *holds. In particular, for every $w \in \mathbb{R}^d$,*

$$\mathbb{V}[\hat{T}(w, \zeta)] \leq \underbrace{2\mathbb{V}[\hat{T}(w^*, \zeta)]}_{\sigma_1} + \underbrace{2\frac{L_{\hat{T}}^2 + q^2}{(1-q)^2}}_{\sigma_2} \|T(w) - w\|^2.$$

We now discuss the popular case of SGD and make a comparison with the related results by Bottou et al. (2018). We assume that $\hat{T}(w, \zeta) = w - \alpha \nabla \hat{\ell}(w, \zeta)$, for a suitable $\alpha > 0$. With this choice, algorithm (17) becomes

$$(\forall\, t \in \mathbb{N}) \quad w_{t+1} = w_t - \eta_t \alpha \nabla_1 \ell(w_t, \zeta_t), \tag{21}$$

which is exactly stochastic gradient descent. We have the following assumption on $\hat{\ell}$.

Assumption F. *$\hat{\ell} : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ is twice continuously differentiable w.r.t. the first variable. Let $\ell(w) := \mathbb{E}[\hat{\ell}(w, \zeta)]$.*

(i) *$\ell(w)$ is τ strongly convex and L-smooth*

(ii) *$\forall w \in \mathbb{R}^d$, $\mathbb{V}[\nabla \hat{\ell}(w, \zeta)] \leq \sigma_1' + \sigma_2' \|\nabla \ell(w)\|^2$.*

Corollary 4.1. *Let Assumption F hold and let $(w_t)_{t \in \mathbb{N}}$ be generated according to algorithm (21) with $\eta_t = \eta \leq 1/(1 + \sigma_2')$. Then*

$$\mathbb{E}[\|w_t - w^*\|^2] \leq r_1^t \big(\mathbb{E}[\|w_0 - w^*\|^2] - r_2\big) + r_2, \tag{22}$$

where

$$r_1 := \begin{cases} 1 - \dfrac{\eta\tau}{L}\Big(2 - \dfrac{\tau}{L}\Big) & \text{if } \alpha = 1/L \\ 1 - 4\dfrac{\eta\tau L}{(L+\tau)^2} & \text{if } \alpha = 2/(L+\tau). \end{cases}$$

$$r_2 := \begin{cases} = \dfrac{\eta\sigma_1'}{\tau(2L-\tau)} & \text{if } \alpha = 1/L \\ \dfrac{\eta\sigma_1'}{\tau L} & \text{if } \alpha = 2/(L+\tau). \end{cases}$$

Moreover, let $\eta_t = \beta/(\gamma + t)$, where

$$\beta > \begin{cases} \dfrac{L^2}{\tau(2L-\tau)} & \text{if } \alpha = 1/L \\ \dfrac{(L+\tau)^2}{4\tau L} & \text{if } \alpha = 2/(L+\tau) \end{cases} \tag{23}$$

and $\gamma \geq \beta(1 + \sigma_2')$. Then, for all $t \in \mathbb{N}$, we have

$$\mathbb{E}[\|w_t - w^*\|^2] \leq \frac{\max\{\gamma \mathbb{E}[\|w_0 - w^*\|^2], r_3\}}{\gamma + t}, \tag{24}$$

where

$$r_3 := \begin{cases} \dfrac{\beta^2 \sigma_1'}{\beta\tau(2L-\tau) - L^2} & \text{if } \alpha = 1/L \\ \dfrac{4\beta^2\sigma_1'}{4\beta\tau L - (L+\tau)^2} & \text{if } \alpha = 2/(L+\tau). \end{cases}$$

Remark 4.1. *In (Bottou et al., 2018), under Assumption F a rate equal to (22) is obtained, but with $\alpha = 1/L$ and*

$$r_1 = 1 - \eta\frac{\tau}{L} \quad \text{and} \quad r_2 = \frac{\eta\sigma_1'}{2\tau}. \tag{25}$$

We see then, that Corollary 4.1 provides better rates. Also, our analysis allows choosing the larger (and optimal) stepsize $2/(L+\tau)$.

Remark 4.2. *In Assumption F, suppose that ζ takes values in $\mathcal{Z} = \{1, \dots, n\}$ with uniform distribution and that for every $i \in \{1, \dots, n\}$, $\hat{\ell}(\cdot, i)$ is strongly convex with modulus τ. This is, for instance, the case of the regularized empirical risk functional,*

$$\hat{\ell}(w, i) = \psi(y_i w^\top x_i) + \frac{\tau}{2}\|w\|^2, \tag{26}$$

where $(x_i, y_i)_{1 \leq i \leq n} \in (\mathbb{R}^d \times \{1, 2\})^n$ is the training set. Then, if the loss function ψ is Lipschitz continuous, as is the case, e.g., of the logistic loss, we have

$$\begin{aligned} \mathbb{V}[\nabla \hat{\ell}(w, i)] &= \mathbb{V}_{i \sim U[\mathcal{Z}]}[\psi'(y_i w^\top x_i) y_i x_i] \\ &\leq \mathrm{Lip}(\psi)^2 \mathbb{E}_{i \sim U[\mathcal{Z}]}[\|x_i\|^2], \end{aligned} \tag{27}$$

so that Assumption F(ii) *is satisfied with $\sigma_2' = 0$.*

5 Solving the Subproblems in SID

We are now ready to show how to generate the sequences $w_t(\lambda)$ and $v_k(w_t(\lambda), \lambda)$ required by Algorithm 1. Let ζ' be a random variable with values in $\mathcal{Z}$ satisfying Assumption B(i). Let $(\zeta_i)_{i \in \mathbb{N}}$ and $(\hat{\zeta}_i)_{i \in \mathbb{N}}$ be independent copies of ζ' and independent from each other, and let $(\eta_{\lambda,i})_{i \in \mathbb{N}}$ be a sequence of stepsizes such that $\sum_{i=0}^{\infty} \eta_{\lambda,i} = +\infty$ and $\sum_{i=0}^{\infty} \eta_{\lambda,i}^2 < +\infty$. For every $w \in \mathbb{R}^d$ we let $w_0(\lambda) = v_0(w, \lambda) = 0$, and, for $k, t \in \mathbb{N}$,

$$w_{t+1}(\lambda) := w_t(\lambda) + \eta_{\lambda,t}(\hat{\Phi}(w_t(\lambda), \lambda, \zeta_t) - w_t(\lambda)) \tag{28}$$

and

$$v_{k+1}(w,\lambda) := v_k(w,\lambda) + \eta_{\lambda,k}(\hat{\Psi}_w(v_k(w,\lambda),\lambda,\hat{\zeta}_k) - v_k(w,\lambda)), \tag{29}$$

where $\hat{\Psi}_w(v,\lambda,z) := \partial_1\hat{\Phi}(w,\lambda,z)^\top v + \nabla_1 E(w,\lambda)$.

We note that if the Jacobian-vector product above is computed using reverse mode automatic differentiation, the costs of evaluating $\hat{\Psi}_w$ and $\hat{\Phi}$ are of the same order of magnitude. Furthermore, thanks to the definition of $\hat{\Psi}$, we can solve both subproblems in Algorithm 1 using the procedure described in Section 4. In particular, if we set $\eta_{\lambda,t} = \eta_{\lambda,k}$, we can obtain similar convergence guarantees for both (28) and (29). The case of decreasing step sizes is treated in the following result, which is a direct consequence of Theorem 4.2.

Theorem 5.1. *Let Assumption A(i) and B hold. Let $\lambda \in \Lambda$ and let $w_t(\lambda)$ and $v_k(w,\lambda)$ be defined as in (28) and (29). Then, for every $w \in \mathbb{R}^d$, we have*

$$\lim_{t\to\infty} w_t(\lambda) = w(\lambda), \quad \lim_{k\to\infty} v_k(w,\lambda) = v(w,\lambda) \quad \mathbb{P}\text{-a.s.}$$

Moreover, let $\sigma_{\lambda,2} := 2(L^2_{\hat{\Phi},\lambda}+q^2_\lambda)/(1-q_\lambda)^2$ and $\eta_{\lambda,i} := \beta_\lambda/(\gamma_\lambda+i)$ with $\beta_\lambda > 1/(1-q^2_\lambda)$ and $\gamma_\lambda \geq \beta_\lambda(1+\sigma_{\lambda,2})$. Then for every $w \in \mathbb{R}^d$

$$\mathbb{E}[\|w_t(\lambda) - w(\lambda)\|^2] \leq \frac{d_{w,\lambda}}{\gamma_\lambda + t} \tag{30}$$

$$\mathbb{E}[\|v_k(w,\lambda) - v(w,\lambda)\|^2] \leq \frac{d_{v,\lambda}}{\gamma_\lambda + k} \tag{31}$$

where

$$d_{w,\lambda} := \max\left\{\gamma_\lambda\|w(\lambda)\|^2, \frac{\beta^2_\lambda\sigma_{\lambda,1}}{\beta_\lambda(1-q^2_\lambda)-1}\right\},$$
$$d_{v,\lambda} := \frac{\|\nabla_1 E(w,\lambda)\|^2}{(1-q_\lambda)^2}\max\left\{\gamma_\lambda, \frac{2\beta^2_\lambda L^2_{\hat{\Phi},\lambda}}{\beta_\lambda(1-q^2_\lambda)-1}\right\}$$
$$\sigma_{\lambda,1} := 2\mathbb{V}[\hat{\Phi}(w(\lambda),\lambda,\zeta)].$$

In a similar manner, using Theorem 4.1, one can have rates of convergence also using a constant stepsize, although in that case, we do not have asymptotic convergence of the iterates.

Remark 5.1. *For the setting considered in Theorem 5.1 the bound given in Theorem 3.4 yields*

$$MSE_{\hat{\nabla}f} \leq 2\frac{m_{2,\lambda}L^2_{E,\lambda}}{(1-q_\lambda)^2} + O\left(\frac{1}{\gamma_\lambda+t} + \frac{1}{\gamma_\lambda+k}\right). \tag{32}$$

where $MSE_{\hat{\nabla}f} = \mathbb{E}\big[\|\hat{\nabla}f(\lambda) - \nabla f(\lambda)\|^2\big]$

Crucially, typical bilevel problems in machine learning come in the form of (3), where the lower-level objective $\ell(w,\lambda) := \mathbb{E}[\hat{\ell}(w,\lambda,\zeta)]$ is Lipschitz smooth and strongly convex w.r.t. w. In this scenario, there is a vast amount of stochastic methods in literature (see e.g. Bottou et al. (2018) for a survey) achieving convergence rates in expectations of the kind provided in Theorem 5.1 or even better. For example, when $\ell(w,\lambda)$ has a finite sum structure, as in the case of the regularized empirical risk, exploiting variance reduction techniques makes the convergence rate $\rho_\lambda(t)$ linear. In this situation the following assumption is made.

Assumption G. *$\hat{\ell}$ is twice differentiable w.r.t. the first two arguments and such that, for every $w \in \mathbb{R}^d$, $\lambda \in \Lambda$, $j \in \{1,2\}$*

$$\mathbb{E}[\nabla_1\hat{\ell}(w,\lambda,\zeta)] = \nabla\ell(w,\lambda),$$
$$\mathbb{E}[\nabla^2_{1j}\hat{\ell}(w,\lambda,\zeta)] = \nabla^2_{1j}\ell(w,\lambda).$$

Moreover, for every $w \in \mathbb{R}^d$, $\lambda \in \Lambda$ and $x \in \mathcal{Z}$ there exists $L_\ell, m_\ell \geq 0$ such that:

$$\|\nabla^2_{11}\hat{\ell}(w,\lambda,\zeta)\| \leq L_\ell \quad \mathbb{V}[\nabla^2_{12}\hat{\ell}(w,\lambda,\zeta)] \leq m_\ell.$$

If $\hat{\ell}$ satisfies Assumption G, then $\hat{\Phi}(w,\lambda) = w - \alpha_\lambda\nabla_1\hat{\ell}(w,\lambda)$ satisfies Assumption B. In addition, since $I - \partial_1\Phi(w,\lambda) = \alpha_\lambda\nabla^2_1\ell(w,\lambda)$ is a positive definite matrix, we have that the solution to the linear system (5) can be written as

$$v(w,\lambda) = \arg\min_v g(v;w,\lambda)$$
$$g(v;w,\lambda) := \frac{\alpha_\lambda}{2}v^\top\nabla^2_1\ell(w,\lambda)v - v^\top\nabla_1 E(w,\lambda),$$

and $g(v;w,\lambda) = \mathbb{E}[\hat{g}(v;w,\lambda,\zeta)]$, where

$$\hat{g}(v;w,\lambda,\zeta) = \frac{\alpha_\lambda}{2}v^\top\nabla^2_1\hat{\ell}(w,\lambda,\zeta)v - v^\top\nabla_1 E(w,\lambda).$$

We can easily see that $g(\cdot;w,\lambda)$ is a strongly convex quadratic function with Lipschitz smooth constant and modulus of strong convexity at least as good as the ones of $\alpha_\lambda\ell(\cdot,\lambda)$. Thus, we can solve both subproblems in Algorithm 1 using the same stochastic optimization algorithm, achieving the same theoretical performance for both rates $\rho_\lambda(t)$ and $\sigma_\lambda(k)$.

We finally observe that the methods in (28)-(29) can be rewritten as

$$w_{t+1}(\lambda) := w_t(\lambda) - \eta_{\lambda,t}\alpha_\lambda\nabla_1\hat{\ell}(w_t(\lambda),\lambda,\zeta_t)$$
$$v_{k+1}(w,\lambda) := v_k(w,\lambda) - \eta_{\lambda,k}\nabla_1\hat{g}(v;w,\lambda,\hat{\zeta}_k)$$

which correspond to SGD on $\alpha_\lambda\ell(w,\lambda)$ and on $g(v;w,\lambda)$ respectively.

6 Experiments

In this section we present preliminary experiments evaluating the effectiveness of the SID method for estimating the hypergradient of f in a real data scenario.

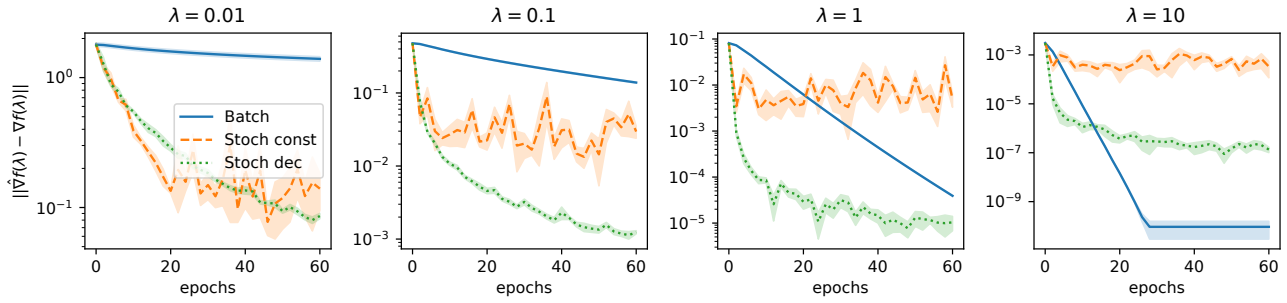

Figure 1: Experiment with a single regularization parameter. Convergence of three variants of SID for 4 choices of the regularization hyperparameter $\lambda \in \mathbb{R}_{++}$. Here, 2 epochs refer, in the Batch version, to one iteration on the lower-level problem plus one iteration on the linear system, whereas, in the Stochastic versions, they refer to 100 iterations on the lower-level problem plus 100 iterations on the linear system. The plot shows mean (solid lines) and std (shaded regions) over 5 runs, which vary the train/validation splits and, for the stochastic methods, the order and composition of the minibatches.

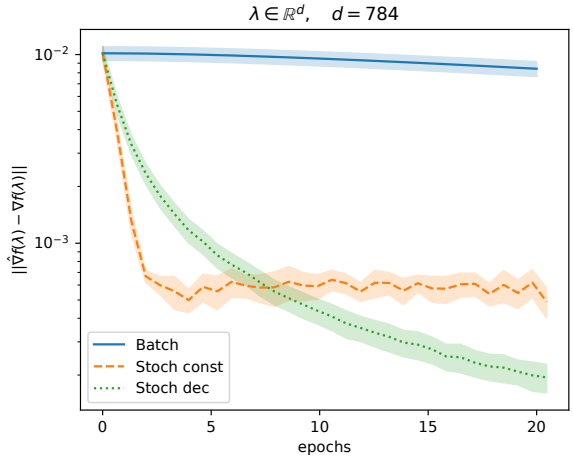

Figure 2: Experiment with multiple regularization parameters. Convergence of three variants of SID for several choices of the regularization hyperparameter $\lambda \in \mathbb{R}^d_{++}$. The plot shows mean (solid lines) and std (shaded regions) over 10 runs. For each run, $\lambda_i = e^{\epsilon_i}$, where $\epsilon_i \sim \mathcal{U}[-2, 2]$ for every $i \in \{1, \dots, d\}$. Epochs are defined as in Figure 1.

In Appendix C.1 we provide additional experiments on more realistc scenarios and with additional SID variants. We focus on a hyperparameter optimization problem where we want to optimize the regularization parameter(s) in regularized logistic regression. Specifically, we consider a binary classification problem with the aim to distinguish between odd and even numbers in the MNIST dataset. Referring to problem (3), we set

$$f(\lambda) = \sum_{i=n_{\text{tr}}+1}^{n_{\text{tr}}+n_{\text{val}}} \psi(y_i x_i^\top w(\lambda)),$$

$$w(\lambda) = \arg\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n_{\text{tr}}} \psi(y_i x_i^\top w) + R(w, \lambda),$$

where $\psi(u) = \log(1 + e^{-u})$ is the logistic loss, $(x_i, y_i)_{1 \le i \le n_{\text{tr}}+n_{\text{val}}} \in (\mathbb{R}^p \times \{0, 1\})^{n_{\text{tr}}+n_{\text{val}}}$ are training and validation examples, and $R(w, \lambda)$ is set according to either of the two situations below

- *one regularization parameter*:
  $R(w, \lambda) = \frac{\lambda}{2}\|w\|^2$, $\lambda \in \mathbb{R}_{++}$
- *multiple regularization parameters*:
  $R(w, \lambda) = \frac{1}{2} w^\top \text{diag}(\lambda) w$ where $\text{diag}(\lambda)$ is the diagonal matrix formed by the elements of $\lambda \in \mathbb{R}^d_{++}$.

We set $n_{\text{tr}} = n_{\text{val}} = 5000$, i.e., we pick 10000 examples from the MNIST training set and we group them into a training and a validation set of equal size. We set Φ to be the full gradient descent map on the lower-level objective, with optimal choice for the stepsize[3]. This map is a contraction because the lower objective is strongly convex and Lipschitz-smooth. We test the following three variants of SID (Algorithm 1), where we always solve the lower-level problem with t iterations of the procedure (28) and the linear system with $k = t$ iterations of the algorithm (29). However, we make different choices for $\eta_{\lambda,t}$ and the estimator $\hat{\Phi}$.

Batch. This variant of Algorithm 1 corresponds to the (deterministic) gradient descent algorithm with constant stepsize. We set $t_{\text{Batch}} = k_{\text{Batch}} = 30$ and, for every $t = 0, \dots, t_{\text{Batch}}$, $\eta_{\lambda,t} = \eta_{\lambda,k} = 1$ and $\hat{\Phi}(w, \lambda, \zeta) = \Phi(w, \lambda)$.

Stoch const. For this variant, $\hat{\Phi}(w, \lambda, \zeta)$ corresponds to one step of stochastic gradient descent on a randomly sampled minibatch of 50 examples. Thus, $t_{\text{Stoch const}} = t_{\text{Batch}} \times 100$, $k_{\text{Stoch const}} = k_{\text{Batch}} \times 100$, and we pick $\eta_{\lambda,t} = \eta_{\lambda,k} = 1$, for $t = 0, \dots, t_{\text{Stoch const}}$.

[3] We set the stepsize equal to two divided by the sum of the Lipschitz and strong convexity constants of the lower-level objective. This gives the optimal contraction rate q_λ.

Stoch dec. For this variant the estimator $\hat{\Phi}$ is the same as for the *Stoch const* strategy, but we use decreasing stepsizes. More precisely, $\eta_{\lambda,t} = \eta_{\lambda,k} = \beta_\lambda/(\gamma_\lambda + t)$ with $\beta_\lambda = 2/(1-q_\lambda^2)$ and $\gamma_\lambda = \beta_\lambda$. Moreover, as before, $t_{\text{Stoch dec}} = t_{\text{Batch}} \times 100$, $k_{\text{Stoch dec}} = k_{\text{Batch}} \times 100$.

We note that the *Batch* strategy is exactly the *fixed point method* described by Grazzi et al. (2020), which converges linearly to the true hypergradient. Moreover, for the stochastic versions, we can write $\hat{\Phi}(w, \lambda, \zeta) = \Phi_1(w, \zeta) + \Phi_2(w, \lambda)$, so that we are in the case discussed in Remark 4.2 and hence, $\sigma_2 = m_{2,\lambda} = 0$. In this situation, it follows from Remark 5.1 that the *Stoch dec* version of Algorithm 1 converges in expectation to the true hypergradient with a rate $O(1/(\gamma_\lambda + t))$, whereas, according to Corollary 4.1 and Theorem 3.4, the *Stoch const* version can possibly approach the true hypergradient in a first phase (at linear rate), but ultimately might not converge to it.

In Figures 1 and 2 we show the squared error between the approximate and the true hypergradient ($\hat{\nabla} f(\lambda)$ and $\nabla f(\lambda)$ respectively) for the two regularization choices described above[4]. In both figures we can see the effectiveness of the proposed SID method (and especially the *Stoch dec* variant) against its deterministic version (AID) previously studied in (Grazzi et al., 2020).

7 Conclusions and Future Work

In this paper we studied a stochastic method for the approximation of the hypergradient in bilevel problems defined through a fixed-point equation of a contraction mapping. Specifically, we presented a stochastic version of the approximate implicit differentiation technique (AID), which is one of the most effective solutions for hypergradient computation as recently shown in (Grazzi et al., 2020). Our strategy (SID) estimates the hypergradient with the aid of two stochastic solvers in place of the deterministic solvers used in AID. We presented a formal description and a theoretical analysis of SID, ultimately providing a bound for the mean square error of the corresponding hypergradient estimator. As a byproduct of the analysis, we provided an extension of the SGD algorithm for stochastic fixed-point equations. We have also conducted numerical experiments which confirm that using stochastic instead of deterministic solvers in SID can indeed yield a more accurate hypergradient approximation.

[4]Since for regularized logistic regression, the hypergradient is not available in closed form, we compute it by using the Batch version with $t = k = 2000$ (4000 epochs in total).

We believe that our analysis of stochastic fixed-point algorithms can be further extended to include variance reduction strategies and other advances commonly used for SGD. A good starting point for this extension can be the work by Gorbunov et al. (2020), which provides a unified theory for SGD methods in the strongly convex setting. Another promising direction would be the analysis of an overall bilevel optimization procedure using SID to approximate the hypergradient, which we have not addressed in the present work.

References

Ablin, P., Peyré, G., and Moreau, T. (2020). Super-efficiency of automatic differentiation for functions defined as a minimum. *arXiv preprint arXiv:2002.03722*.

Almeida, L. B. (1987). A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In *Proceedings, 1st First International Conference on Neural Networks*, volume 2, pages 609–618. IEEE.

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989.

Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311.

Couellan, N. and Wang, W. (2016). On the convergence of stochastic bi-level gradient methods. *Optimization*.

Denevi, G., Ciliberto, C., Grazzi, R., and Pontil, M. (2019a). Learning-to-learn stochastic gradient descent with biased regularization. *arXiv preprint arXiv:1903.10399*.

Denevi, G., Stamos, D., Ciliberto, C., and Pontil, M. (2019b). Online-within-online meta-learning. In *Advances in Neural Information Processing Systems*, pages 13110–13120.

Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. (2017). Forward and reverse gradient-based hyperparameter optimization. In *Proceedings*

of the 34th International Conference on Machine Learning-Volume 70, pages 1165–1173. JMLR. org.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. (2018). Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1563–1572.

Ghadimi, S. and Wang, M. (2018). Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*.

Gorbunov, E., Hanzely, F., and Richtárik, P. (2020). A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 680–690. PMLR.

Grazzi, R., Franceschi, L., Pontil, M., and Salzo, S. (2020). On the iteration complexity of hypergradient computation. *arXiv preprint arXiv:2006.16218*.

Liu, H., Simonyan, K., and Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Lorraine, J., Vicol, P., and Duvenaud, D. (2019). Optimizing millions of hyperparameters by implicit differentiation. *arXiv preprint arXiv:1911.02590*.

Maclaurin, D., Duvenaud, D., and Adams, R. (2015). Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pages 2113–2122.

Nguyen, L. M., Nguyen, P. H., Richtárik, P., Scheinberg, K., Takáč, M., and van Dijk, M. (2019). New convergence aspects of stochastic gradient algorithms. *Journal of Machine Learning Research*, 20(176):1–49.

Pedregosa, F. (2016). Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning*, pages 737–746.

Pineda, F. J. (1987). Generalization of backpropagation to recurrent neural networks. *Physical review letters*, 59(19):2229.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. (2019). Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pages 113–124.

Robbins, H. and Siegmund, D. (1971). A convergence theorem for non negative almost supermartingales and some applications. *Optimizing Methods in Statistics*, pages 233–257.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Zhou, P., Yuan, X., Xu, H., Yan, S., and Feng, J. (2019). Efficient meta learning via minibatch proximal update. In *Advances in Neural Information Processing Systems*, pages 1534–1544.