Anytime Online-to-Batch, Optimism and Acceleration

Ashok Cutkosky [1]

Abstract

A standard way to obtain convergence guarantees in stochastic convex optimization is to run an online learning algorithm and then output the average of its iterates: the actual iterates of the online learning algorithm do not come with individual guarantees. We close this gap by introducing a black-box modification to any online learning algorithm whose iterates converge to the optimum in stochastic scenarios. We then consider the case of smooth losses, and show that combining our approach with optimistic online learning algorithms immediately yields a fast convergence rate of $O(L/T^{3/2} + \sigma/\sqrt{T})$ on L-smooth problems with σ^2 variance in the gradients. Finally, we provide a reduction that converts any adaptive online algorithm into one that obtains the optimal accelerated rate of $\tilde{O}(L/T^2 + \sigma/\sqrt{T})$, while still maintaining $\tilde{O}(1/\sqrt{T})$ convergence in the non-smooth setting. Importantly, our algorithms adapt to L and σ automatically: they do not need to know either to obtain these rates.

1. Online-to-Batch Conversions

We consider convex stochastic optimization problems, where our objective is to minimize some convex function $\mathcal{L} : D \to \mathbb{R}$ where D is some convex domain. We do not have true access to $\mathcal{L}$, however. Instead, we have a stochastic gradient oracle that given a point $x \in D$ will provide a random value g such that $\mathbb{E}[g] = \nabla\mathcal{L}(x)$. Our objective is to use this noisy information to optimize $\mathcal{L}$.

A simple and extremely effective method for solving stochastic optimization problems is through online learning and online-to-batch conversion (Shalev-Shwartz, 2011; Cesa-Bianchi et al., 2004). These techniques require remarkably few assumptions about the nature of the expected loss or the stochasticity in the system and yet still obtain optimal or near-optimal guarantees. This has helped fuel the widespread adoption of online learning algorithms as the method-of-choice in training machine learning models. Briefly, an online learning algorithm accepts a sequence of convex loss functions $\ell_1, \dots, \ell_T$ and outputs a sequence of iterates $w_1, \dots, w_T \in D$ where D is some convex space and w_t is output *before* the algorithm observes ℓ_t. Performance is measured by the regret:

$$R_T(x^\star) = \sum_{t=1}^{T} \ell_t(w_t) - \ell_t(x^\star)$$

A standard goal in online learning is to achieve *sublinear regret*, which means that $\lim_{T\to\infty} R_T(x^\star)/T = 0$. This indicates that the algorithm is doing just as well "on average" as the fixed benchmark point $x^\star$. In fact, most algorithms obtain non-asymptotic guarantees of the form $R_T(x^\star) = O(\sqrt{T})$, so that $R_T(x^\star)/T = O(1/\sqrt{T})$.

Online learning algorithms often adopt an adversarial model, in which no relationship is posited between ℓ_t, but in our stochastic optimization problem we know that the ℓ_t are generated by some random process. This is where the Online-to-Batch conversion technique comes in (Cesa-Bianchi et al., 2004). The classic argument is as follows: Set $\ell_t(x) = \langle g_t, x\rangle$ where g_t is a stochastic gradient evaluated at w_t. Then observe $\mathcal{L}(w_t) - \mathcal{L}(x^\star) \le \mathbb{E}[\langle g_t, w_t - x^\star\rangle]$ and apply Jensen's inequality to obtain:

$$\mathbb{E}\left[\mathcal{L}\left(\frac{\sum_{t=1}^T w_t}{T}\right) - \mathcal{L}(x^\star)\right] \le \frac{\mathbb{E}[R_T(x^\star)]}{T}$$

We therefore output $\hat{x} = \frac{\sum_{t=1}^T w_t}{T}$ as an estimate of $x^\star$, and so long as the algorithm obtains sublinear regret, $\mathcal{L}(\hat{x}) - \mathcal{L}(x^\star)$ will approach zero in expectation. In fact, with $R_T(x^\star) = O(\sqrt{T})$, one obtains a convergence rate $O(1/\sqrt{T})$, which is often statistically optimal.

One drawback of the online-to-batch conversion is that the iterates w_t produced by the algorithm (where the noisy gradients are actually evaluated) do not necessarily converge to the optimal loss value. In fact, there is typically very little known about the behavior of any individual w_t. This is aesthetically unsatisfying and may even reduce performance. For example, *optimistic* online algorithms can take advantage of stability in the gradients, performing well when

[1]Google Research, California, USA. Correspondence to: Ashok Cutkosky <ashok@cutkosky.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

$g_{t-1} \approx g_t$. We hope for this behavior because intuitively the iterates should converge to $x^\star$ and so become closer together. Unfortunately, because actually we usually have few guarantees about the individual iterates w_t, it may not hold that $g_{t-1} \approx g_t$. We would like to make intuition match theory by enforcing some kind of stability in the iterates.

We address this problem by providing a black-box online-to-batch conversion: the iterates x_t produced by our algorithm converge in the sense that $\mathcal{L}(x_t) \to \mathcal{L}(x^\star)$ (Section 2). We call this property *anytime*, because the last iterate is always a good estimate of $x^\star$ at any time. Our reduction is quite simple, and bears strong similarity to the classical one. It stabilizes the iterates x_t, and we can exploit this stability when $\mathcal{L}$ is smooth. For example, when applied to an optimistic online algorithm, our reduction can leverage stability to improve the convergence rate on smooth losses from $O(L/T)$ to $O(L/T^{3/2})$ (Section 4.1). Further, our reduction also has a surprising connection to the linear coupling framework for *accelerated* algorithms (Allen-Zhu & Orecchia, 2014). We develop this connection to provide an algorithm that obtains a near-optimal (up to log factors) $\tilde{O}(L/T^2 + \sigma/\sqrt{T})$ convergence rate for stochastic smooth losses with $\sigma^2 = \mathrm{Var}(g_t)$ without knowledge of L or σ while still guaranteeing $\tilde{O}(1/\sqrt{T})$ convergence rate for non-smooth losses (Section 4.2). In addition to these new algorithms, we feel that our analysis itself is interesting for its appealingly simplicity.

1.1. Notation and Definitions

We frequently use the compressed-sum notation $\alpha_{1:t} = \sum_{i=1}^t \alpha_i$ for any indexed variables α_t. A convex function f is L-smooth if $f(x+\delta) \le f(x) + \langle \nabla f(x), \delta\rangle + \frac{L}{2}\|\delta\|^2$ for and x, δ, and f is μ strongly convex if $f(x+\delta) \ge f(x) + \langle \nabla f(x), \delta\rangle + \frac{\mu}{2}\|\delta\|^2$ for all x, δ. Given a convex function f we say that g is a subgradient of f at x, or $g \in \partial f(x)$ if $f(y) \ge f(x) + \langle g, y - x\rangle$ for all y. $\nabla f(x) \in \partial f(x)$ if f is differentiable.

2. Anytime Online-to-Batch

In this section we provide our anytime online-to-batch conversion. Our algorithm is actually nearly identical to the classic online to batch: we set the tth iterate x_t to be the average of the first t iterates of some online learning algorithm $\mathcal{A}$. The key difference is that we evaluate the stochastic gradient oracle at x_t, rather than the iterates provided by $\mathcal{A}$. As a result, the outputs of $\mathcal{A}$ in some sense exist only for analysis and are not directly visible outside the algorithm. Further, we incorporate *weights* α_t into our conversion. Inspired by (Levy, 2017), these weights play a role in achieving faster rates on smooth losses, as well as removing log factors on strongly-convex losses. We provide specific pseudocode and analysis in Algorithm 1 and Theorem 1 below.

Algorithm 1 Anytime Online-to-Batch

Input: Online learning algorithms $\mathcal{A}$ with convex domain D. Non-negative weights $\alpha_1, \dots, \alpha_T$ with $\alpha_1 > 0$.
Get initial point $w_1 \in D$ from $\mathcal{A}$.
for $t = 1$ **to** T **do**
  $x_t \leftarrow \frac{\sum_{i=1}^t \alpha_i w_t}{\alpha_{1:t}}$.
  Play x_t, receive subgradient g_t.
  Send $\ell_t(x) = \langle \alpha_t g_t, x\rangle$ to $\mathcal{A}$ as the tth loss.
  Get w_{t+1} from $\mathcal{A}$.
end for
return x_T.

Theorem 1. *Suppose $g_1, \dots, g_t$ satisfy $\mathbb{E}[g_t|x_t] \in \partial\mathcal{L}(x_t)$ for some function $\mathcal{L}$ and g_t is independent of all other quantities given x_t. Let $R_T(x^\star)$ be a bound on the linearized regret of $\mathcal{A}$:*

$$R_T(x^\star) \ge \sum_{t=1}^T \langle \alpha_t g_t, w_t - x^\star\rangle$$

Then for all $x^\star \in D$, Algorithm 1 guarantees:

$$\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \le \mathbb{E}\left[\frac{R_T(x^\star)}{\sum_{t=1}^T a_t}\right]$$

Further, suppose that D has diameter $B = \sup_{x,y\in D}\|x - y\|$ and $\|g_t\|_\star \le G$ with probability 1 for some G. Then with probability at least $1-\delta$,

$$\mathcal{L}(x_T) - \mathcal{L}(x^\star) \le \frac{R_T(x^\star) + 2BG\sqrt{\sum_{t=1}^T \alpha_t^2 \log(2/\delta)}}{\sum_{t=1}^T \alpha_t}$$

Proof. First, observe that

$$\alpha_t(x_t - w_t) = \alpha_{1:t-1}(x_{t-1} - x_t)$$

where by mild abuse of notation we define $\alpha_{1:0} = 0$ and let x_0 be an arbitrary element of D.

Now we use the standard convexity argument to say:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^T \alpha_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star))\right] &\le \mathbb{E}\left[\sum_{t=1}^T \alpha_t\langle g_t, x_t - x^\star\rangle\right] \\
&= \mathbb{E}\left[\sum_{t=1}^T \alpha_t\langle g_t, x_t - w_t\rangle + \alpha_t\langle g_t, w_t - x^\star\rangle\right] \\
&\le \mathbb{E}\left[R_T(x^\star)\right] + \mathbb{E}\left[\sum_{t=1}^T \alpha_{1:t-1}\langle g_t, x_{t-1} - x_t\rangle\right]
\end{aligned}$$

Next we use convexity again to argue $\mathbb{E}[\langle g_t, x_{t-1} - x_t\rangle] \le \mathbb{E}[\mathcal{L}(x_{t-1}) - \mathcal{L}(x_t)]$, and then we subtract

$\mathbb{E}[\sum_{t=1}^T \alpha_t \mathcal{L}(x_t)]$ from both sides:

$$\mathbb{E}\left[\sum_{t=1}^T \alpha_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star))\right]$$
$$\leq \mathbb{E}\left[R_T(x^\star)\right] + \mathbb{E}\left[\sum_{t=1}^T \alpha_{1:t-1}(\mathcal{L}(x_{t-1}) - \mathcal{L}(x_t))\right]$$
$$\mathbb{E}\left[-\alpha_{1:T}\mathcal{L}(x^\star)\right]$$
$$\leq \mathbb{E}\left[R_T(x^\star)\right] + \mathbb{E}\left[\sum_{t=1}^T \alpha_{1:t-1}\mathcal{L}(x_{t-1}) - \alpha_{1:t}\mathcal{L}(x_t)\right]$$

Finally, telescope the above sum to conclude:

$$\mathbb{E}\left[\alpha_{1:T}\mathcal{L}(x_T) - \alpha_{1:T}\mathcal{L}(x^\star)\right] \leq \mathbb{E}\left[R_T(x^\star)\right]$$

from which the in-expectation statement of the Theorem follows.

For the high-probability statement, let H_{t-1} be the history $g_{t-1}, x_{t-1}, \dots, g_1, x_1$. Let $G_t = \mathbb{E}[g_t | H_{t-1}, x_t, w_t]$. Note that G_t is still a random variable, and satisfies $G_t \in \partial \mathcal{L}(x_t)$. Next, let $\epsilon_t = \alpha_t\langle G_t, w_t - x^\star\rangle - \alpha_t\langle g_t, w_t - x^\star\rangle$. Then we have $\mathbb{E}[\epsilon_t | H_{t-1}, x_t, w_t] = 0$ and:

$$\sum_{t=1}^T \epsilon_t = \sum_{t=1}^T \alpha_t\langle G_t, x_t - x^\star\rangle - \sum_{t=1}^T \alpha_t\langle g_t, x_t - x^\star\rangle$$
$$|\epsilon_t| \leq 2\alpha_t BG \text{ with probability } 1$$

So by the Azuma-Hoeffding bound, with probability at least $1 - \delta$:

$$\sum_{t=1}^T \epsilon_t \leq 2BG\sqrt{\sum_{t=1}^T \alpha_t^2 \log(2/\delta)}$$

Therefore with probability at least $1 - \delta$, we have

$$\sum_{t=1}^T \alpha_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star)) \leq \sum_{t=1}^T \alpha_t\langle G_t, x_t - x^\star\rangle$$
$$\leq \sum_{t=1}^T \alpha_t\langle G_t, x_t - w_t\rangle + \sum_{t=1}^T \alpha_t\langle g_t, w_t - x^\star\rangle + \sum_{t=1}^T \epsilon_t$$
$$\leq \sum_{t=1}^T \alpha_t\langle G_t, x_t - w_t\rangle + R_T(x^\star) + 2BG\sqrt{\sum_{t=1}^T \alpha_t^2 \log\left(\frac{2}{\delta}\right)}$$

Now an identical argument to the in-expectation part of the Theorem (but without need for taking expectations) yields:

$$\mathcal{L}(x_T) - \mathcal{L}(x^\star) \leq \frac{R_T(x^\star) + 2BG\sqrt{\sum_{t=1}^T \alpha_t^2 \log(2/\delta)}}{\sum_{t=1}^T \alpha_t}$$

□

As a corollary, we observe that the simple setting of $\alpha_t = 1$ for all T yields a direct analog of the classic online-to-batch conversion guarantee:

Corollary 1. *Under the assumptions of Theorem 1, set $\alpha_t = 1$ for all t. Then $R_T(x^\star) = \sum_{t=1}^T \langle g_t, w_t - x^\star\rangle$, which is the usual un-weighted regret. We have*

$$\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \leq \mathbb{E}\left[\frac{R_T(x^\star)}{T}\right]$$

Further, $x_T = \frac{1}{T}\sum_{t=1}^T w_t$.

Corollary 1 is quite similar to the classic online-to-batch conversion result: in both cases, the average of the online learner's predictions has excess loss bounded by the average regret. Again, the critical difference is that in Algorithm 1, the actual outputs where the gradients are evaluated are the averaged outputs of the online learner. Thus the loss of the iterates converges to the minimum loss for Algorithm 1, which is not the case for the standard reduction.

In addition to this anytime online-to-batch result, we show below that Algorithm 1 also maintains low regret:

Corollary 2. *Under the assumptions of Theorem 1, let $R^M(x^\star) \geq \max_t R_t(x^\star)$. Then we have*

$$\mathbb{E}\left[\sum_{t=1}^T \alpha_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star))\right]$$
$$\leq \mathbb{E}\left[R^M(x^\star)\left(1 + \log\left(\frac{\alpha_{1:T}}{\alpha_1}\right)\right)\right]$$

Proof. From Theorem 1 we have

$$\mathbb{E}[\alpha_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star))] \leq \mathbb{E}\left[\frac{\alpha_t R_t(x^\star)}{\alpha_{1:t}}\right] \leq \mathbb{E}\left[\frac{\alpha_t R_t^M(x^\star)}{\alpha_{1:t}}\right]$$

Then observe that $\log(a) + b/(a+b) \leq \log(a+b)$ and sum over t to conclude the Corollary. □

Recall that essentially all online learning regret bounds are non-decreasing in T, so that $\max_t R_t(x^\star) = R_T(x^\star)$. Thus the regret of Algorithm 1 is only a logarithmic factor worse than the regret of the original online learner. Moreover, in the typical case that $R_t(x^\star) = O(\sqrt{t})$, a trivial modification of the above proof shows that $\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \leq O(1/\sqrt{T})$, so that in many cases one should not even incur the log factor.

In fact, the anytime result is significantly more powerful than a standard regret bound because it provides point-wise bounds. This allows us to achieve a variety of different weighted regret bounds *simultaneously*:

Corollary 3. *Under the assumptions of Theorem 1, further suppose that $R_T(x^\star)$ is non-decreasing in T and set $\alpha_t = 1$.*

Let $s_t = t^k$ for some constants $k > 0$ (note that Algorithm 1 is not aware of s_t). Then

$$\mathbb{E}\left[\frac{\sum_{t=1}^T s_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star))}{s_{1:T}}\right] \le O\left(\frac{R_T(x^\star)}{T}\right)$$

Proof. Observe $s_{1:t} = \Theta(t^{k+1})$ so that $\mathbb{E}[s_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star))/s_{1:T}] \le O(\mathbb{E}[R_T(x^\star)]t^{k-1}/T^{k+1})$, and sum over t. □

3. General Analysis

In this section we provide a more general version of our online-to-batch reduction. The previous analysis appears to critically rely on linearized regret $\mathbb{E}[\sum_{t=1}^T \alpha_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star))] \le \mathbb{E}[\sum_{t=1}^T \alpha_t \langle g_t, x_t - x^\star\rangle]$. This inequality may be tight for general convex losses, but in many cases we may want to take advantage of some known non-linearity in the losses. For example, when the loss function is μ-strongly convex, one can use the inequality $\mathcal{L}(x_t) - \mathcal{L}(x^\star) \le \ell_t(x_t) - \ell_t(x^\star)$ where $\ell_t(x) = \langle \nabla \mathcal{L}, x\rangle + \frac{\mu}{2}\|x - x_t\|^2$, leading to a $O(\log(T)/T)$ convergence rate rather than $O(1/\sqrt{T})$ (Hazan et al., 2007). In order to incorporate this information in our framework, we propose Algorithm 2.

Algorithm 2 modifies Algorithm 1 by considering an oracle that produces losses ℓ_t rather than stochastic gradients g_t. Specifically, we will require ℓ_t that are convex and lower-bound $\mathcal{L}$ in expectation. This generalizes the linear losses of Algorithm 1, and it may often be possible to construct nonlinear ℓ_t via only a gradient oracle, such as in the strongly-convex case. Our strategy for using these losses is essentially unchanged from that of Algorithm 1, but now our analysis is slightly more delicate since we cannot exploit the nice algebraic properties of linearity.

Algorithm 2 General Anytime Online-to-Batch

Input: Online learning algorithms $\mathcal{A}$ with convex domain D. Non-negative weights $\alpha_1, \dots, \alpha_T$ with $\alpha_1 > 0$
Get initial point $w_1 \in D$ from $\mathcal{A}$.
for $t = 1$ **to** T **do**
  $x_t \leftarrow \frac{\sum_{i=1}^t \alpha_i w_i}{\alpha_{1:t}}$
  Play x_t, compute loss ℓ_t.
  Send $\alpha_t \ell_t(x)$ to $\mathcal{A}$ as the tth loss.
  Get w_{t+1} from $\mathcal{A}$.
end for
return x_T.

Theorem 2. *Suppose ℓ_t is convex and satisfies $\mathcal{L}(x_t) - \mathcal{L}(x) \le \mathbb{E}[\ell_t(x_t) - \ell_t(x)|x_t]$ for all t and for all x. Then with*

$$R_T(x^\star) = \sum_{t=1}^T \alpha_t \ell_t(w_t) - \alpha_t \ell_t(x_t^\star)$$

Algorithm 2 obtains

$$\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \le \mathbb{E}\left[\frac{R_T(x^\star)}{\sum_{t=1}^T \alpha_t}\right]$$

Proof.

$$\begin{aligned}
&\sum_{t=1}^T \alpha_t(\ell_t(x_t) - \ell_t(x_t^\star)) \\
&\le \sum_{t=1}^T \alpha_t(\ell_t(x_t) - \ell_t(w_t)) + \sum_{t=1}^T \alpha_t(\ell_t(w_t) - \ell_t(x_t^\star)) \\
&= R_T(x^\star) + \sum_{t=1}^T \alpha_t(\ell_t(x_t) - \ell_t(w_t)) \qquad (1)
\end{aligned}$$

Now observe that $x_t = \frac{\alpha_{1:t-1}x_{t-1} + \alpha_t w_t}{\alpha_{1:t-1}}$. Therefore by Jensen's inequality we have

$$\ell_t(x_t) \le \frac{\alpha_{1:t-1}\ell_t(x_{t-1}) + \alpha_t \ell_t(w_t)}{\alpha_{1:t}}$$
$$\alpha_t \ell_t(x_t) - \alpha_t \ell_t(w_t) \le \alpha_{1:t-1}(\ell_t(x_{t-1}) - \ell_t(x_t))$$

Now plug this into (1):

$$\begin{aligned}
&\sum_{t=1}^T \alpha_t(\ell_t(x_t) - \ell_t(x^\star)) \\
&\le R_T(x^\star) + \sum_{t=1}^T \alpha_{1:t-1}\ell_t(x_{t-1}) - \alpha_{1:t-1}\ell_t(x_t)
\end{aligned}$$

Now observe that $\mathbb{E}[\ell_t(x_{t-1}) - \ell_t(x_t)] \le \mathbb{E}[\mathcal{L}(x_{t-1}) - \mathcal{L}(x_t)]$. So taking expectations yields:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T \alpha_t(\mathcal{L}(x_t) - \mathcal{L}(x_t^\star))\right] \\
&\le \mathbb{E}\left[R_T(x^\star) + \sum_{t=1}^T \alpha_{1:t-1}(\mathcal{L}(x_{t-1}) - \mathcal{L}(x_t))\right]
\end{aligned}$$

Now the rest of the proof is identical to that of Theorem 1. □

3.1. Strongly Convex losses

In this section we apply the more general Algorithm 2 to μ-strongly-convex losses. We recover standard convergence rates using only a gradient oracle and knowledge of the strong-convexity parameter μ. We note that similar results also apply to exp-concave losses or other cases with lower-bounded Hessians.

Corollary 4. *Suppose D has diameter B, $\|g_t\| \le G$ with probability 1, and $\mathcal{A}$ is Follow-the-Leader: $w_{t+1} = \text{argmin} \sum_{i=1}^t \ell_i(w)$. Suppose $\mathcal{L}$ is μ-strongly convex and we set $\ell_t(x) = \langle g_t, x\rangle + \frac{\mu}{2}\|x - x_t\|^2$ where $\mathbb{E}[g_t|x_t] = \nabla\mathcal{L}(x_t)$. Let $\alpha_t = 1$ for all t. Then we have*

$$R_T(x^\star) \le \frac{(\mu B + G)^2(\log(T) + 1)}{2\mu}$$

and

$$\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \le \frac{(\mu B + G)^2(\log(T) + 1)}{2\mu T}$$

Proof. The fact that $\mathcal{L}(x_t) - \mathcal{L}(x^\star) \le \mathbb{E}[\ell_t(x_t) - \ell_t(x^\star)|x_t]$ follows from strong-convexity. Observe that $\|\nabla\ell_t(w_t)\| = \|g_t + \mu(w_t - x_t)\| \le G + \mu B$ so that ℓ_t is $G + \mu B$-Lipschitz. Then the bound on R_T follows from standard analysis of the follow-the-leader algorithm using the fact that $\sum_{i=1}^t \ell_i(w)$ is $t\mu$-strongly convex (McMahan, 2014):

$$R_T(x^\star) \le \sum_{t=1}^T \frac{\|\nabla\ell_t(w_t)\|^2}{2t\mu}$$

and then use $\sum_{i=1}^t 1/i \le \log(T) + 1$. □

This corollary provides the anytime analog of the standard online-to-batch result for strongly-convex losses. However, it is well-known that in the stochastic case the logarithmic factor is unnecessary. Prior work has removed it via diverse mechanisms, including restarting schemes (Hazan & Kale, 2014) and tail-averaging (Rakhlin et al., 2012). Here we use the weights α_t to easily remove the log factor, similar to the analogous scheme for the classic online-to-batch conversion (Lacoste-Julien et al., 2012; Bubeck et al., 2015).

Corollary 5. *Under the assumptions of Corollary 4, suppose that $\alpha_t = t$ for all t. Then we have*

$$R_T(x^\star) \le \frac{T(\mu B + G)^2}{\mu}$$

and

$$\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \le \frac{2(\mu B + G)^2}{\mu(T + 1)}$$

Proof. In this case, $\alpha_t\ell_t$ is $t(\mu B + G)$-Lipschitz and $\sum_{i=1}^t \alpha_i\ell_i$ is $\alpha_{1:t}\mu = \frac{T(T+1)\mu}{2}$ strongly convex. Thus the regret of Follow-the-Leader is bounded by

$$\begin{aligned} R_T(x^\star) &\le \sum_{t=1}^T \frac{\|t\nabla\ell_t(w_t)\|^2}{t(t+1)\mu} \\ &\le T\frac{(\mu B + G)^2}{\mu} \end{aligned}$$

Now divide by $\alpha_{1:T} = T(T + 1)/2$ to see the claim. □

4. Adaptivity and Smoothness

Many so-called "adaptive" online algorithms obtain regret bounds of the form $R_T(x^\star) \le O\left(\psi(x^\star)\sqrt{\sum_{t=1}^T \|g_t\|^2}\right)$ for various functions ψ. For example, Mirror-Descent and FTRL-based algorithms often obtain $\psi(x^\star) = B$, where B is the diameter of the space D (McMahan & Streeter, 2010; Duchi et al., 2010; Hazan et al., 2008) while so-called "parameter-free" algorithms can obtain $\psi(x^\star) = \tilde{O}(\|x^\star\|)$, providing optimal adaptivity to $\|x^\star\|$ at the expense of logarithmic factors (Cutkosky & Orabona, 2018). These adaptive bounds can be shown to obtain the better regret guarantee $\mathbb{E}\left[\sum_{t=1}^T \mathcal{L}(w_t) - \mathcal{L}(x^\star)\right] \le O\left(L\psi(x^\star)^2 + \psi(x^\star)\sigma\sqrt{T}\right)$ when the loss $\mathcal{L}$ is L-smooth and g_t has variance σ, by exploiting the self-bounding property $\|\nabla\mathcal{L}(x)\|^2 \le L(\mathcal{L}(x) - \mathcal{L}(x^\star))$ (Srebro et al., 2010; Cutkosky & Busa-Fekete, 2018; Levy et al., 2018).

The appealing property of this argument is that the algorithm knows neither L nor σ and yet automatically adapts to both parameters, matching the performance of an optimally-tuned SGD algorithm. Since Algorithm 1 also obtains low regret, we can make a similar claim:

Corollary 6. *Suppose $R_T(x^\star) \le \psi(x^\star)\sqrt{\sum_{t=1}^T \alpha_t^2\|g_t\|^2}$. Suppose $\mathcal{L}$ is L-smooth and obtains its minimum at $x^\star \in D$. Suppose g_t has variance at most σ^2. Then with $\alpha_t = 1$ for all t, Algorithm 1 obtains:*

$$\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \le O\left(\frac{\psi(x^\star)^2 L\log^2(T)}{T} + \frac{\sigma\log(T)}{\sqrt{T}}\right)$$

Proof. Define $\Delta_t = \mathbb{E}[\mathcal{L}(x_t) - \mathcal{L}(x^\star)]$. Observe that

$$\mathbb{E}[\|\nabla g_t\|^2] \le \mathbb{E}[\|\nabla\mathcal{L}(x_t)\|^2] + \sigma^2 \le L\Delta_t + \sigma^2$$

$$\mathbb{E}[R_T(x^\star)] \le \psi(x^\star)\sqrt{L\sum_{t=1}^T \Delta_t + T\sigma^2}$$

Then apply Corollary 2 and quadratic formula to obtain $\sum_{t=1}^T \Delta_t \le O\left(\psi(x^\star)^2 L\log^2(T) + \sigma\log(T)\sqrt{T}\right)$ when $\alpha_t = 1$ and observe $\Delta_T \le \mathbb{E}[R_T(x^\star)]/T$ to prove the Corollary. □

The assumption that $x^\star \in D$ and the log factors in this analysis are a bit troubling. In the next section we exploit optimism instead of the self-bounding property, which yields much better results with much less effort.

4.1. Optimism for Faster Rates

In this section we show how to leverage our online-to-batch scheme in combination with *optimistic* online learning to further speed up the convergence rate. We will achieve a rate of $O(L/T^{3/2} + \sigma/\sqrt{T})$ with no knowledge of either L or σ, resulting in a kind of interpolation between the $O(L/T + \sigma/\sqrt{T})$ rate and the optimal accelerated rate of $O(L/T^2 + \sigma/\sqrt{T})$ (Lan, 2012).

An optimistic online learning algorithm is an online learner that is given access to a series of "hints" $\hat{g}_1, \dots, \hat{g}_T$ where $\hat{g}_t$ is revealed to the learner after g_{t-1} but *before* it commits to w_t (Hazan & Kale, 2010; Rakhlin & Sridharan, 2013; Chiang et al., 2012; Mohri & Yang, 2016). Optimistic algorithms attempt to guarantee small regret when $\hat{g}_t \approx g_t$, because in this scenario the learner has a good guess for what the future will contain. In particular, the optimistic algorithm of (Mohri & Yang, 2016) guarantees regret:

$$R_T(x^\star) \le B\sqrt{2\sum_{t=1}^T \alpha_t^2 \|\hat{g}_t - g_t\|^2}$$

where B is the diameter of the D. A common choice for $\hat{g}_t$ is g_{t-1}. Intuitively, this choice is "optimistic" in the sense that we are hoping $g_{t-1} \approx g_t$, which is the case on smooth losses if the iterates are close together. Fortunately, it *is* the case that x_t is necessarily close to x_{t-1}, so we use this regret bound for faster convergence in Algorithm 3 and Theorem 3.

Algorithm 3 Optimistic Anytime Online-to-Batch

Input: Optimistic Online algorithm $\mathcal{A}$ with domain D.
Non-negative weights $\alpha_1, \dots, \alpha_T$ with $\alpha_1 > 0$.
Get initial point $w_1 \in D$ from $\mathcal{A}$.
Set $g_0 = 0$.
for $t = 1$ **to** T **do**
  Send $\alpha_t g_{t-1}$ to $\mathcal{A}$ ad tth hint.
  $x_t \leftarrow \frac{\sum_{i=1}^t \alpha_i w_t}{\alpha_{1:t}}$.
  Play x_t, receive subgradient g_t.
  Send $\ell_t(x) = \langle \alpha_t g_t, x\rangle$ to $\mathcal{A}$ as the tth loss.
  Get w_{t+1} from $\mathcal{A}$.
end for
return x_T.

Theorem 3. *Suppose D has diameter B and $\mathcal{A}$ obtains the regret bound $R_T(x^\star) \le B\sqrt{2\sum_{t=1}^T \alpha_t^2\|\hat{g}_t - g_t\|^2}$ when given hints $\hat{g}_t$ ahead of the gradient g_t. Set $\alpha_t = t$ for all t. Suppose each g_t has variance at most σ^2, and $\mathcal{L}$ is L-smooth. Then Algorithm 3 yields:*

$$\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \le O\left(\frac{LB^2}{T^{3/2}} + \frac{\sigma B}{\sqrt{T}}\right)$$

Proof. Since we set $\hat{g}_t = g_{t-1}$, the assumption on $\mathcal{A}$ implies:

$$R_T(x^\star) \le B\sqrt{2\sum_{t=1}^T \alpha_t^2 \|g_{t-1} - g_t\|^2}$$

We can write $g_t = \nabla\mathcal{L}(x_t) + \zeta_t$ where ζ_t is some mean-zero random variable with $\mathbb{E}[\|\zeta_t\|^2] \le \sigma^2$. Then by smoothness, for $t > 1$ we have

$$\begin{aligned}\|g_t - g_{t-1}\| &\le \|\nabla\mathcal{L}(x_t) - \nabla\mathcal{L}(x_{t-1})\| + \|\zeta_t - \zeta_{t-1}\| \\ &\le L\|x_t - x_{t-1}\| + \|\zeta_t - \zeta_{t-1}\| \\ &\le \frac{L\alpha_t B}{\alpha_{1:t}} + \|\zeta_t\| + \|\zeta_{t-1}\| \\ \mathbb{E}[\|\hat{g}_t - g_t\|^2] &\le 5\frac{L^2\alpha_t^2 B^2}{(\alpha_{1:t})^2} + 10\sigma^2\end{aligned}$$

where in the last step we used $(a+b+c)^2 \le 5(a^2+b^2+c^2)$. Further, for $t = 1$, we have

$$\begin{aligned}\mathbb{E}[\|g_1\|^2] &\le \mathbb{E}[(\|\nabla\mathcal{L}(x_1) - \nabla\mathcal{L}(x^\star)\| + \|\zeta_t\|)^2] \\ \mathbb{E}[\|g_1 - \hat{g}_1\|^2] &\le 2L^2B^2 + 2\sigma^2 \le 5\frac{L^2B^2\alpha_1^2}{(\alpha_{1:1})^2} + 10\sigma^2\end{aligned}$$

Next, observe that $\alpha_{1:t} > t^2/2$ so that

$$\mathbb{E}[\|\hat{g}_t - g_t\|^2] \le 20\frac{L^2B^2}{t^2} + 10\sigma^2$$

Now observe $\sum_{t=1}^T t^2 < 3(T+1)^3/2$ and apply Jensen:

$$\begin{aligned}\mathbb{E}[R_T(x^\star)] &\le \mathbb{E}\left[B\sqrt{2\sum_{t=1}^T \alpha_t^2\|\hat{g}_t - g_t\|^2}\right] \\ &\le B\sqrt{30(T+1)^3\sigma^2 + 40L^2B^2T}\end{aligned}$$

And by Theorem 1 we have the desired result:

$$\mathbb{E}[\mathcal{L}(x_T) - \mathcal{L}(x^\star)] \le \frac{4\sqrt{10}LB^2}{T^{3/2}} + \frac{4\sqrt{10}\sigma B}{\sqrt{T}}$$

□

Note that the ordinary online-to-batch conversion may not be able to obtain this rate: here we are critically relying on the stability of the iterates x_t to guarantee that g_t and g_{t-1} are not too far apart, while in the standard online-to-batch conversion one would require stability in the w_t, which may not occur.

4.2. Acceleration

In the *deterministic* setting, (Levy et al., 2018) showed how to use adaptive step-sizes in conjuction with the linear-coupling framework (Allen-Zhu & Orecchia, 2014) to derive an accelerated algorithm that adapts to the smoothness parameter L. In this section we show that our Algorithm 1 and analysis is actually very similar in spirit to the linear-coupling scheme and so we can also derive an accelerated algorithm that adapts to both smoothness *and* variance optimally. To our knowledge this is the first accelerated algorithm to adapt to variance. Our analysis is arguably simpler than prior work: our proof is much shorter, we rely on only relatively simple properties of α_t and we do not use the internals of the online algorithm.

Unlike previously in this paper, but similar to (Levy et al., 2018), here we will require $\mathcal{L}$ to be defined on an entire vector space rather than potentially bounded domain D. We will also assume knowledge of some parameter B such that $\|x^\star\| \le B/2$. Lifting these restrictions are both valuable future directions.

Algorithm 4 Adaptive Stochastic Acceleration

Input: Bound $B \ge 2\|x^\star\|$, value c, Online learning algorithms $\mathcal{A}$ with domain $D = \{\|w\| \le B/2\}$.
Get initial point $w_1 \in D$ from $\mathcal{A}$.
$y_0 \leftarrow w_1$.
for $t = 1$ **to** T **do**
  $\alpha_t \leftarrow t$.
  $\tau_t \leftarrow \frac{\alpha_t}{\sum_{i=1}^t \alpha_i}$.
  $x_t \leftarrow (1-\tau_t)y_{t-1} + \tau_t w_t$.
  Play x_t, receive subgradient g_t.
  $\eta_t \leftarrow \frac{cB}{\sqrt{1+\sum_{i=1}^t \alpha_{1:i}\|g_i\|^2}}$
  $y_t \leftarrow x_t - \eta_t g_t$.
  Send $\ell_t(x) = \langle \alpha_t g_t, x\rangle$ to $\mathcal{A}$ as the tth loss.
  Get w_{t+1} from $\mathcal{A}$.
end for
return x_T.

Theorem 4. *Suppose $\mathbb{E}[g_t] = \nabla \mathcal{L}(x_t)$ for some L-smooth function $\mathcal{L}$ with domain an entire Hilbert space H. Suppose $\|g_t\| \le G$ with probability 1 and g_t has variance at most σ^2 for all t. Suppose $\|x^\star\| \le B/2$. Let D be the ball of radius $B/2$ in H and suppose $\mathcal{A}$ guarantees regret*

$$R_T(x^\star) \le kB\sqrt{\sum_{t=1}^T \alpha_t \|g_t\|^2}$$

For some k. Then with $c = \sqrt{2}k$, Algorithm 4 guarantees:

$$\mathbb{E}\left[\mathcal{L}(y_T) - \mathcal{L}(x^\star)\right] \le \frac{2\sqrt{2}kB + 2kLB^2\log(1+G^2T^3)}{T^2} + \frac{2kB\sigma\sqrt{2\log(1+G^2T^3)}}{\sqrt{T}}$$

Proof. The opening of our proof is again very similar to that of Theorem 1: observe that

$$\mathbb{E}\left[\sum_{t=1}^T \alpha_t(\mathcal{L}(x_t) - \mathcal{L}(x^\star))\right] \le \mathbb{E}\left[R_T(x^\star) + \sum_{t=1}^T a_{1:t-1}\langle g_t, y_{t-1} - x_t\rangle\right]$$

Next we use convexity again to argue $\mathbb{E}[\langle g_t, y_{t-1} - x_t\rangle] \le \mathbb{E}[\mathcal{L}(y_{t-1}) - \mathcal{L}(x_t)]$, and then we subtract $\mathbb{E}[\sum_{t=1}^T \alpha_t \mathcal{L}(x_t)]$ from both sides:

$$\mathbb{E}[-\alpha_{1:T}\mathcal{L}(x^\star)] \tag{2}$$

$$\le \mathbb{E}\left[R_T(x^\star) + \sum_{t=1}^T \alpha_{1:t-1}\mathcal{L}(y_{t-1}) - \alpha_{1:t}\mathcal{L}(x_t)\right] \tag{3}$$

Now we use smoothness to relate $\mathcal{L}(y_t)$ to $\mathcal{L}(x_t)$. Defining $\zeta_t = g_t - \nabla\mathcal{L}(x_t)$ and $\beta_t = \alpha_{1:t}$, we have:

$$\mathbb{E}[\mathcal{L}(y_t)] \le \mathbb{E}[\mathcal{L}(x_t) + \nabla\mathcal{L}(x_t)(y_t - x_t) + \frac{L}{2}\|x_t - y_t\|^2]$$
$$\le \mathbb{E}\left[\mathcal{L}(x_t) - \eta_t\|g_t\|^2 + \eta_t\langle\zeta_t, g_t\rangle + \frac{L\eta_t^2\|g_t\|^2}{2}\right]$$

Then multiply by β_t:

$$\mathbb{E}[\beta_t(\mathcal{L}(y_t) - \mathcal{L}(x_t))]$$
$$\le \mathbb{E}\left[-\frac{cB\beta_t\|g_t\|^2}{\sqrt{1+\sum_{i=1}^t \beta_i\|g_i\|^2}} + \frac{L\beta_t\eta_t^2\|g_t\|^2}{2} + \beta_t\langle\zeta_t, g_t\rangle\right]$$

Next, we borrow Lemma A.2 from (Levy et al., 2018): for positive numbers $x_1, \dots, x_n$

$$\sqrt{\sum_{i=1}^n x_i} \le \sum_{i=1}^n \frac{x_i}{\sqrt{\sum_{i'=1}^i x_{i'}}} \le 2\sqrt{\sum_{i=1}^n x_i}$$

Also, observe from concavity of log that:

$$\sum_{i=1}^n \frac{x_i}{1+\sum_{i'=1}^i x_{i'}} \le \log\left(1+\sum_{i=1}^n x_i\right)$$

Using this we obtain

$$\mathbb{E}\left[\sum_{t=1}^T \beta_t(\mathcal{L}(y_t) - \mathcal{L}(x_t))\right]$$
$$\le \mathbb{E}\left[-cB\sqrt{1+\sum_{t=1}^T \beta_t\|g_t\|^2} + \frac{c^2B^2L\log(1+G^2\beta_{1:T})}{2} + cB + \sum_{t=1}^T \langle\zeta_t, \beta_t g_t\rangle\eta_t\right]$$

Next, use Cauchy-Schwarz:

$$\mathbb{E}\left[\sum_{t=1}^T \langle\zeta_t, \beta_t g_t\rangle\eta_t\right] \le \mathbb{E}\left[\sqrt{\sum_{t=1}^T \beta_t\|\zeta_t\|^2}\sqrt{\sum_{t=1}^T \beta_t\|g_t\|^2\eta_t^2}\right]$$
$$\le \mathbb{E}\left[cB\sqrt{\sum_{t=1}^T \beta_t\|\zeta_t\|^2}\sqrt{\log\left(1+\sum_{t=1}^t \beta_t\|g_t\|^2\right)}\right]$$

And then use Jensen's inequality:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T \langle \zeta_t, \beta_t g_t\rangle \eta_t\right]\\
&\le \mathbb{E}\left[cB\sqrt{\sum_{t=1}^T \beta_t\|\zeta_t\|^2}\sqrt{\log(1+G^2\beta_{1:T})}\right]\\
&\le cB\sigma\sqrt{\beta_{1:T}\log(1+G^2\beta_{1:T})}
\end{aligned}$$

Where in the last line we observed $\mathbb{E}[\|\zeta_t\|^2]\le \sigma^2$. Combining everything, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T -\alpha_t\mathcal{L}(x^\star)\right]\\
&\le \mathbb{E}\left[R_T(x^\star)+\sum_{t=1}^T \alpha_{1:t-1}\mathcal{L}(y_{t-1})-\alpha_{1:t}\mathcal{L}(y_t)\right]\\
&+\mathbb{E}\left[\frac{c^2LB^2\log(1+G^2\beta_{1:T})}{2}-cB\sqrt{1+\sum_{t=1}^T\alpha_{1:t}\|g_t\|^2}\right.\\
&\qquad\left.-cB+cB\sigma\sqrt{\beta_{1:t}\log(1+G^2\beta_{1:t})}\right]
\end{aligned}$$

Now observe that $\alpha_{1:t}>\alpha_t^2/2$ and recall $R_T(x^\star)\le kB\sqrt{\sum_{t=1}^T\alpha_t^2\|g_t\|^2}$. Therefore since $c=\sqrt{2}k$ we cancel the $R_T(x^\star)$, observe $\beta_{1:T}\le\sum_{t=1}^T t^2\le T^3$, and telescope to obtain:

$$\begin{aligned}
\mathbb{E}[\alpha_{1:T}(\mathcal{L}(y_T)-\mathcal{L}(x^\star))]\le cB+\frac{c^2B^2L\log(1+G^2T^3)}{2}\\
+cBT^{3/2}\sigma\sqrt{\log(1+G^2T^3)}
\end{aligned}$$

and dividing by $\alpha_{1:T}=\frac{T(T+1)}{2}$ completes the proof. □

We remark also that, similar to the algorithm of (Levy et al., 2018), our Algorithm 4 is *universal* in the sense that for non-smooth losses we recover the $O(1/\sqrt{T})$ rate with no modifications. In fact, our analysis improves somewhat over (Levy et al., 2018) in that we maintain an adaptive convergence rate in the non-smooth setting.[1]

Theorem 5. *Suppose $\mathbb{E}[g_t]=\mathcal{L}(x_t)$ for some convex function $\mathcal{L}$. Then Algorithm 4 guarantees:*

$$\begin{aligned}
&\mathbb{E}[[\mathcal{L}(y_T)-\mathcal{L}(x^\star)]\\
&\le \mathbb{E}\left[\frac{2R_T(x^\star)+B\sqrt{2\sum_{t=1}^T t^2\|\nabla\mathcal{L}(y_t)\|^2}\sqrt{\log(1+G^3T^3)}}{T^2}\right]
\end{aligned}$$

Note that in the setting with $\|g_t\|\le G$ and $R_T(x^\star)=O\left(\sqrt{\sum_{t=1}^T\alpha_t^2\|g_t\|^2}\right)$, Theorem 5 implies a convergence rate of $O(\sqrt{\log(T)/T})$.

[1]We suspect this same adaptive non-smooth rate can be achieved by (Levy et al., 2018) via similar improved analysis.

Proof. We start from (3), and again proceed to relate $\mathcal{L}(y_t)$ to $\mathcal{L}(x_t)$, this time without the aid of smoothness:

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(y_t)-\mathcal{L}(x_t)]&\le\mathbb{E}[\langle\nabla\mathcal{L}(y_t),y_t-x_t\rangle]\\
&\le\mathbb{E}[\|\nabla\mathcal{L}(y_t)\|\|g_t\|\eta_t]
\end{aligned}$$

So by Cauchy-Schwarz, again defining $\beta_t=\alpha_{1:t}$ we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T\beta_t(\mathcal{L}(y_t)-\mathcal{L}(x_t))\right]\le\mathbb{E}\left[\sum_{t=1}^T\beta_t\|\nabla\mathcal{L}(y_t)\|\|g_t\|\eta_t\right]\\
&\le\mathbb{E}\left[\sqrt{\sum_{t=1}^T\beta_t\|\nabla\mathcal{L}(y_t)\|^2]}\sqrt{\sum_{t=1}^T\beta_t\|g_t\|^2\eta_t^2}\right]\\
&\le\mathbb{E}\left[B\sqrt{\sum_{t=1}^T\beta_t\|\nabla\mathcal{L}(y_t)\|^2}\sqrt{\log(1+G^3T^3)}\right]
\end{aligned}$$

And combining everything yields

$$\begin{aligned}
&\mathbb{E}[-\alpha_{1:T}\mathcal{L}(x^\star)]\\
&\le\mathbb{E}\left[R_T(x^\star)+B\sqrt{\sum_{t=1}^T\beta_t\|\nabla\mathcal{L}(y_t)\|^2}\sqrt{\log(1+G^3T^3)}\right.\\
&\qquad\left.+\sum_{t=1}^T\alpha_{1:t-1}\mathcal{L}(y_{t-1})-\alpha_{1:t}\mathcal{L}(y_t)\right]
\end{aligned}$$

Telescope the sum and rearrange to prove the theorem. □

5. Conclusion

We have provided a variant on the standard online-to-batch conversion technique that enables us to compute gradients at the iterates produced by the conversion algorithm rather than those produced by the online learning algorithm. This stabilizes the sequence of iterates and enables low regret even with respect to arbitrary polynomial weights. We show how to apply our approach to easily remove the log factors in stochastic strongly-convex optimization. Further, for smooth losses, we gain stability in the gradients which can be used by optimistic online algorithms. Finally, a small modification allows us to achieve the optimal stochastic accelerated rates. Not only is this the first method to adapt to both variance and smoothness optimally, it also is more general than prior analyses by virtue of being a black-box reduction from *any* sufficiently adaptive online learning algorithm. Finally, a recent connection between optimism and acceleration by (Wang & Abernethy, 2018) suggests that it may be possible to improve our optimistic analysis to match the accelerated rate in an even simpler manner.

References

Allen-Zhu, Z. and Orecchia, L. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Cesa-Bianchi, N., Conconi, A., and Gentile, C. On the generalization ability of on-line learning algorithms. *Information Theory, IEEE Transactions on*, 50(9):2050–2057, 2004.

Chiang, C.-K., Yang, T., Lee, C.-J., Mahdavi, M., Lu, C.-J., Jin, R., and Zhu, S. Online optimization with gradual variations. In *Conference on Learning Theory*, pp. 6–1, 2012.

Cutkosky, A. and Busa-Fekete, R. Distributed stochastic optimization via adaptive sgd. In *Advances in Neural Information Processing Systems*, pp. 1914–1923, 2018.

Cutkosky, A. and Orabona, F. Black-box reductions for parameter-free online learning in banach spaces. *arXiv preprint arXiv:1802.06293*, 2018.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference on Learning Theory (COLT)*, 2010.

Hazan, E. and Kale, S. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80(2-3):165–188, 2010.

Hazan, E. and Kale, S. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Hazan, E., Rakhlin, A., and Bartlett, P. L. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, pp. 65–72, 2008.

Lacoste-Julien, S., Schmidt, M., and Bach, F. A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012.

Lan, G. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

Levy, K. Online to offline conversions, universality and adaptive minibatch sizes. In *Advances in Neural Information Processing Systems*, pp. 1613–1622, 2017.

Levy, Y. K., Yurtsever, A., and Cevher, V. Online adaptive methods, universality and acceleration. In *Advances in Neural Information Processing Systems*, pp. 6501–6510, 2018.

McMahan, H. B. A survey of algorithms and analysis for adaptive online learning. *arXiv preprint arXiv:1403.3465*, 2014.

McMahan, H. B. and Streeter, M. Adaptive bound optimization for online convex optimization. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, 2010.

Mohri, M. and Yang, S. Accelerating online convex optimization via adaptive prediction. In *Artificial Intelligence and Statistics*, pp. 848–856, 2016.

Rakhlin, A. and Sridharan, K. Online learning with predictable sequences. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, pp. 993–1019, 2013. URL http://jmlr.org/proceedings/papers/v30/Rakhlin13.html.

Rakhlin, A., Shamir, O., Sridharan, K., et al. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, volume 12, pp. 1571–1578. Citeseer, 2012.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Srebro, N., Sridharan, K., and Tewari, A. Smoothness, low noise and fast rates. In *Advances in neural information processing systems*, pp. 2199–2207, 2010.

Wang, J.-K. and Abernethy, J. D. Acceleration through optimistic no-regret dynamics. In *Advances in Neural Information Processing Systems*, pp. 3828–3838, 2018.