A Greedy Framework for First-Order Optimization

Jacob Steinhardt*
Department of Computer Science
Stanford University
Stanford, CA 94305
jsteinhardt@cs.stanford.edu

Jonathan Huggins*
Department of EECS
Massachusetts Institute of Technology
Cambridge, MA 02139
jhuggins@mit.edu

Introduction. Recent work has shown many connections between conditional gradient and other first-order optimization methods, such as herding [3] and subgradient descent [2]. By considering a type of *proximal conditional method*, which we call boosted mirror descent (BMD), we are able to unify all of these algorithms into a single framework, which can be interpreted as taking successive arg-mins of a sequence of surrogate functions. Using a standard online learning analysis based on Bregman divergences, we are able to demonstrate an $O(1/T)$ convergence rate for all algorithms in this class.

Setup. More concretely, suppose that we are given a function $L : U \times \Theta \to \mathbb{R}$ defined by

$$L(u, \theta) = h(u) + \langle u, \theta \rangle - R(\theta) \tag{1}$$

and wish to find the *saddle point*

$$L_* \stackrel{\text{def}}{=} \min_u \max_\theta L(u, \theta). \tag{2}$$

We should think of u as the primal variables and θ as the dual variables; we will assume throughout that h and R are both convex. We will also abuse notation and define $L(u) \stackrel{\text{def}}{=} \max_\theta L(u, \theta)$; we can equivalently write $L(u)$ as

$$L(u) = h(u) + R^*(u), \tag{3}$$

where R^* is the Fenchel conjugate of R. Note that $L(u)$ is a convex function. Moreover, since $R \leftrightarrow R^*$ is a one-to-one mapping, for *any* convex function L and *any* decomposition of L into convex functions h and R^*, we get a corresponding two-argument function $L(u, \theta)$.

Primal algorithm. Given the function $L(u, \theta)$, we define the following optimization procedure, which will generate a sequence of points $(u_1, \theta_1), (u_2, \theta_2), \dots$ converging to a saddle point of L. First, take a sequence of weights $\alpha_1, \alpha_2, \dots$, and for notational convenience define

$$\hat{u}_t = \frac{\sum_{s=1}^t \alpha_s u_s}{\sum_{s=1}^t \alpha_s} \quad \text{and} \quad \hat{\theta}_t = \frac{\sum_{s=1}^t \alpha_s \theta_s}{\sum_{s=1}^t \alpha_s}.$$

Then the *primal boosted mirror descent* (PBMD) algorithm is:

1. $u_1 \in \arg\min_u h(u)$
2. $\theta_t \in \arg\max_{\theta \in \Theta} \langle \theta, u_t \rangle - R(\theta) = \partial R^*(u_t)$
3. $u_{t+1} \in \arg\min_u h(u) + \langle \hat{\theta}_t, u \rangle = \partial h^*(-\hat{\theta}_t)$

As long as h is strongly convex, for the proper choice of α_t we obtain the bound (see Corollary 2):

$$\sup_{\theta \in \Theta} L(\hat{u}_T, \theta) \leq L_* + O(1/T). \tag{4}$$

As an example, suppose that we are given a γ-strongly convex function f: that is, $f(x) = \frac{\gamma}{2}\|x\|_2^2 + f_0(x)$, where f_0 is convex. Then we let $h(x) = \frac{\gamma}{2}\|x\|_2^2$, $R^*(x) = f_0(x)$, and obtain the updates:

*Both authors contributed equally to this work.

1. $u_1 = 0$
2. $\theta_t = \partial f_0(u_t)$
3. $u_{t+1} = -\frac{1}{\gamma}\hat{\theta}_t = -\frac{\sum_{s=1}^{t} \alpha_s \partial f_0(u_s)}{\gamma \sum_{s=1}^{t} \alpha_s}$

We therefore obtain a variant on subgradient descent where u_{t+1} is a weighted average of the first t subgradients (times a step size $1/\gamma$). Note that these are the subgradients of f_0, which are related to the subgradients of f by $\partial f_0(x) = \partial f(x) - \gamma x$.

Dual algorithm. We can also concern the dual form of our mirror descent algorithm (*dual boosted mirror descent*, or DBMD):

1. $\theta_1 \in \arg\min_\theta R(\theta)$
2. $u_t \in \arg\min_u h(u) + \langle \theta_t, u\rangle = \partial h^*(-\theta_t)$
3. $\theta_{t+1} \in \arg\max_{\theta\in\Theta} \langle \theta, \hat{u}_t\rangle - R(\theta) = \partial R^*(\hat{u}_t)$

Convergence now hinges upon strong convexity of R rather than h, and we obtain the same $1/T$ convergence rate (see Corollary 4):

$$\sup_{\theta\in\Theta} L(\hat{u}_T, \theta) \leq L_* + O(1/T). \tag{5}$$

An important special case is $h(u) = \left\{ \begin{array}{lcl} 0 & : & u \in K \\ \infty & : & u \notin K \end{array} \right.$, where K is some compact set. Also let $R^* = f$, where f is an arbitrary strongly convex function. Then we are minimizing f over the compact set K, and we obtain the updates from conditional gradient:

1. $\theta_1 = \partial f(0)$
2. $u_t \in \arg\min_{u\in K} \langle \theta_t, u\rangle$
3. $\theta_{t+1} = \partial f(\hat{u}_t)$

Our notation is slightly different from previous presentations in that we use linear weights (α_t) instead of geometric weights (often denoted ρ_t, as in [2]). However, under the mapping $\alpha_t = \rho_t / \prod_{s=1}^{t}(1-\rho_s)$, we obtain an algorithm equivalent to the usual formulation.

Discussion. Our framework is intriguing in that it involves a purely greedy minimization of surrogate loss functions (alternating between the primal and dual), and yet is powerful enough to capture both subgradient descent and conditional gradient descent, as well as a host of other first-order methods, including the low-rank SDP solver introduced by Arora, Hazan, and Kale [1]. Briefly, the AHK algorithm seeks to maximize $\sum_{j=1}^{m} \frac{1}{2}(\mathrm{Tr}(A_j^T X) - b_j)^2$ subject to the constraints $X \succeq 0$ and $\mathrm{Tr}(X) \leq \rho$.[1] We can then define

$$h(X) = \left\{ \begin{array}{lcl} 0 & : & X \succeq 0 \text{ and } \mathrm{Tr}(X) \leq \rho \\ \infty & : & \text{else} \end{array} \right. \tag{6}$$

and

$$R^*(X) = \sum_{j=1}^{m} \frac{1}{2}(\mathrm{Tr}(A_j^T X) - b_j)^2. \tag{7}$$

Note that this decomposition is actually a special case of the conditional gradient decomposition above, and so we obtain the updates

$$X_{t+1} \in \mathrm{argmin}_{X\succeq 0, Tr(X)\leq\rho} \sum_{j=1}^{m} \left[\mathrm{Tr}(A_j^T \hat{X}_t) - b_j\right] \mathrm{Tr}(A_j^T X), \tag{8}$$

whose solution turns out to be $\rho v v^T$, where v is the top singular vector of the matrix $-\sum_{j=1}^{m} \left[\mathrm{Tr}(A_j^T \hat{X}_t) - b_j\right] A_j$. This example serves both to illustrate the flexibility of our framework and to highlight the interesting fact that the Arora-Hazan-Kale SDP algorithm is a special case of conditional gradient (to get the original formulation in [1], we need to replace the function $\frac{1}{2}x^2$ with $x_+ \log x_+$, where $x_+ = \max(x, 0)$).

[1] This is actually a variant of their algorithm, which we present for ease of exposition.

q-herding. In addition to unifying several existing methods, our framework allows us to extend herding to a an algorithm that we call *q-herding*. Herding is an algorithm for constructing pseudosamples that match a specified collection of moments from a distribution; it was originally introduced by Welling [4] and was shown to be a special case of conditional gradient by Bach et al. [3]. It can be cast as trying to minimize $\|\mathbb{E}_\mu[\phi(x)] - \bar{\phi}\|_2^2$ for μ in the probability simplex over $\mathcal{X}$, for a given $\phi : \mathcal{X} \to \mathbb{R}^d$ and $\bar{\phi} \in \mathbb{R}^d$. As shown in [3], the herding updates are equivalent to DBMD with $h(\mu) \equiv 0$ and $R(\theta) = \theta^T \bar{\phi} + \frac{1}{2}\|\theta\|_2^2$. The implicit assumption in the herding algorithm is that $\|\phi(x)\|_2$ is bounded. We are able to construct a more general algorithm that only requires $\|\phi(x)\|_p$ to be bounded for some $p \geq 2$. This *q-herding* algorithm works by taking $R(\theta) = \theta^T \bar{\phi} + \frac{1}{q}\|\theta\|_q^q$, where $\frac{1}{p} + \frac{1}{q} = 1$. In this case our convergence results imply that $\|\mathbb{E}_\mu[\phi(x)] - \bar{\phi}\|_p^p$ decays at a rate of $O(1/T)$.

Convergence results. We end by stating our formal convergence results. For the primal algorithm (PBMD) we have:

Theorem 1. *Suppose that h is strongly convex with respect to a norm $\|\cdot\|$ and let $r = \sup_\theta \|\theta\|_*$. Then*

$$\sup_\theta L(\hat{u}, \theta) \leq \sup_\theta L(u^*, \theta) + \frac{2r^2}{A_T} \sum_{t=1}^{T} \frac{\alpha_{t+1}^2 A_t}{A_{t+1}^2}. \tag{9}$$

Corollary 2. *Under the hypotheses of Theorem 1, for $\alpha_t = 1$ we have*

$$\sup_\theta L(\hat{u}, \theta) \leq \sup_\theta L(u^*, \theta) + \frac{2r^2(\log(T) + 1)}{T}. \tag{10}$$

and for $\alpha_t = t$ we have

$$\sup_\theta L(\hat{u}, \theta) \leq \sup_\theta L(u^*, \theta) + \frac{8r^2}{T}. \tag{11}$$

Similarly, for the dual algorithm (DBMD) we have:

Theorem 3. *Suppose that R is strongly convex with respect to a norm $\|\cdot\|$ and let $r = \sup_u \|u\|_*$. Then*

$$\sup_\theta L(\hat{u}, \theta) \leq \sup_\theta L(u^*, \theta) + \frac{2r^2}{A_T} \sum_{t=1}^{T} \frac{\alpha_{t+1}^2 A_t}{A_{t+1}^2}. \tag{12}$$

Corollary 4. *Under the hypotheses of Theorem 3, for $\alpha_t = 1$ we have*

$$\sup_\theta L(\hat{u}, \theta) \leq \sup_\theta L(u^*, \theta) + \frac{2r^2(\log(T) + 1)}{T} \tag{13}$$

and for $\alpha_t = t$ we have

$$\sup_\theta L(\hat{u}, \theta) \leq \sup_\theta L(u^*, \theta) + \frac{8r^2}{T} \tag{14}$$

Thus, a step size of $\alpha_t = t$ yields the claimed $O(1/T)$ convergence rate.

References

[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 339–348. IEEE, 2005.

[2] F Bach. Duality between subgradient and conditional gradient methods. *arXiv.org*, November 2012.

[3] F Bach, Simon Lacoste-Julien, and Guillaume Obozinski. On the Equivalence between Herding and Conditional Gradient Algorithms. In *ICML*. INRIA Paris - Rocquencourt, LIENS, March 2012.

[4] Max Welling. Herding dynamical weights to learn. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, June 2009.