Konstantinos Slavakis,
Seung-Jun Kim,
Gonzalo Mateos,
and Georgios B. Giannakis

Stochastic Approximation vis-à-vis Online Learning for Big Data Analytics

We live in an era of data deluge, where data translate to knowledge and can thus contribute in various directions if harnessed and processed intelligently. There is no doubt that signal processing (SP) is of uttermost relevance to timely big data applications such as real-time medical imaging, smart cities, network state visualization and anomaly detection (e.g., in the power grid and the Internet), health informatics for personalized treatment, sentiment analysis from online social media, Web-based advertising, recommendation systems, sensor-empowered structural health monitoring, and e-commerce fraud detection, just to name a few. Accordingly, abundant chances unfold to SP researchers and practitioners for fundamental contributions in big data theory and practice.

With such big blessings, however, come big challenges. The sheer volume and dimensionality of data often make it impossible to run analytics and traditional batch inferential methods on standalone processing units. With regards to scalability, online data processing is well motivated as the computational complexity of jointly processing the entire data set as a batch is prohibitive. Furthermore, there are many applications in which data themselves are made available in a streaming fashion, meaning that smaller chunks of data are acquired sequentially in time, e.g., nodes of a large network transmitting small blocks of data to a central unit continuously and incoherently in time. As information sources unceasingly produce data in real time, analytics must often be performed on the fly, typically without a chance to revisit previous data. In addition, big data tasks are often subject to stringent time constraints so that a high-quality answer obtained slowly via batch techniques can be less useful than a medium-quality answer that is obtained fast in an online fashion.

Digital Object Identifier 10.1109/MSP.2014.2345536
Date of publication: 15 October 2014

RELEVANCE

In this context, this lecture note presents recent advances in online learning for big data analytics. It is demonstrated that many of these approaches, mostly developed within the machine-learning discipline, have strong ties with workhorse statistical SP tools such as stochastic approximation (SA) and stochastic gradient (SG) algorithms. Important differences and novel aspects are highlighted as well. A key message conveyed is that seminal works on SA, such as by Robbins–Monro and Widrow, which go back half a century, can play instrumental roles in modern online learning tasks for big data analytics. Consequently, ample opportunities arise for the SP community to contribute in this growing and inherently cross-disciplinary field, spanning multiple areas across science and engineering.

THIS LECTURE NOTE PRESENTS RECENT ADVANCES IN ONLINE LEARNING FOR BIG DATA ANALYTICS.

PREREQUISITES

The required background includes basics of linear algebra, probability theory, convex analysis, and stochastic optimization.

STOCHASTIC APPROXIMATION BASICS

Consider the prototypical statistical learning problem in the realm of stochastic optimization (SO) [2], [3] where given a loss function f, one aims at minimizing the expected loss $\mathbb{E}_y\{f(\boldsymbol{w};\boldsymbol{y})\}$, possibly augmented with a complexity-controlling convex regularizer $r(\boldsymbol{w})$, with respect to (w.r.t.) a deterministic parameter (weight) vector $\boldsymbol{w}\in\mathscr{W}$. An example of $r(\boldsymbol{w})$ is the recently popular sparsity-promoting l_1-norm of the $p\times 1$ vector $\boldsymbol{w}$ where $r(\boldsymbol{w})=\|\boldsymbol{w}\|_1:=\sum_{i=1}^{p}|w_i|$. Expectation $\mathbb{E}_y\{\cdot\}$ is taken w.r.t. the typically unknown probability distribution of data $\boldsymbol{y}$ describing, e.g., input-response pairs in a supervised learning setting, and $\mathscr{W}$ denotes a subset of some Euclidean space, introduced here to cover general cases where constraints are imposed on $\boldsymbol{w}$. In lieu of the aforementioned distributional information, given training data $\{\boldsymbol{y}_t\}_{t=1}^{T}$ one can instead opt for solving the empirical risk minimization problem

$$\min_{\boldsymbol{w}\in\mathscr{W}}\frac{1}{T}\sum_{t=1}^{T}f(\boldsymbol{w};\boldsymbol{y}_t)+r(\boldsymbol{w}),\qquad(1)$$

which is an approximation of its ensemble counterpart, specifically $\min_{\boldsymbol{w}\in\mathscr{W}}[\mathbb{E}_y\{f(\boldsymbol{w};\boldsymbol{y})\}+r(\boldsymbol{w})]$. Beyond a purely learning paradigm, one should appreciate the generality offered by (1), since it can subsume, e.g., (constrained) maximum-likelihood problems with f identified as the log-likelihood function and data assumed statistically independent.

In big data settings, T can be huge, potentially infinite in a real-time paradigm where t identifies time instances of data acquisition. Moreover, the search space $\mathscr{W}$ can be excessively high-dimensional with complex structure. This observation justifies the inclusion of a regularizer in (1) to effectively reduce the dimensionality

1053-5888/14©2014IEEE

and/or size of $\mathscr{W}$ and yield parsimonious models that are interpretable and have satisfactory predictive performance. Unsurprisingly, there has been growing interest over the last decade in devising scalable and fast online algorithms for big data learning tasks such as (1).

The main premise of SO is centered around solving the minimization task [cf. (1)]

$$\min_{w\in\mathbb{R}^p}[\varphi(w):=\mathbb{E}_y\{f(w;y)\}] \qquad (2)$$

without having $\mathbb{E}_y\{\cdot\}$ available; see, e.g., [3]. (Compared to (1) and its ensemble version, both $\mathscr{W}$ and the regularizer r have been dropped here for brevity.) Key features present in SO algorithms are: 1) The data comprise a sequence of either dependent vectors with (asymptotically) vanishing covariance, or, independent identically distributed (i.i.d.) realizations $\{y_t\}_{t=1}^T$ of y; and, 2) given (w, y_t), there is a means of obtaining an unbiased "stochastic" gradient estimate $\nabla f(w; y_t)$, so that $\mathbb{E}_y\{\nabla f(w; y_t)\} = \nabla\varphi(w)$.

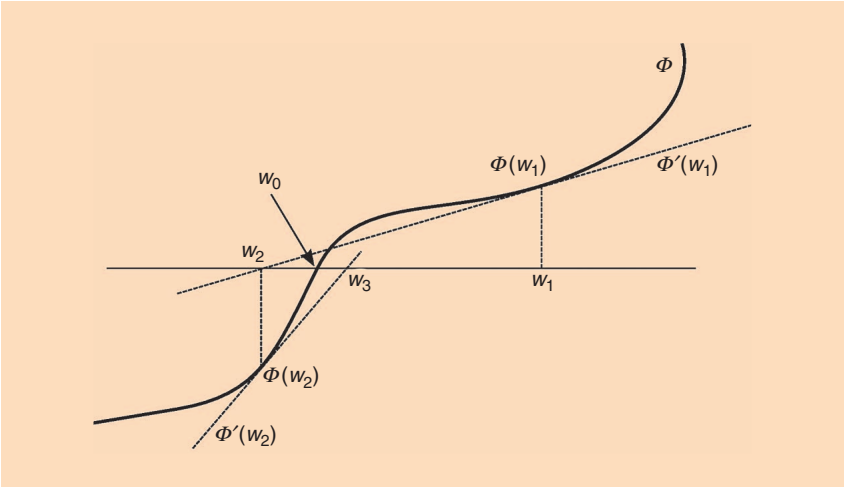
For φ smooth, minimizing φ in (2) amounts to searching for a zero of $\Phi(w):=\nabla\varphi(w)$, i.e., a w_0 such that (s.t.) $\Phi(w_0)=0$ [3]. The classical Newton–Raphson (N-R) algorithm provides the means to achieve this goal. For w scalar and with $'$ denoting differentiation, the sequence generated by the recursion $w_{k+1}:=w_k-\Phi(w_k)/\Phi'(w_k)=w_k-\varphi'(w_k)/\varphi''(w_k)$ converges under mild conditions to a root of $\Phi(w)$, and thus to a minimizer of $\varphi(w)$. An illustration of the N-R iteration can be seen in Figure 1. Starting from w_1 and using the derivatives $\{\Phi'(w_k)\}_{k=1}^{+\infty}$ in the N-R iteration, the resultant updates $\{w_k\}_{k=2}^{+\infty}$ gradually approach w_0, where $\Phi(w_0)=0$. Such a simple recursion can be readily extended to the $p\times 1$ vector case as $w_{k+1}:=w_k-H_\varphi^{-1}(w_k)\nabla\varphi(w_k)$, where now $H_\varphi(w_k)$ stands for the $p\times p$ Hessian matrix of φ at w_k with (i,j)th entry $\partial^2\varphi(w_k)/(\partial w_i\partial w_j)$.

Clearly, the N-R algorithm cannot be applied if $\mathbb{E}_y\{\cdot\}$ is not available; e.g., if the probability density function (pdf) of y is unknown, or, when computing $\mathbb{E}_y\{\cdot\}$ entails cumbersome integration over high-dimensional domains. To alleviate this burden, SA through the celebrated Robbins–Monro algorithm relies on the sequence of realizations $\{y_t\}$ and ingeniously uses the instantaneous $\nabla f(w_t; y_t)$ instead of the ensemble $\nabla\varphi(w_k)$ (indexes have been changed from k to t, for time-adaptive operation). With μ_t denoting the step-size, SA generates the online (or stochastic) gradient descent (OGD) iteration

$$w_{t+1}=w_t-\mu_t\nabla f(w_t;y_t), \qquad (3)$$

which "learns" expectations on the fly. This point is better illustrated in "Online Averaging as SA."

Several well-known adaptive SP and online learning algorithms stem from OGD.



[FIG1] The N-R method for finding a w_0 s.t. $\Phi(w_0)=0$.

LMS AS SA

Consider, for instance, scalar d_t and vector x_t processes that comprise the training data collected in $y_t:=[d_t, x_t^\top]^\top$, and let $f(w;y_t):=(d_t-w^\top x_t)^2/2$, where $\top$ stands for transposition. It can be readily verified that $\nabla f(w;y_t)=(w^\top x_t-d_t)x_t$, and application of OGD yields $w_{t+1}=w_t-\mu_t(w^\top x_t-d_t)x_t$, which is nothing but the celebrated least mean-squares (LMS) algorithm [3].

RLS AS SA

The OGD class can be further broadened by allowing matrix step-sizes $\{M_t\}$ instead of scalar ones $\{\mu_t\}$ to obtain $w_{t+1}=w_t-M_t\nabla f(w_t;y_t)$. To highlight the potential of this extension, consider (jointly) wide sense stationary $\{d_t, x_t\}_{t=1}^\infty$, with $C_{xx}:=\mathbb{E}_x\{x_tx_t^\top\}$, as well as $r_{dx}:=\mathbb{E}_{d,x}\{d_tx_t\}$. It turns out that the solution of $\min_w\mathbb{E}_{d,x}\{(d_t-w^\top x_t)^2\}$ is the linear minimum mean-square error estimator $w_0=C_{xx}^{-1}r_{dx}$. However, without knowing C_{xx} one relies on the sample average estimate $\hat{C}_t:=(1/t)\sum_{\tau=1}^t x_\tau x_\tau^\top$, and on OGD with $M_t:=(1/t)\hat{C}_t^{-1}$ to obtain

$$w_{t+1}=w_t-\frac{1}{t}\hat{C}_t^{-1}x_t(w_t^\top x_t-d_t) \qquad (4a)$$

$$\hat{C}_{t+1}^{-1}=\frac{t+1}{t}\Big[\hat{C}_t^{-1}-\hat{C}_t^{-1}x_{t+1}x_{t+1}^\top\hat{C}_t^{-1}/(t+x_{t+1}^\top\hat{C}_t^{-1}x_{t+1})\Big], \qquad (4b)$$

where the matrix inversion lemma is applied to carry out efficiently the inversion in (4b). Recursions (4) comprise the well-known recursive least-squares (RLS) algorithm [3].

ONLINE AVERAGING AS SA

The solution of $\min_{\boldsymbol{w}}\mathbb{E}_{\boldsymbol{y}}\{\|\boldsymbol{w}-\boldsymbol{y}\|_2^2/2\}$ is clearly $\boldsymbol{w}_0=\mathbb{E}_{\boldsymbol{y}}\{\boldsymbol{y}\}$. Following the SA rationale, consider $f(\boldsymbol{w};\boldsymbol{y}_t):=\|\boldsymbol{w}-\boldsymbol{y}_t\|_2^2/2$. The OGD iteration is $\boldsymbol{w}_{t+1}=\boldsymbol{w}_t-\mu_t(\boldsymbol{w}_t-\boldsymbol{y}_t)$, and if $\boldsymbol{w}_1:=\boldsymbol{0}$ as well as $\mu_t:=1/t$, simple mathematical induction yields $\boldsymbol{w}_{t+1}=(1/t)\sum_{\tau=1}^t\boldsymbol{y}_\tau$, which in accordance with the law of large numbers converges to $\boldsymbol{w}_0=\mathbb{E}_{\boldsymbol{y}}\{\boldsymbol{y}\}$ as $t\to+\infty$ [3].

PERFORMANCE OF SA ALGORITHMS

Based on the samples $\{y_t\}$, SA algorithms produce estimates $\{w_t\}$ that allow for estimation, tracking, and out-of-sample inference tasks, such as prediction. Performance analysis of SA schemes has leveraged advances in martingale and ordinary differential equation theories to establish, e.g., in the stationary case, convergence of $\{w_t\}$ to a time-invariant w_0 in probability, or with probability one, or in the mean-square sense [3]. In this stationary setting, convergence of OGD requires step-sizes selected to diminish with a certain rate. Specifically, $\{\mu_t\}$ must satisfy 1) $\mu_t \geq 0$, 2) $\lim_{t\to\infty}\mu_t = 0$, and 3) $\sum_{t=1}^{\infty}\mu_t = +\infty$. Clearly, 1)–3) are satisfied for $\mu_t := 1/t$, which vanishes as $t \to +\infty$ but not too fast so that 3) enables $\{w_t\}$ to reach asymptotically the desired w_0.

Departing from the standard route of SA convergence analysis [3], recent results take advantage of convexity if it is present in the objective function. Specifically for convex costs, the OGD recursion (3) generalizes to: $w_{t+1} = \mathcal{P}_{\mathcal{W}}[w_t - \mu_t \nabla f(w_t; y_t)]$, where $\mathcal{P}_{\mathcal{W}}(w) := \operatorname{argmin}_{w'\in\mathcal{W}} \|w - w'\|_2$ stands for the projection mapping onto a closed and convex constraint set $\mathcal{W}$. For φ differentiable and strongly convex with index $c > 0$, it holds that $\varphi(w') \geq \varphi(w) + (w' - w)^{\top}\nabla\varphi(w) + (c/2)\|w' - w\|_2^2$, for all (w, w). With step-sizes selected as $\mu_t := \mu/t$ with $\mu > 1/(2c)$, and for bounded stochastic gradients as in $\sup_w \mathbb{E}_y\left\{\|\nabla f(w; y)\|_2^2\right\} \leq \Delta$, it can be verified that the error $\mathbb{E}_y\left\{\|w_t - w_0\|_2^2\right\}$, where $w_0 = \operatorname{argmin}_{w\in\mathcal{W}}\mathbb{E}_y\{f(w; y)\}$, satisfies the following finite-sample bound [2]:

$$\mathbb{E}_y\{\|w_t - w_0\|_2^2\} \leq \frac{Q(\mu)}{t},$$

with

$$Q(\mu) := \max\{\mu^2\Delta^2/(2\mu c - 1), \|w_1 - w_0\|^2\}.$$

If, in addition, $\nabla\varphi$ is L-Lipschitz continuous, i.e., $\|\nabla\varphi(w) - \nabla\varphi(w')\|_2 \leq L\|w - w'\|_2$, $\forall w, w'$, then a similar finite-sample bound holds also for the sequence of function values $\{\varphi(w_t)\}$ [2]

$$\mathbb{E}_y\{\varphi(w_t) - \varphi(w_0)\} \leq \frac{LQ(\mu)}{2t},$$

where expectation is taken over $\{w_t\}$, which involves stochastic gradients.

Performance analysis of SA algorithms deals with convergence of $\{w_t\}$, whereas the online convex optimization framework outlined in a subsequent section starts from (1), invokes fewer or no assumptions on the underlying pdfs, and asserts convergence of the costs $\{f(w_t; y_t)\}$, rather than $\{w_t\}$.

THE OCO FRAMEWORK CAN BE VIEWED AS A MULTIROUND GAME BETWEEN A PLAYER (LEARNER) AND AN ADVERSARY.

Recently, SA was combined with the alternating direction method of multipliers (ADMM) which is attractive for offline optimization of composite costs [4]. The resultant SA-ADMM solver [5] is suitable for online optimization of composite costs such as $\min_{w\in\mathcal{W}}[\mathbb{E}_y\{f(w; y)\} + r(w)]$, in a fully distributed fashion—an operational mode that is highly desirable for big data applications.

SEQUENTIAL OPTIMIZATION AND DATA SKETCHING

The importance of sequential optimization along with the attractive operation of random sampling (also known as *sketching*) of big data will be illustrated in this subsection in the context of the familiar LS task:

$$\min_{w\in\mathbb{R}^p}\left[\frac{1}{2T}\|X^{\top}w - d\|_2^2 = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{2}(x_t^{\top}w - d_t)^2\right], \quad (5)$$

where $X := [x_1, \ldots, x_T]$ denotes the $p \times T$ matrix that gathers all available regressor or input vectors, and $d := [d_1, \ldots, d_T]^{\top}$ the $T \times 1$ vector of desired outputs (responses). Although irrelevant to the minimization in (5), the normalization with T is included to draw connections with (1). In this sense, the loss function becomes $f(w; y_t) = (x_t^{\top}w - d_t)^2/2$, with $y_t := [d_t, x_t^{\top}]^{\top}$, and its gradient $\nabla f(\cdot; y_t)$ is Lipschitz continuous with constant $L_t = \|x_t\|_2^2$. Different from the previous discussion, here T is fixed, and "online" means processing $\{d_t, x_t\}_{t=1}^{T}$ sequentially.

Searching for a solution w_0 of (5) requires eigen-decomposition of $XX^{\top}$, which incurs complexity $\mathcal{O}(T\,p^2)$. Alternatively, the standard gradient descent recursion $w_{k+1} = w_k - \mu_k(XX^{\top}w_k - Xd)$ entails $\mathcal{O}(p^2)$ computations per iteration k. Both cases are prohibitive in big data settings where the number of samples, T, is massive and/or the data dimensionality, p, can be huge. To surmount these obstacles, solving for w_0 can rely on subsampling (also known as *sketching* to obtain a subset of) the rows of $X^{\top}$, along with the corresponding entries of d, to reduce complexity w.r.t. T, while visiting them in a sequential fashion that scales linearly with p.

Kaczmarz's algorithm, a special case of the projections onto convex sets (POCS) method [6], produces a sequence of estimates $\{w_k\}$ to solve (5). For an arbitrary initial estimate w_1, the kth iteration of Kaczmarz's algorithm selects a row $t(k)$ of $X^{\top}$, together with the corresponding entry $d_{t(k)}$, and projects the current estimate w_k onto the set of all minimizers $\mathcal{H}_{t(k)} := \{w | x_{t(k)}^{\top}w = d_{t(k)}\}$ of $f(w; y_{t(k)})$, which is nothing but a hyperplane (a closed and convex set). Hence, the $(k+1)$st estimate is

$$w_{k+1} := \mathcal{P}_{\mathcal{H}_{t(k)}}(w_k) = w_k - \frac{x_{t(k)}^{\top}w_k - d_{t(k)}}{\|x_{t(k)}\|_2^2}x_{t(k)}, \quad (6)$$

where $\mathcal{P}_{\mathcal{H}_{t(k)}}$ stands for the projection mapping onto $\mathcal{H}_{t(k)}$. Notice here that the complexity of computing $\mathcal{P}_{\mathcal{H}_{t(k)}}(w_k)$ scales linearly with p. If every (d_t, x_t) is visited infinitely often, then under several conditions (6) converges to a solution of (5) [6]. Visiting each (d_t, x_t) a large number of times is prohibitive with big data since T can be excessively large. In contrast, poor selection of rows can slow down convergence; see Figure 2. Nevertheless, randomly drawing rows with equal probabilities has been shown empirically to accelerate convergence relative to cyclic revisits of rows [7]. Judicious sampling schemes can yield further speedups, as highlighted in "Accelerating SG via Nonuniform Sampling."

LEARNING VIA ONLINE CONVEX OPTIMIZATION

Recently, online learning approaches based on a online convex optimization (OCO) framework have attracted significant attention, as they do not require elaborate statistical models for data and yet can provide robust performance guarantees. This is true even under an adversarial setup, where the data sequence $\{\boldsymbol{y}_t\}$ may be generated strategically in reaction to the learner's iterates $\{\boldsymbol{w}_t\}$, as in the humans-in-the-loop applications such as the Web advertising optimization.

The OCO framework can be viewed as a multiround game between a player (learner) and an adversary [10]. In the context of the learning formulation in (1), the learner plays an action $\boldsymbol{w}_t \in \mathscr{W}$ in round t, where $\mathscr{W}$ is assumed to be closed and convex. Based on the action $\boldsymbol{w}_t$ that the player took, the adversary provides some feedback information $\mathscr{F}_t$, manifested in the data (feature) vector $\boldsymbol{y}_t$, based on which a convex loss function $\mathscr{L}_t: \mathscr{W} \to \mathbb{R} \cup \{+\infty\}$ is constructed, such as $\mathscr{L}_t(\boldsymbol{w}) := f(\boldsymbol{w}; \boldsymbol{y}_t) + r(\boldsymbol{w})$. The learner then suffers the loss at $\boldsymbol{w}_t$, specifically, $\mathscr{L}_t(\boldsymbol{w}_t)$. The overall process is depicted in Figure 3.

The learner's goal is to minimize the so-termed *regret* $R(T)$ over T rounds, defined as

$$R(T) := \sum_{t=1}^{T} \mathscr{L}_t(\boldsymbol{w}_t) - \min_{\boldsymbol{w} \in \mathscr{W}} \sum_{t=1}^{T} \mathscr{L}_t(\boldsymbol{w}), \tag{7}$$

which captures how much worse the learner performed cumulatively, compared to the case where a single best action is chosen with the knowledge of the entire sequence of cost functions $\{\mathscr{L}_t\}_{t=1}^{T}$ in hindsight. In particular, OCO aims at producing a sequence $\{\boldsymbol{w}_t\}$, which gives rise to sublinear regret, that is, the one with $R(T)/T \to 0$ as T grows. The key question now for the learner is how to pick $\boldsymbol{w}_t$ in each round t.

OCO ALGORITHMS AND PERFORMANCE

An important class of algorithms that can achieve the desired sublinear regret bound is based on the online mirror descent (OMD) iteration [11]. In a nutshell, the method minimizes a first-order approximation of $\mathscr{L}_t$ at the current iterate $\boldsymbol{w}_t$, while encouraging the search in the vicinity of $\boldsymbol{w}_t$. Specifically, OMD computes the next round iterate $\boldsymbol{w}_{t+1}$ as

$$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathscr{W}} (\boldsymbol{w} - \boldsymbol{w}_t)^{\top} \mathscr{L}_t'(\boldsymbol{w}_t) + \frac{1}{\mu} D_{\psi}(\boldsymbol{w}, \boldsymbol{w}_t), \tag{8}$$

where $'$ denotes a (sub)gradient of a function, $\mu > 0$ is a learning rate parameter, and $D_{\psi}(\boldsymbol{w}, \boldsymbol{v})$ is the Bregman divergence associated with a continuously differentiable and strongly convex ψ, defined as

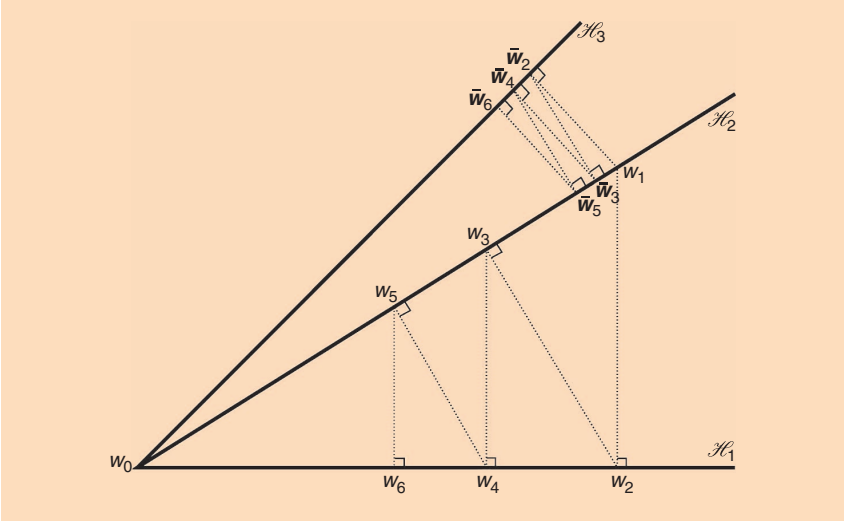
$$D_{\psi}(\boldsymbol{w}, \boldsymbol{v}) := \psi(\boldsymbol{w}) - \psi(\boldsymbol{v}) - (\boldsymbol{w} - \boldsymbol{v})^{\top} \nabla \psi(\boldsymbol{v}). \tag{9}$$

ACCELERATING SG VIA NONUNIFORM SAMPLING

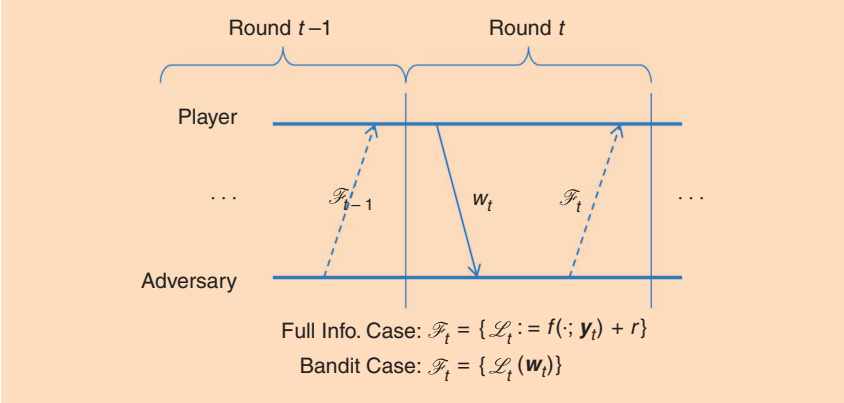
In the noiseless case ($\boldsymbol{X}^{\top}\boldsymbol{w} = \boldsymbol{d}$), randomly drawing rows in proportion to their Lipschitz constants L_t is known to provide finite-sample bounds of the form [7]

$$\mathbb{E}_{\mathscr{R}}\{\|\boldsymbol{w}_k - \boldsymbol{w}_0\|_2^2\} \le [1 - \kappa(\boldsymbol{X})^{-2}]^k \|\boldsymbol{w}_1 - \boldsymbol{w}_0\|_2^2,$$

where $\kappa(\boldsymbol{X})$ stands for the condition number of $\boldsymbol{X}$, and $\mathbb{E}_{\mathscr{R}}\{\cdot\}$ denotes expectation w.r.t. the distribution over which $\{d_t, \boldsymbol{x}_t\}$ are selected. The previous nonuniform sampling scheme yields better convergence rates than those resulting from uniform sketching [7]. More information on (non)uniform sketching and its application to SG descent methods can be found in [8] and [9].



[FIG2] Kaczmarz's algorithm for three hyperplanes $\{\mathscr{H}_t\}_{t=1}^{3}$ with the nonempty intersection $\{\boldsymbol{w}_0\} = \cap_{t=1}^{3} \mathscr{H}_t$. Row (hyperplane) selection affects convergence rate; $\{\boldsymbol{w}_k\}$, which alternates between $\mathscr{H}_1$ and $\mathscr{H}_2$ approaches $\boldsymbol{w}_0$ faster than $\{\bar{\boldsymbol{w}}_k\}$, which is generated via $\mathscr{H}_2, \mathscr{H}_3$.



[FIG3] OCO as a multiround game.

In the special case of using $\psi(\boldsymbol{w}) := \|\boldsymbol{w}\|_2^2/2$, the corresponding $D_\psi(\boldsymbol{w},\boldsymbol{v}) = \|\boldsymbol{w}-\boldsymbol{v}\|_2^2/2$, and the OMD update in (8) boils down to OGD [10], establishing an immediate link between OCO and SA. In general, a judicious choice of ψ can capture the structure of the search space $\mathscr{W}$, leading to an efficient update formula for $\boldsymbol{w}_t$. For example, when $\mathscr{W}$ is the probability simplex, i.e., $\mathscr{W} = \{\boldsymbol{w} \mid w_i \geq 0, \sum_i w_i = 1\}$, setting $\psi(\boldsymbol{w}) := \sum_i w_i \log w_i$ in (8) and (9) yields the exponentiated gradient algorithm, which obviates the need to explicitly impose the probability simplex constraints [10]. Aiming at an efficient use of prior information on $\boldsymbol{w}$, a notable generalization of OMD is offered by the "COMID Algorithm."

Both COMID and OMD (which is a special case of COMID) can attain sublinear regret bounds. Specifically, $R(T) = \mathcal{O}(\sqrt{T})$ in general, and the bound becomes $\mathcal{O}(\log T)$ when $\mathscr{L}_t$ is strongly convex [10], [12]. Noteworthy differences between SA and OCO are outlined in "SA vis-à-vis OCO."

ONLINE LEARNING WITH BANDIT FEEDBACK

The bandit setup of OCO refers to the case where the feedback $\mathscr{F}_t$ from the adversary does not explicitly reveal the cost function $\mathscr{L}_t(\cdot)$ but only the sample cost $\mathscr{L}_t(\boldsymbol{w}_t)$ due to action $\boldsymbol{w}_t$; refer also to Figure 3. For example, $\boldsymbol{w}_t$ may represent the advertising budget allocated to different media channels, and $\mathscr{L}_t(\boldsymbol{w}_t)$ the corresponding overall cost (e.g., the total advertising expense minus the resulting income). In this case, it may be difficult to know the explicit form of $\mathscr{L}_t$, but $\mathscr{L}_t(\boldsymbol{w}_t)$ can be easily observed.

SEQUENTIAL OR ONLINE LEARNING SCHEMES TOGETHER WITH RANDOM SAMPLING OR DATA SKETCHING METHODS ARE EXPECTED TO PLAY A PRINCIPAL ROLE IN SOLVING LARGE-SCALE OPTIMIZATION TASKS.

The idea of bandit OCO is to estimate the necessary gradient using SA in the context of OGD. Specifically, a key observation is that if one can evaluate a function $f: \mathbb{R}^p \to \mathbb{R}$ at $\boldsymbol{w}$ perturbed by a small $\delta\boldsymbol{v}$, where $\delta > 0$ and $\boldsymbol{v}$ is uniformly distributed on the surface of a unit sphere, then $(p/\delta) f(\boldsymbol{w}+\delta\boldsymbol{v})\boldsymbol{v}$ offers an unbiased estimate of the gradient at $\boldsymbol{w}$ of a locally smoothed version of f [14]. Thus, plugging this noisy gradient directly into the OGD update in the spirit of SA, one can still establish a sublinear regret bound. However, the best bound found in [14] is $\mathcal{O}(T^{3/4})$, slower than the $\mathcal{O}(\sqrt{T})$-bound for the full information case, illustrating the price to pay for the lack of information.

LESSONS LEARNED AND FUTURE AVENUES

This lecture note offered a short exposition of recent advances in online learning for big data analytics, highlighting their differences and many similarities with prominent statistical SP tools such as SA and SO methods. It was demonstrated that the seminal Robbins–Monro algorithm, the workhorse behind several classical SP tools such as the LMS and RLS algorithms, carries rich potential for solving large-scale learning tasks under low computational budget. It was also explained that sequential or online learning schemes together with random sampling or data sketching methods are expected to play a principal role in solving large-scale optimization tasks. A short description of the OCO framework revealed its flexibility on the variety of optimization tasks that can be accommodated, including scenarios where data are provided in an adversarial fashion or with limited feedback. Yet, such a flexibility comes at a price; OCO-based statistical analysis refers mostly to bounds of the regret cost. Based on the common ground between OCO and SA, OCO can only benefit from the rich theoretical armory of SA, e.g., the martingale theory, where results pertain also to convergence of the primal (random) variables of the optimization task at hand. Vice versa, SA can also profit from the powerful toolbox of convex analysis, the engine behind OCO, for establishing strong analytical claims in the big data context. In closing, Figure 4 depicts the unique and complementary strengths SA, SO, and OCO offer to online learning, as well as adaptive SP theory and big data applications.

COMID ALGORITHM

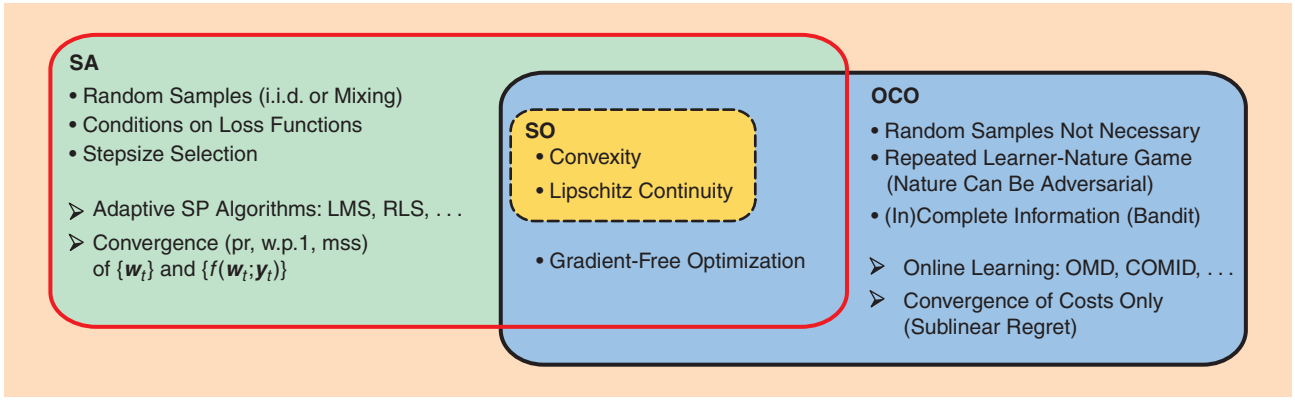
While the OMD update provides a computationally attractive solution to (1), the linearization involved often defeats one of the purposes of the regularizer r, which is to promote a priori known structure in the solution. For example, setting $r(\boldsymbol{w})$ proportional to the ℓ_1-norm of $\boldsymbol{w}$ encourages sparsity in $\boldsymbol{w}$. To properly capture such a benefit, one has to respect the composite structure of $\mathscr{L}_t$, which decomposes into the data-dependent part $f_t(\boldsymbol{w}) := f(\boldsymbol{w};\boldsymbol{y}_t)$ and the invariant part $r(\boldsymbol{w})$ [12], [13]. In particular, the composite objective mirror descent (COMID) algorithm relies on [12]

$$\boldsymbol{w}_{n+1} = \arg\min_{\boldsymbol{w}\in\mathscr{W}} (\boldsymbol{w}-\boldsymbol{w}_t)^\top f_t'(\boldsymbol{w}_t) + r(\boldsymbol{w}) + \frac{1}{\mu} D_\psi(\boldsymbol{w},\boldsymbol{w}_t), \tag{S1}$$

where it is seen that the regularizer is not linearized.

SA VIS-A-VIS OCO

Compared to the SA approaches, the OCO framework does not require stochastic models. This is a salient departure from typical SA setups, since the regret bounds are guaranteed even for $\{\boldsymbol{y}_t\}$ that may have been generated adversarially, i.e., with $\boldsymbol{y}_t$ arbitrary correlated to past actions $\{\boldsymbol{w}_\tau\}_{\tau\leq t}$ and past data $\{\boldsymbol{y}_\tau\}_{\tau<t}$. On the other hand, the bounds pertain to convergence of the sequence of costs rather than the iterates $\{\boldsymbol{w}_t\}$ themselves. Nonetheless, building upon the flexibility offered by OCO, certain limited feedback learning tasks are feasible as elaborated in the "Online Learning with Bandit Feedback" section, where, interestingly, the SA ideas prove instrumental once again.



[FIG4] **SA/SO vis-à-vis OCO: features and implications.**

ACKNOWLEDGMENTS

The work in this lecture note was supported by the National Science Foundation grants EARS-1343248 and EAGER-1343860, and the MURI grant AFOSR FA9550-10-1-0567.

AUTHORS

Konstantinos Slavakis (kslavaki@umn.edu) is a research associate professor in the Department of Electrical and Computer Engineering and Digital Technology Center, University of Minnesota, United States.

Seung-Jun Kim (sjkim@umbc.edu) is an assistant professor in the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, United States.

Gonzalo Mateos (gmateosb@ece.rochester.edu) is an assistant professor in the Department of Electrical and Computer Engineering, University of Rochester, New York, United States.

Georgios B. Giannakis (georgios@umn.edu) is a professor in the Department of Electrical and Computer Engineering and director of the Digital Technology Center, University of Minnesota, Minneapolis, United States.

REFERENCES

[1] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics," *IEEE Signal Processing Mag.*, vol. 31, no. 5, pp. 18–31, Sept. 2014.

[2] A. Nemirovski, A. Juditski, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, 2009.

[3] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer, 1997.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. Belmond, MA: Athena Scientific, 1997.

[5] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Processing*, vol. 57, no. 6, pp. 2365–2381, 2009.

[6] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM Review*, vol. 38, no. 3, pp. 367–426, Sept. 1996.

[7] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *J. Fourier Anal. Appl.*, vol. 15, no. 2, pp. 262–278, 2009.

[8] D. Needell, N. Srebro, and R. Ward. (2013, Feb.). Stochastic gradient descent and the randomized Kaczmarz algorithm. ArXiv e-prints. [Online]. Available: arXiv:1310.5715v2

[9] A. Nedić and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.

[10] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, Mar. 2012.

[11] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Oper. Res. Lett.*, vol. 31, no. 3, pp. 167–175, 2003.

[12] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, "Composite objective mirror descent," in *Proc. Conf. Learning Theory*, Haifa, Israel, June 2010, pp. 14–26.

[13] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 2543–2596, Oct. 2010.

[14] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, Vancouver, Jan. 2005, pp. 385–394.

[SP]