A Note on the Unification of Adaptive Online Learning

Wenwu He, James Tin-Yau Kwok, *Senior Member, IEEE*, Ji Zhu, and Yang Liu

***Abstract*— In online convex optimization, adaptive algorithms, which can utilize the second-order information of the loss function's (sub)gradient, have shown improvements over standard gradient methods. This paper presents a framework Follow the Bregman Divergence Leader that unifies various existing adaptive algorithms from which new insights are revealed. Under the proposed framework, two simple adaptive online algorithms with improvable performance guarantee are derived. Furthermore, a general equation derived from a matrix analysis generalizes the adaptive learning to nonlinear case with kernel trick.**

***Index Terms*— Adaptive gradient descent (GD), Follow the Bregman Divergence Leader (FTBDL), online learning, second-order information.**

I. Introduction

ONLINE learning, in which the instances arrive sequentially, is a popular and natural approach in many real-time and life-long learning problems. It is also advantageous in large-scale learning because of its efficiency and competitive performance.

In the basic setting of online convex optimization, an online algorithm iteratively estimates a weight $w_t \in \mathcal{F} \subseteq \mathbb{R}^n$ ($\mathcal{F}$ is assumed to be closed and convex). w_t is often used to define a prediction function $f_t(x) = \langle w_t, x\rangle$ on an input instance $x \in \mathbb{R}^n$ at round t. Then, the algorithm suffers a loss $\ell_t(w_t)$, where $\ell_t(\cdot)$ is also convex. Typically, its performance over a total of T iterations is measured by the regret $\text{Regret}_T = \sum_{t=1}^{T}(\ell_t(w_t) - \ell_t(\mathring{w}))$, where $\mathring{w} \in \mathcal{F}$ is a competitor. Note that this paper is focusing on online learning algorithms with full information and that the entire loss function and the gradient (or Hessian) are observed and computable, that is, bandit-type algorithms are not considered herein.

A standard learning procedure for online algorithms is the gradient descent (GD) [1], which updates the weight as $w_{t+1} = \Pi_{\mathcal{F}}(w_t - \eta_t g_t)$. Here, g_t is the gradient (or subgradient) of ℓ_t with respect to w_t, $\eta_t > 0$ is the stepsize, and $\Pi_{\mathcal{F}}$ is the Euclidean projection operator onto $\mathcal{F}$. A common setting for the stepsize is $\eta_t = \eta t^{-(1/2)}$ for some $\eta > 0$.

In cases where different feature dimensions carry different amounts of information, GD can be significantly improved by incorporating the second-order information of the loss's (sub)gradient. A variety of these adaptive algorithms have been studied in recent years. Two representative examples are the adaptive FOrward-Backward Splitting (FOBOS) (A-FOBOS) [2], [3] and adaptive regularized dual averaging (RDA) (A-RDA) [2], [4]. They update the weight as

$$w_{t+1} = \Pi_{\mathcal{F}}^{A_t}\left(w_t - \eta A_t^{-1} g_t\right)$$

where $\Pi_{\mathcal{F}}^{A}(\cdot) = \arg\min_{w\in\mathcal{F}} \|w - \cdot\|_A$ is the projection operator, $\|\cdot\|_A = (\langle\cdot, A\cdot\rangle)^{1/2}$, and the positive definite matrix A_t contains the second-order information. Another algorithm that is closely related to A-FOBOS is follow the proximal-regularized leader (FTPRL) [5], but without the sparseness regularizer. McMahan [6] also presents AODG [7] as a simpler version of adaptive RDA, where the adaptive behavior is realized by an identity matrix with time-varying magnitude.

Other adaptive algorithms include the second-order perceptron (SOP) [8], which updates the input correlation matrix and uses it for prediction. A similar algorithm that also uses the input correlation matrix is adaptive regularization of weights (AROW) [9]. It maintains a Gaussian distribution over the learned weights and combines it with the large margin principle. A variant of AROW that aims to obtain robust performance is narrow AROW (NAROW) [10], which uses both adaptive and fixed second-order information. A recent algorithm that uses a similar idea is exact soft confidence-weighted (SCW) learning [11]. It improves AROW by adding an adaptive margin.

Interestingly, for expconcave losses,[1] two algorithms, follow the approximate leader (FTAL) and online Newton step (ONS), also use the second-order information of the gradient and obtain the logarithmic regret of $O(\ln T)$ [15]. More recently, Orabona *et al.* [16] showed that the ONS has a regret on the

Manuscript received June 30, 2015; revised November 16, 2015; accepted January 28, 2016. This work was supported in part by the Research Grants Council, Hong Kong, under Grant 614513, in part by the China Scholarship Council Award, and in part by the National Natural Science Foundation of China under Grant 61304199 and Grant 61304210.

W. He is with the School of Mathematics and Physics, Fujian University of Technology, Fuzhou 350118, China, also with the Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China, and also with the Fujian Collaborative Innovation Center for Beidou Navigation and Intelligent Traffic, Fuzhou 350118, China (e-mail: hwwhbb@163.com).

J. T.-Y. Kwok is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: jamesk@cse.ust.hk).

J. Zhu is with the Department of Statistics and the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-1107 USA (e-mail: jizhu@umich.edu).

Y. Liu is with the Department of Computer Science, Harvard University, Cambridge, MA 02138 USA (e-mail: yangl@seas.harvard.edu).

Digital Object Identifier 10.1109/TNNLS.2016.2527053

[1] Examples of expconcave loss include the log-loss $\ell_t(w_t) = -\ln(\langle w_t, x_t\rangle)$ which arises in the problem of universal portfolio management [12], and the square loss $\ell_t(w_t) = (\langle w_t, x_t\rangle - y_t)^2$, which is widely used in regression problems [13], [14]. For a strongly convex loss, regret in scale of $O(\ln T)$ can be derived, but it is rarely used in learning problems. Therefore, expconcave can be viewed as a relaxation of strongly convex.

order of $\ln(1 + L_T^*)$ for smooth and expconcave losses, where L_T^* is the cumulative loss of the best competitor. In this case, the regret can become a constant when $L_T^* = 0$, and it is at most $O(\ln T)$.

Ross *et al.* [17] introduce a normalized adaptive gradient (NAG) descent algorithm that incorporates scale invariance to adaptive gradient. NAG is robust to features scales and collapses the range hyperparameter search required to achieve good performance. The idea of adaption presented in [17] is similar to that presented in [2] or [5]. A noticeable adaption method, which is different to what we discuss here, is variance-based stochastic gradient descent (SGD) (V-SGD) [18]. V-SGD uses a per-parameter learning rate proportional to an estimate of gradient squared divided by variance and second derivative. Schaul *et al.* [18] analyze the asymptotic convergence while the rate and the regret bound are not clear.

The theme of this paper is to develop a general framework for the understanding of existing adaptive algorithms, which can then allow the development of new algorithms. In particular, we introduce a framework, Follow the Bregman Divergence Leader (FTBDL), to unify the existing second-order online learning algorithms. In addition, two simple adaptive algorithms, adaptive exponential gradient (EG) and simple augment adaptive algorithm, are proposed, presented, and analyzed. Finally, the existing adaptive algorithms mainly consider linear learning and extending them to nonlinear cases is nontrivial. We propose a matrix equation and prove that it can provide a general way for extending linear adaptive learning to a nonlinear one.

Kernel trick is popular for nonlinear learning [19], [20]. The input x_t is first mapped to a reproducing kernel Hilbert space feature space $\mathcal{H}$, i.e., $x_t : \mapsto \phi(x_t) \subset \mathcal{H}$, then the inner production $\langle x_i, x_j \rangle$ is replaced with $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Due to the representation theorem [20], $w_t = \sum_{\tau \in S_t} \alpha_\tau \phi(x_\tau)$, where S_t is the support set. By the property of reproduction, $f_t(x) = \sum_{\tau \in S_t} \alpha_\tau k(x_\tau, x)$ is a case of being kernelized. Then, any algorithms of which the final computation can be reduced to inner product of input can be kernelized. However, the algorithm of interest may involve other operation, and then further discussion is deserved.

In the literature, Cesa-Bianchi *et al.* [8] use a matrix equation [21] to kernelize SOP. Another approach in [9] and [22] uses a representer theorem to kernelize AROW. Nevertheless, both methods may not be suitable to other adaptive algorithms, such as A-FOBOS or A-RDA in [2], where a matrix square root operation is involved. There are also other types of adaption for the stepsize, such as stochastic metadescent [23], which has also been used to kernelize online learning [24], [25]. These methods are different from what the paper considers, and overall lack a solid theoretical foundation for regret guarantees.

The rest of this paper is organized as follows. Section II provides a review on the well-known algorithm follow the regularizer leader (FTRL) and adaptive online learning in general. Section III proposes the general framework, FTBDL. The simple adaptive algorithm, adaptive EG, is also presented. The unification of adaptive algorithms is presented in Section IV, where the simple augment adaptive algorithm is introduced and analyzed. Section V presents a matrix equation that provides a general way to extend adaptive learning to nonlinear cases by kernelization. Section VI gives some concluding remarks. The proofs of the main results are included in the Appendix. More details can be found in the full version [26].

Notation: We use $\|x\|$ to denote the norm and, in particular, $\|x\|_p$ to denote the p-norm of a vector $x \in \mathbb{R}^n$, often $p \in \{1, 2, \infty\}$; $\mathbb{N}^+$ is the set of positive integers; S_{++}^n is the set of all $n \times n$ positive definite matrices; and S_+^n is the set of all $n \times n$ positive semidefinite matrices. Moreover, $A \succ 0$ (resp. $A \succeq 0$) when $A \in S_{++}^n$ (resp. $A \in S_+^n$). For a matrix A, $\det(A)$ is its determinant, $\mathrm{Tr}(A)$ is its trace, and $\mathrm{diag}(A)$ is its diagonal matrix. For $A \in S_{++}^n$, $\|x\|_A = (\langle x, Ax \rangle)^{1/2}$, where $\langle \cdot, \cdot \rangle$ is the inner product. Moreover, $\Pi_{\mathcal{F}}^A(v) = \arg\min_{u \in \mathcal{F}} \|u - v\|_A$, where $u, v \in \mathbb{R}^n$, is the projection of v based on $\| \cdot \|_A$.

II. Review

A differentiable function R is σ strongly convex with respect to a norm $\| \cdot \|$ if $R(u) - R(v) - \langle \nabla R(v), u - v \rangle \geq (\sigma/2)\|u-v\|^2$ for any $u, v \in \mathcal{F}$. When $\sigma = 1$, the function will be simply called strongly convex, e.g., $R(w) = (1/2)\|w\|_2^2$ is strongly convex.

Denote $B_R(\cdot, \cdot)$ as the measure BD and $\Pi_{R,\mathcal{F}}(v) = \arg\min_{u \in \mathcal{F}} B_R(u, v)$ is the projection based on it. For a strongly convex and differentiable function $R(\cdot)$, $B_R(u, v) = R(u) - R(v) - \langle \nabla R(v), (u - v) \rangle$ (for more details, see Section III-A). The conjugate dual (CD) of R is[2] $R^*(\theta) = \sup_{w \in \mathcal{F}} \{\langle w, \theta \rangle - R(w)\}$. Moreover, let $[t] = \{1, \ldots, t\}$, $(\cdot)_{1:t}$ be the shorthand for $\sum_{\tau=1}^t (\cdot)_\tau$, and $(\cdot)_{\sim S_t} = \sum_{\tau \in S_t} (\cdot)_\tau$ be the partial sum, where $S_t \subseteq [t]$. The loss $\ell_t(\cdot)$ is convex, and $g_t \in \partial \ell_t(w_t)$ is the gradient (subgradient) of $\ell_t(w_t)$. Let $G_p = \max_t \|g_t\|_p$, $D_p = \max_{x,y \in \mathcal{F}} \|x - y\|_p$ is the diameter of feasible region. A loss $\ell_t(w)$ is ε-expconcave if for $\forall w \in \mathcal{F}$ and $t > 0$, $\exists\ \varepsilon > 0$, such that $\nabla^2(\exp(-\varepsilon \ell_t(w))) \preceq 0$.

A. Follow the (Regularized) Leader

Follow the leader (FTL) [27], [28] is one of the classic online learning algorithm. For $t > 1$, it updates w_t as $w_{t+1} = \arg\min_{w \in \mathcal{F}} \sum_{\tau=1}^t \ell_\tau(w)$. FTL is based on empirical risk minimization and relies entirely on the observed history [29]. As such, its solution for w_t may shift drastically from round to round. An interesting example in which FTL fails can be found in [28].

A natural modification of the FTL is to add a regularizer, leading to the FTRL algorithm [27]–[29]. Given a strongly convex regularizer $R(\cdot)$, FTRL updates w_t as

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} R(w) + \eta \sum_{\tau=1}^{t} \ell_\tau(w) \qquad (1)$$

where $\eta > 0$ is the stepsize. A popular choice for $R(w)$ is $(1/2)\|w\|_2^2$ [28]. FTRL outputs any $w_1 \in \mathcal{F}$ in the first round.

[2] Examples: for $q > 1$, let $R(w) = (1/2)\|w\|_q^2$, then $R^*(\theta) = (1/2)\|\theta\|_p^2$, where $(1/p) + (1/q) = 1$; for $A \in S_{++}^n$, let $R(w) = (1/2)\|w\|_A^2$, then $R^*(w) = (1/2)\|w\|_{A^{-1}}^2$.

In the online learning literature, a standard trick for reduction is to replace $\ell_\tau(w)$ in (1) by $\langle g_\tau, w\rangle$, where $g_\tau \in \partial\ell_\tau(w_\tau)$ is the (sub)gradient of the loss at w_τ [1]. The justification is that as the loss ℓ_t is convex, we have $\ell_t(w_t) - \ell_t(\mathring{w}) \le \langle g_t, w_t - \mathring{w}\rangle$ for all $\mathring{w} \in \mathcal{F}$. Therefore, $\text{Regret}_T \le \sum_{t=1}^{T}\langle g_t, w_t - \mathring{w}\rangle$, and the right-hand side provides an upper bound for the regret.

B. Follow the Proximal-Regularized Leader

As mentioned in Section I, in cases where different feature dimensions carry different amounts of information, a fixed regularizer for all t as used in the standard GD may not be desirable, and it can be significantly improved by incorporating the second-order information of the loss (sub)gradient.[3] Given $Q_1 \in S^n_{++}$ and $Q_t \in S^n_{+}, \forall t \ge 2$, the FTPRL algorithm [5] adapts the regularizer as $R_t(w) = (1/2)\|Q_t^{1/2}(w - w_t)\|_2^2 = (1/2)\|w - w_t\|^2_{Q_t}$, and updates w_t as

$$w_{t+1} = \arg\min_{w\in\mathcal{F}} \sum_{\tau=1}^{t} (R_\tau(w) + \langle g_\tau, w\rangle). \tag{2}$$

Various Q_t's have been discussed in [5]. A simple approach is to define Q_t as a diagonal matrix with elements $(Q_t)_{i,i} = (Q_{1:t})_{i,i} - (Q_{1:t-1})_{i,i}$, where

$$(Q_{1:t})_{i,i} = \frac{\sqrt{\sum_{\tau=1}^{t}\|g_\tau\|_2^2}}{D_2/\sqrt{2}}. \tag{3}$$

This yields a regret bound of $\sqrt{2}D_2(\sum_{t=1}^{T}\|g_t\|_2^2)^{1/2}$ (see Section II-C). Recall that for the standard GD algorithm, its regret bound (with the optimal stepsize) is $\sqrt{2}D_2G_2\sqrt{T}$ [1], [28]. As $(\sum_{t=1}^{T}\|g_\tau\|_2^2)^{1/2} \le \max_t \|g_t\|_2\sqrt{T}$, FTPRL thus has a tighter upper bound on regret than GD.

C. Adaptive Online Learning

As discussed in Section I, besides FTPRL, there exist many other adaptive online learning algorithms. In the following, we provide a brief review. The relationships among them will be discussed in Section IV-A.

The A-FOBOS [2] updates w_t as

$$w_{t+1} = \arg\min_{w\in\mathcal{F}} B_{R_t}(w, w_t) + \eta\langle g_t, w\rangle \tag{4}$$

where $R_t = (1/2)\|w\|^2_{A_t}$, $A_t = aI + (\sum_{\tau=1}^{t} g_\tau g_\tau^\top)^{1/2}$, or simply $A_t = aI + (\text{diag}(Q_{1:t}))^{1/2}$, and $(Q_{1:t})_{i,i} = \sum_{\tau=1}^{t}(g_\tau[i])^2$, where $a \ge 0$.

A-FOBOS trades off the current gradient g_t (to get an improvement using the gradient information) and staying close to w_t using the BD defined on the proximal function R_t (to keep the learning process stable). Instead of using fixed R as in FOBOS [3], here, $R_t = (1/2)\|w\|^2_{A_t}$ that adapts the proximal function in a data-driven way. The resulting algorithm is similar to the second-order GD, and constructs an approximation to the Hessian of the loss.

Extending the adaptive idea to RDA [2], [4] renders A-RDA, which updates w_t as $w_{t+1} = \arg\min_{w\in\mathcal{F}}(1/t)R_t + \eta\langle\bar{g}_t, w\rangle$, where $\bar{g}_t = (1/t)g_{1:t}$ is the average gradient, and $R_t = (1/2)\|w\|^2_{A_t}$ as defined in A-FOBOS. Equivalently

$$w_{t+1} = \arg\min_{w\in\mathcal{F}} R_t + \eta\langle g_{1:t}, w\rangle. \tag{5}$$

Interestingly, McMahan [6] represents AODG [7] as a simpler version of A-RDA ($\eta = 1$, $R_t = (\sigma_{1:t}/2)\|w\|_2^2$, i.e., $A_t = \sigma_{1:t}I$), which updates w_t as $w_{t+1} = w_t - (1/\sigma_{1:t})(g_t + \sigma_t w_t) = -(g_{1:t}/\sigma_{1:t})$, where $\sigma_1 > 0$, $\sigma_t \ge 0$ for $t > 1$.

By using the input correlation matrix, the SOP in [8] updates w_t as[4]

$$w_t = \left(aI + \sum_{\tau\in\mathcal{M}_{t-1}\cup t} x_\tau x_\tau^\top\right)^{-1} \sum_{\tau\in\mathcal{M}_{t-1}} y_\tau x_\tau \tag{6}$$

where $a \ge 0$. As we know, the performance of the perceptron algorithm is governed by geometrical properties of the input data. It is harder to learn when the ellipsoid of the input data becomes more flat along the target hyperplane. Intuitively, the adaptive matrix plumps up the input data and makes it easier for the perceptron algorithm to learn.

In a similar form (but the motivation is to give a confidence over the weights to learn), NAROW in [10] updates w_t as

$$w_t = \left(I + \sum_{\tau\in\mathcal{M}_{t-1}\cup\mathcal{U}_{t-1}\cup t} \frac{x_\tau x_\tau^\top}{r_\tau}\right)^{-1} \sum_{\tau\in\mathcal{M}_{t-1}\cup\mathcal{U}_{t-1}} y_\tau x_\tau \tag{7}$$

where $r_\tau > 0$.

For an expconcave loss, FTAL [15] updates w_t as

$$w_{t+1} = \arg\min_{w\in\mathcal{F}} \sum_{\tau=1}^{t} \tilde{\ell}_\tau(w) \tag{8}$$

where $\tilde{\ell}_\tau(w) \triangleq \ell_\tau(w_\tau) + g_\tau^\top(w - w_\tau) + (1/2r)(w - w_\tau)^\top g_\tau g_\tau^\top(w - w_\tau)$ and $(1/r) = (1/2)\min\{(1/4D_2G_2), \varepsilon\}$. It is understandable that $\tilde{\ell}_\tau(w)$ is a second-order approximations of $\ell_\tau(w)$. As shown in [15], FTAL is equivalent to the following algorithm, named ONS for its close connection to the Newton method, which updates w_t as:

$$w_{t+1} = \Pi^{A_t}_{\mathcal{F}}\{A_t^{-1}b_t\} \tag{9}$$

where $A_t = \sum_{\tau=1}^{t} g_\tau g_\tau^\top$, $b_t = \sum_{\tau=1}^{t}(g_\tau g_\tau^\top w_\tau - r g_\tau)$, and $(1/r) = (1/2)\min\{(1/4D_2G_2), \varepsilon\}$. We will see that in the later part Section IV-A, of this paper, equivalently (9) can be updated with a closed-form (14), where $\eta = r$ and r and A_t are the same as in (9).

III. Adaptive Online Learning Framework

Section II lists a number of existing algorithms. To understand their behavior more clearly and shed new insights, we present in this section a general framework, which lays the basis of unification of existing algorithms.

[3] Interesting examples can be found in [2] and [5], or references in Section I.

[4] Here, $\mathcal{M}_t$ is the index set of the mistake rounds, and $\mathcal{U}_t$ is the index set of rounds, whose prediction is correct but $\ell_\tau(w_\tau) > 0$.

A. Bregman Divergence

A core theme of the section is that of the measure, i.e., BD, which was first introduced in [30]. Recalling the definition of BD, for a strongly convex and differentiable function $R(\cdot)$, $B_R(u, v) = R(u) - R(v) - \langle \nabla R(v), (u - v) \rangle$. That is, BD from u to v is the difference between $R(u)$ and its linear approximation via the first-order Taylor expansion of R at v. For the convexity of R, this difference is always nonnegative. In addition, by [31, Lemma 1], BD is strongly convex with respect to its first argument.

We introduce several properties of BD. First, $B_R(u, v) = 0$ when $u = v$, and in general, $B_R(u, v) \neq B_R(v, u)$. Hence, typically, BD is not a metric. We, however, have the following properties that are needed in this paper.

1) *Additive:* $B_{h+f}(u, v) = B_h(u, v) + B_f(u, v)$ if h and f are convex and differentiable.
2) $B_{h+f}(u, v) = B_h(u, v)$ if f is linear.
3) *Three-Point Equality:* $B_R(u, v) + B_R(v, w) = B_R(u, w) + \langle u - v, \nabla R(w) - \nabla R(v) \rangle$.

More properties of BD can be found in [27] and [32].

In particular, when $R(w) = (1/2)\|w\|_2^2$, $B_R(w, v) = (1/2)\|w - v\|_2^2$. When $R(w) = \sum_{i=1}^{n} (w[i] \ln w[i] - w[i])$, where $w_i \geq 0 \ \forall i$, $B_R(w, v) = \sum_{i=1}^{n} (w[i] \ln(w[i]/v[i]) + v[i] - w[i])$ is the generalized Kullback-Leibler (KL) divergence [27], [32], [33]. For the two BDs, Kivinen and Warmuth [33] developed the linear GD algorithm and EG algorithm, respectively. BD has also been used in other contexts, for instance, in clustering [34], in the learning with submodular functions [35], and recently in alternating direction method of multipliers [36].

Interestingly, BD is also helpful for online learning with limited feedback (bandit), as shown in [37], where a curious connection between the notion of BDs and self-concordant barriers is discussed and analyzed. In addition, BD can be extended to measure the nearness of matrix [38] to recover, for instance, the squared Frobenius norm which may be used in principal component analysis (PCA) or incremental PCA [39], or the von Neumann divergence which has been employed for online PCA [40]. An early work [41] shows the link between BD and PCA. Please refer to the references (and references within) for more details.

B. Follow the Bregman Divergence Leader

In this section, we propose the FTBDL algorithm (shown in Algorithm 1) to provide a basic framework for the unification of existing adaptive algorithms (a formal and deeper unification will be presented in Section IV). It replaces the fixed regularizer $R(w)$ of FTRL in (1) with $\sum_{\tau=1}^{t} B_{R_\tau}(w, v_\tau)$, which is adaptive. The update rule is

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \sum_{\tau=1}^{t} (B_{R_\tau}(w, v_\tau) + \eta \ell_\tau(w)) \tag{10}$$

where $R_\tau(\cdot)$ is strongly convex. Hereafter, we replace $\ell_\tau(w)$ with $\langle g_t, w \rangle$ unless otherwise specified. In the sequel, we either set $v_\tau = w_\tau$, so that the learner tries to keep w close to the w_τ learned in the previous round; or set v_τ to a fixed v (e.g., $v_\tau = 0$, or $v_\tau = 1$ when the KL divergence is used), such that the learner tries to keep it close to a fixed point v.

Algorithm 1 Follow the Bregman Divergence Leader

```
Input: $\eta > 0$, and a sequence of strongly convex and differentiable functions $R_1, \ldots, R_T$.
Initialize: $w_1 \in \mathcal{F}$.
for $t = 1, 2, \ldots$ do
    Suffer loss $\ell_t(w_t)$ and compute its subgradient $g_t$;
    $w_{t+1} = \arg\min_{w \in \mathcal{F}} \sum_{\tau=1}^{t} \big(B_{R_\tau}(w, v_\tau) + \eta \langle g_\tau, w \rangle\big)$.
end for
```

Remark 1: When $R_\tau(w) = (\eta/2)\|Q_\tau^{1/2} w\|_2^2$ and $v_\tau = w_\tau$, FTBDL reduces to FTPRL in Section II-B.

Remark 2: When $Q_\tau = g_\tau g_\tau^\top$, $R_\tau(w) = (\eta/2)\|Q_\tau^{1/2} w\|_2^2$, and $v_\tau = w_\tau$, FTBDL reduces to FTAL in Section II-C. Thus, the second-order information is captured by the adaptive regularizer $B_{R_\tau}(w, w_\tau)$, which is updated with newly arrived instances.

Let $\ell_\tau^{R_\tau}(w) = B_{R_\tau}(w, v_\tau) + \eta \langle g_\tau, w \rangle$. Equation (10) can be rewritten as

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \sum_{\tau=1}^{t} \ell_\tau^{R_\tau}(w) = \arg\min_{w \in \mathcal{F}} \ell_{1:t}^{R_\tau}(w) \tag{11}$$

where $\ell_{1:t}^{R_\tau}(w) \equiv \sum_{\tau=1}^{t} \ell_\tau^{R_\tau}(w)$. The following proposition will show that w_{t+1} can be computed in the closed-form:

$$w_{t+1} = \Pi_{R_{1:t}, \mathcal{F}}\left(\nabla R_{1:t}^{*} \left(\sum_{\tau=1}^{t} (\nabla R_\tau(v_\tau) - \eta g_\tau) \right) \right). \tag{12}$$

The proof is shown in Appendix A.

Proposition 1: For update rule (10), we have

$$w_{t+1} = \Pi_{\ell_{1:t}^{R_\tau}, \mathcal{F}}\left(\arg\min_{w \in \mathbb{R}^n} \ell_{1:t}^{R_\tau}(w) \right)$$

and that

$$\Pi_{\ell_{1:t}^{R_\tau}, \mathcal{F}}(v) = \Pi_{R_{1:t}, \mathcal{F}}(v).$$

For $\mathcal{F} = \mathbb{R}^n$, (11) can be solved by setting its gradient to be 0

$$\sum_{\tau=1}^{t} (\nabla R_\tau(v_\tau) - \eta g_\tau) = \sum_{\tau=1}^{t} \nabla R_\tau(w) = \nabla R_{1:t}(w).$$

Recall that $\nabla R^*(\cdot) = (\nabla R(\cdot))^{-1}$ [27], [28]. We obtain the update in (12).

Remark 3: When $R_\tau = (\sigma_\tau/2)\|w\|_2^2$, $v_\tau = 0$ and $\eta = 1$, (11) becomes a simple version of A-RDA

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \frac{\sigma_{1:t}}{2} \|w\|_2^2 + \langle g_{1:t}, w \rangle \tag{13}$$

and its closed-form solution is

$$w_{t+1} = \Pi_{\frac{\sigma_{1:t}}{2}\|w\|_2^2, \mathcal{F}}\left(-\frac{g_{1:t}}{\sigma_{1:t}} \right) = \Pi_{\mathcal{F}}\left(-\frac{g_{1:t}}{\sigma_{1:t}} \right).$$

C. Theoretical Properties

Adaptive algorithms improve over standard GD using the gradient's second-order information. Typically, the second-order information is captured by a properly defined matrix.[5] For example, A-FOBOS in (4) replaces the GD update with

$$w_{t+1} = \Pi_{\mathcal{F}}^{A_t}\big(w_t - \eta A_t^{-1} g_t\big). \tag{14}$$

Actually, for A-FOBOS, $B_{R_t}(w, w_t) = (1/2)\|w - w_t\|_{A_t}^2$, and the closed-form solution of (4) leads to (14). Clearly, when $A_t = \sqrt{t}I$, it reduces to standard GD.

As will be revealed in Section IV-A, different adaptive algorithms mainly differ in the choice of the matrix A_t, and on whether v_τ is origin-centered (i.e., $v_\tau = 0$ as in A-RDA) or updated iteratively (e.g., $v_\tau = w_\tau$ as in A-FOBOS). Here, we summarize several special cases and connect them back to existing methods.

1) $v_\tau = w_\tau$: Let $R_\tau(w) = (1/2)\|w\|_{A_\tau - A_{\tau-1}}^2$, then A-FOBOS is a specific case of the following update rule, called mirror descent (MD) [42], [43]:

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} B_{R_{1:t}}(w, w_t) + \eta\langle g_t, w\rangle. \tag{15}$$

Typically, MD is not considered with a changing function R. We generalize it by adding a strongly convex function R_t to the BD on each round. Equation (15) can be computed in the closed-form with (12), and as shown in Proposition 3, this updating rule-based learning algorithm has also a performance guarantee.

The following proposition will show the equivalence between MD and FTBDL, and then, FTBDL (10) covers A-FOBOS (4). The proof is shown in Appendix B.

Proposition 2: Assume that $\widehat{w}_1 = \widetilde{w}_1$. The MD update

$$\widehat{w}_{t+1} = \arg\min_{w \in \mathcal{F}} \{B_{R_{1:t}}(w, \widehat{w}_t) + \eta\langle g_t, w\rangle\}$$

and FTBDL update

$$\widetilde{w}_{t+1} = \arg\min_{w \in \mathcal{F}} \sum_{\tau=1}^{t} (B_{R_\tau}(w, \widetilde{w}_\tau) + \eta\langle g_\tau, w\rangle)$$

are equivalent in that $\widehat{w}_t = \widetilde{w}_t$ for all $t > 0$.

By the equivalence, regret bounds can be derived with ease from existing results. For example, using [2, Lemma 16] or [42, Th. 4.1], the regret for FTBDL ($v_\tau = w_\tau$) can be bounded as follows. The proof can be found in Appendix C.

Proposition 3: When $v_\tau = w_\tau$, the regret for Algorithm 1 with respect to $\mathring{w} \in \mathcal{F}$ is bounded by

$$\text{Regret}_T \le \frac{1}{\eta}\sum_{t=1}^{T} B_{R_t}(\mathring{w}, w_t) + \frac{\eta}{2}\sum_{t=1}^{T} \|g_t\|_{R_{1:t}^*}^2. \tag{16}$$

This allows us to derive specific regrets that correspond to different BDs using different R values.

For example, to recover FTPRL, set $R_t(w) = (1/2)\|w\|_{Q_t}^2$, $B_{R_t}(w, w_t) = (1/2)\|w - w_t\|_{Q_t}^2$, and $\eta = 1$. Then

$$\text{Regret}_T \le \sum_{t=1}^{T} \frac{1}{2}\|\mathring{w} - w_t\|_{Q_t}^2 + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{Q_{1:t}^*}^2. \tag{17}$$

The above regret gains a factor 1/2 in the second right-hand side term, compared with that of [5, Th. 2]. Using the diagonal Q_t in (3), (17) becomes

$$\text{Regret}_T \le \frac{1}{2}\max_{t \in [T]} \|w - w_t\|_2^2 Q_{1:T} + \frac{1}{2}\sum_{t=1}^{T} \frac{\|g_t\|_2^2}{Q_{1:t}}.$$

Furthermore, using [5, Lemma 7], i.e., for any nonnegative real number sequnce $s_1, \ldots, s_T$

$$\sum_{t=1}^{T} \frac{s_t}{\sqrt{\sum_{\tau=1}^{t} s_\tau}} \le 2\sqrt{\sum_{t=1}^{T} s_t} \tag{18}$$

the regret reduces to $\sqrt{2}D_2(\sum_{t=1}^{T}\|g_t\|_2^2)^{1/2} \le \sqrt{2}D_2 G_2 \sqrt{T}$, and then, it has a tighter regret than GD.

2) $v_\tau = v$: When v_τ is fixed to a given v, we can (recall the additive property of BD) rewrite (10) as

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \{B_{R_{1:t}}(w, v) + \eta\langle g_{1:t}, w\rangle\}. \tag{19}$$

Let $\widetilde{g}_t = g_t - \nabla R_t(v)$, (19) can be replaced with

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \{R_{1:t}(w) + \eta\langle \widetilde{g}_{1:t}, w\rangle\}.$$

Set $\nabla R_t(v) = 0$ (e.g., $\nabla R(0) = 0$ when $R(w) = (1/2)\|w\|_A^2$, also $\nabla R(1) = 0$ for KL divergence), this update further reduces to

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \{R_{1:t}(w) + \eta\langle g_{1:t}, w\rangle\}. \tag{20}$$

For update (20), we have the following regret (the proof is shown in Appendix D).

Proposition 4: Let $R_t(\cdot)$ be the strongly convex regularizer and w_t be the sequence generated by update rule (20), where $t \in [T]$. Then, the regret with respect to $\mathring{w} \in \mathcal{F}$ is bounded by

$$\text{Regret}_T \le \frac{1}{\eta}R_{1:T}(\mathring{w}) - \frac{1}{\eta}U + \frac{\eta}{2}\sum_{t=2}^{T} \|g_t\|_{R_{1:t-1}^*}^2 + O(1) \tag{21}$$

where $O(1)$ corresponds to the term $\langle g_1, w_1 - w_2\rangle$ and $U = \sum_{t=1}^{T} R_t(w_{t+1})$.

A similar result can be obtained for the general case in (19)

$$\text{Regret}_T \le \frac{1}{\eta}B_{R_{1:T}}(\mathring{w}, v) - \frac{1}{\eta}U + \frac{\eta}{2}\sum_{t=2}^{T} \|g_t\|_{R_{1:t-1}^*}^2 + O(1)$$

where $O(1)$ corresponds to the term $\langle g_1, w_1 - w_2\rangle \le G_2 D_2$ and $U = \sum_{t=1}^{T} B_{R_t}(w_{t+1}, v)$, which is always nonnegative.

The above bounds allow for more refined results to be derived in specific cases. Consider that simple A-RDA (13) and let $\sigma_{1:t} = 2\sqrt{2}G_2\sqrt{t}/D_2$. Bound (21) provides the

[5] Generally speaking, the second-order methods require $O(n^2)$ space to store the adaptive matrix A_t and $O(n^2)$ time each step to compute A_t^{-1} using Woodbury identity [21], given the gradient (subgradient). For the first-order methods like GD, only $O(n)$ time is required each step. One motivation using diagonal versions of adaptive online learning is to reduce the computation cost. Hazan *et al.* [15] provide a detailed discussion on the computational complexity including the computation of projection step. Second-order methods, however, may provide a regret lower than that of the first-order methods, and then fewer iterations are required, e.g., $O(\ln(T))$ regret for ONS versus $O(\sqrt{T})$ for GD.

Algorithm 2 Adaptive Exponential Gradient

```
Input: $\sigma_1 > 0$, $\sigma_t \geq 0$ for $t > 1$, $\eta > 0$ and $\eta_t = \frac{\eta}{\sigma_{1:t}}$; $B_R(w, v) = \sum_{i=1}^{n} w[i] \ln \frac{w[i]}{v[i]}$ and $\mathcal{F} = \{w \mid \|w\|_1 = 1, \forall i : w_i \geq 0\}$.
Initialize: $w_1 = v = [\frac{1}{n}, \cdots, \frac{1}{n}]^\top$.
for $t = 1, 2, \ldots$ do
    Suffer loss $\ell_t(w_t)$ and compute its subgradient $g_t$;
    $w_{t+1} = \arg\min_{w \in \mathcal{F}} \{B_R(w, v) + \eta_t \langle g_{1:t}, w\rangle\}$.
end for
```

regret $(\sqrt{2}/2)D_2G_2\sqrt{T} - U + O(1)$, which gains a factor $1/2$ (meanwhile $U > 0$) over standard GD ($\mathcal{F} = \{w \mid \|w\|_2 \leq D_2/2\}$) [1].

D. Adaptive Exponential Gradient

In this section, we present a new adaptive algorithm. Consider the normalized KL divergence, i.e., $R(w) = \sum_{i=1}^{n} w[i] \ln w[i]$, $w \in \mathcal{F}$, where $\mathcal{F} = \{w \mid \|w\|_1 = 1, \forall i : w_i \geq 0\}$, and then $B_R(w, v) = \sum_{i=1}^{n} w[i] \ln(w[i]/v[i])$. $R(w)$ is strongly convex with respect to $\|\cdot\|_1$ [28] and the dual norm $\|\cdot\|_\infty$. In addition, the CD is given by $R^*(\theta) = \ln(\sum_{i=1}^{n} e^{\theta[i]})$ [28]. The resulting algorithm is often called normalized EG [28], [33].

Now, we design a simple adaptive EG (Algorithm 2). Let $v = [(1/n), \ldots, (1/n)]^\top$, $\sigma_1 > 0$, $\sigma_t \geq 0$ for $t > 1$, $R_{1:t}(w) = \sigma_{1:t}R(w)$, and $\eta_t = (\eta/\sigma_{1:t})$. Using the fact that $B_{R_{1:t}}(w, v) = \sigma_{1:t}B_R(w, v)$, we update w_t ($w_1 = v$) with

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \{B_R(w, v) + \eta_t \langle g_{1:t}, w\rangle\}. \tag{22}$$

By (12), it can be updated in the closed-form, and formally, we have the following results.

Proposition 5: The closed-form update for adaptive EG algorithm (22) is

$$w_{t+1}[i] = \frac{w_t[i]e^{-\eta_t g_t[i]}}{\sum_{j=1}^{n} w_t[j]e^{-\eta_t g_t[j]}} \tag{23}$$

and it is equivalent to the following update:

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \{B_R(w, w_t) + \eta_t \langle g_t, w\rangle\} \tag{24}$$

where $w_1 = v$.

Algorithm 2 is similar to normalized EG but has an adaptive stepsize η_t. For its regret, we have the following upper bound.

Proposition 6: The regret for Algorithm 2 with respect to $\mathring{w} \in \mathcal{F}$ is bounded by

$$\text{Regret}_T \leq \frac{\sigma_{1:T}}{\eta} \ln n + \frac{\eta}{2} \sum_{t=1}^{T} \frac{\|g_t\|_\infty^2}{\sigma_{1:t}}. \tag{25}$$

In particular, setting $\sigma_{1:t} = \sqrt{t}$, $\eta = \sqrt{\ln n}/G_\infty$ yields

$$\text{Regret}_T \leq 2G_\infty\sqrt{T \ln n}. \tag{26}$$

Furthermore, setting $\sigma_{1:t} = (\sum_{\tau=1}^{t} \|g_\tau\|_\infty^2)^{1/2}$, $\eta = G_\infty$ yields

$$\text{Regret}_T \leq 2\sqrt{\ln n \sum_{t=1}^{T} \|g_t\|_\infty^2} \leq 2G_\infty\sqrt{T \ln n}. \tag{27}$$

Regret bound (26) is identical to that of [28, Corollary 2.14] by setting[6] $\eta = \sqrt{\ln n/T}/G_\infty$ therein, but η in (26) is independent of horizon T. That is, Algorithm 2 is suitable to cases where T is not known *a priori*. Hence, it is applicable to real-time or life-long problems, or the games where the adversary decides the horizon. The regret (27) is even tighter than (26), and η is also free with respect to T.

IV. Unification of Adaptive Algorithms

FTBDL proposed in Section III provides a basis for the unification of different adaptive algorithms, and preliminarily, it have been showed that FTBDL unifies the typical ones, such as A-FOBOS, FTPRL, and A-RDA. In this section, we turn to more general adaptive algorithms, and a closer, wider, and deeper understanding of them under the proposed framework as follows.

1) The algorithms involved will be cleared up.
2) New observations on the connection within them will be discussed.
3) A simple adaptive algorithm will be proposed for the interesting case, where an augment adaptive matrix can be exploited.

A. Unification of Adaptive Algorithms

First, we reformulate FTBDL (10) to the following form:

$$w_{t+1} = \arg\min_{w \in \mathcal{F}} \sum_{\tau \in S_t} (B_{R_\tau}(w, v_\tau) + \eta\langle g_\tau, w\rangle) \tag{28}$$

where $R_\tau = (1/2)\|w\|_{A_\tau - A_{\tau-1}}^2$ (then $R_{1:t} = (1/2)\|w\|_{A_t}^2$) and $A_t \in S_{++}^n$. S_t excludes those rounds in which A_t is not updated, e.g., SOP [8] or NAROW [10] uses $S_t \neq [t]$. $v_t = w_t$ or $v_t = 0$, as discussed in Section III-C. In particular, let $A_t = A_t^{a_t, b}$ and

$$A_t^{a_t, b} = a_t I + \left(\sum_{\tau \in S_t} Q_\tau\right)^b \tag{29}$$

for some $a_t \geq 0, b \in \{(1/2), 1\}$, $Q_\tau \in S_+^n$, and $S_t \subseteq [t]$.

With (28) and (29) at hand, adaptive algorithms presented in Section II can be unified under the FTBDL (Table I).

1) A-FOBOS and FTPRL:

Proposition 7: A-FOBOS in (4) is equivalent to FTBDL in (28) on using $v_\tau = w_\tau$, $A_t = A_t^{a,b}$ and $R_\tau = (1/2)\|w\|_{A_\tau - A_{\tau-1}}^2$, where $a \geq 0$, $b = (1/2)$, $S_t = [t]$, and $Q_\tau = g_\tau g_\tau^\top$ (or Q_τ is diagonal and $(Q_\tau)_{i,i} = (g_\tau[i])^2$).

As we know, the adaptive (second order) information is captured by the second part of (29) and $a \geq 0$ is used to balance the identity matrix I and adaptive part. Without the adaptive part, A-FOBOS reduces to GD using a fixed step size. For $a = 0$, A_t captures the full adaptive information, which may become unstable, and for $a > 0$, A_t stays properly conditioned and A_t^{-1} can be calculated. In practice, optimal setting of a can be tuned by validation.

[6] There is a typo in [28]. $\eta = (\sqrt{\ln n}/G_\infty\sqrt{2T})$ yields the regret $(\sqrt{2} + (\sqrt{2}/2))G_\infty\sqrt{T \ln n}$, which is not optimal. Note that η involves horizon T.

TABLE I
UNIFICATION OF ADAPTIVE ALGORITHMS UNDER THE FTBDL FRAMEWORK

algorithm	Index	Loss	BD center	v_τ	A_t	Q_τ	R_τ	S_t	η	a_t
A-FOBOS[2]	1)	Convex	Last iteration	w_τ	$A_t^{a,1/2}$	$g_\tau g_\tau^\top$	$\frac{1}{2}\lVert w\rVert^2_{A_\tau - A_{\tau-1}}$	$[t]$	>0	$a \geq 0$
FTPRL[5]	1)	Convex	Last iteration	w_τ	$A_t^{a,1}$	$Q_1 \succ 0,\ Q_{\tau>1} \succeq 0$	$\frac{1}{2}\lVert w\rVert^2_{Q_\tau}$	$[t]$	1	0
ONS[15]	2)	Exp-concave	Last iteration	w_τ	$A_t^{a,1}$	$g_\tau g_\tau^\top$	$\frac{1}{2}\lVert w\rVert^2_{Q_\tau}$	$[t]$	r	0
FTAL[15]	2)	Exp-concave	Last iteration	w_τ	$A_t^{a,1}$	$g_\tau g_\tau^\top$	$\frac{1}{2}\lVert w\rVert^2_{Q_\tau}$	$[t]$	r	0
A-RDA[2]	3)	Convex	Origin	0	$A_t^{a,1/2}$	$g_\tau g_\tau^\top$	$\frac{1}{2}\lVert w\rVert^2_{A_\tau - A_{\tau-1}}$	$[t]$	>0	$a \geq G_2$
AODG[6], [7]	3)	Convex	Origin	0	$A_t^{a,1}$	$\sigma_\tau I,\ \sigma_1 > 0,\ \sigma_{\tau>1} \geq 0$	$\frac{1}{2}\lVert w\rVert^2_{Q_\tau}$	$[t]$	1	0
NAROW[10]	4)	Hinge	Origin	0	$\widehat{A}_{t+1}^{a,1}$	$x_\tau x_\tau^\top / r_\tau,\ r_t > 0$	$\frac{1}{2}\lVert w\rVert^2_{A_\tau - A_{\tau-1}}$	$\mathcal{M}_t \cup \mathcal{U}_t$	1	1
SOP[8]	4)	Hinge[7]	Origin	0	$\widehat{A}_{t+1}^{a_t,1}$	$x_\tau x_\tau^\top$	$\frac{1}{2}\lVert w\rVert^2_{A_\tau - A_{\tau-1}}$	$\mathcal{M}_t$	1	≥ 0

[7] SOP is based on Perceptron and here for unification we abuse hinge loss which is counted only when $\ell_t > 0$. Details see the texts.

Here, setting $b = 1/2$ aims to balance two terms in regret (16) and leads to a regret scaled in $O(\sqrt{T})$ for general loss. Actually, for A-FOBOS, regret (16) reduces to the following[8]:

$$\text{Regret}_T \leq \frac{a}{2\eta}\|\mathring{w}\|_2^2 + \frac{D_2^2}{2\eta}\text{Tr}\big(G_T^{1/2}\big) + \eta\text{Tr}\big(G_T^{1/2}\big)$$

where $G_t = \sum_{\tau \in S_t} Q_\tau$. Let $\mathcal{F} = \{w \mid \|w\|_2 \leq D_2/2\}$, then $\eta = D_2/\sqrt{2}$ gives us the optimal performance ($a = 0$), i.e., $\text{Regret}_T \leq \sqrt{2}D_2\text{Tr}(G_T^{1/2}) = \sqrt{2}D_2\sum_{i=1}^n \lambda_i^{1/2}$, where λ_i is the ith eigenvalue of G_T.

Note that, $G_T \in S_{++}^n$ and $\lambda_i > 0$, then

$$\left(\sum_{i=1}^n \lambda_i^{1/2}\right)^2 > \sum_{t=1}^n \lambda_i = \text{Tr}(G_T). \tag{30}$$

Hence, $\text{Tr}(G_T^{1/2}) > (\sum_{i=1}^n \sum_{t=}^T (g_t[i])^2)^{1/2} = (\sum_{t=1}^T \|g_t\|_2^2)^{1/2}$, and the upper bound of A-FOBOS is looser than that of FTPRL using adaptive coordinate-constant regularizer (3).

In general, suppose we consider the case that the feasible region is a L_p-ball, $\mathcal{F} = \{w \mid \|w\|_p \leq D_p/2\}$, $D_p > 0$. A-FOBOS does become more helpful for some feasible region, e.g., when $p = \infty$ [2]: this will be shown later in a special case using a diagonal adaptive matrix. A related and graceful analysis can be found in [5].

FTPRL in [5] is closely related to A-FOBOS, and similarly, it can be recovered under FTBDL.

Proposition 8: FTPRL in (2) is a specific case of FTBRL in (28) with $\eta = 1$, $v_\tau = w_\tau$, $A_t = A_t^{a,b}$, and $R_\tau = (1/2)\|w\|_{Q_\tau}$, where $a = 0$, $b = 1$, $Q_1 \in S_{++}^n$, $Q_{\tau>1} \in S_+^n$, and $S_t = [t]$.

Other than the adaptive coordinate-constant (3), a per-coordinate regularizer is helpful in some cases, e.g., when $\mathcal{F}$ is a L_∞-ball. In particular, let Q_τ be diagonal and

$$(Q_{1:t})_{i,i} = \frac{\sqrt{\sum_{\tau=1}^t (g_\tau[i])^2}}{D_\infty/\sqrt{2}}.$$

It is identical to A-FOBOS using diagonal matrix ($a = 0$) with $\eta = D_\infty/\sqrt{2}$. Then, regret (17) for FTPRL reduces to

$$\text{Regret}_T \leq \sqrt{2}D_\infty \sum_{i=1}^n \sqrt{\sum_{t=1}^T (g_t[i])^2}. \tag{31}$$

To cover a L_∞-ball feasible region, $D_2 = \sqrt{n}D_\infty$, and then for GD, its upper regret bound has to be $\sqrt{2n}D_\infty G_2\sqrt{T}$. Hence, the per-coordinate adaption outperforms GD. Actually, by Cauchy–Schwarz inequality, $\sum_{i=1}^n 1^2 \cdot \sum_{i=1}^n ((\sum_{t=1}^T (g_t[i])^2)^{1/2})^2 \geq (\sum_{t=1}^n 1 \cdot (\sum_{t=1}^T (g_t[i])^2)^{1/2})^2$, which means that $(n\sum_{t=1}^T \|g_t\|_2^2)^{1/2} \geq \sum_{i=1}^n (\sum_{t=1}^T (g_t[i])^2)^{1/2}$. In addition, for the coordinate-constant adaption in the case of interest, its upper regret bound is $\sqrt{2n}D_\infty(\sum_{t=1}^T \|g_t\|_2^2)^{1/2}$. Then, it is also looser than that of per-coordinate adaption. However, for a L_2-ball feasible region, the conclusion is opposite [similar to (30), $\sum_{i=1}^n (\sum_{t=1}^T (g_t[i])^2)^{1/2} \geq (\sum_{t=1}^T \|g_t\|_2^2)^{1/2}$].

2) ONS and FTAL: For a strongly convex loss function, a regret on the order of $O(\ln T)$ can be derived, even for GD. In addition, the magic is that, for general convex loss, e.g., linear loss, $\ell_t(w_t) - \ell_t(\mathring{w}) \leq \langle g_t, w_t - \mathring{w}\rangle$, but for σ-strongly convex loss, $\ell_t(w_t) - \ell_t(\mathring{w}) \leq \langle g_t, w_t - \mathring{w}\rangle - (\sigma/2)\|w_t - \mathring{w}\|_2^2$, where $w_t, \mathring{w} \in \mathcal{F}$. The extra nonpositive term improves the regret, and for $\eta_t = (\sigma t)^{-1}$, the upper bound of the regret for GD is $(G_2^2/\sigma)(1 + \ln T)$ [15].

As discussed in Section I, some widely used loss functions in learning problems are not strongly convex, but they may still obtain low regrets. In particular, for expconcave loss functions, a property analogous to that of strongly convex function can be exploited. That is, for $(1/r) = (1/2)\min\{(1/4D_2G_2), \varepsilon\}$

$$\ell_t(w_t) - \ell_t(\mathring{w}) \leq \langle g_t, w_t - \mathring{w}\rangle - \frac{1}{2r}\|\mathring{w} - w_t\|_{Q_t}^2 \tag{32}$$

where $\exp(-\varepsilon\ell_t(\cdot))$ is concave, $Q_t = g_t g_t^\top$, and $w_t, \mathring{w} \in \mathcal{F}$. Based on this, two algorithms, ONS and FTAL which enjoy the

[8] In regret (16), $\sum_{t=1}^T B_{R_t}(\mathring{w}, w_t) = (1/2)\|\mathring{w} - w_1\|_{A_1}^2 + (1/2)\sum_{t=2}^T \|\mathring{w} - w_t\|_{A_t - A_{t-1}}^2 \leq (a/2)\|\mathring{w} - w_1\|_2^2 + (1/2)\|\mathring{w} - w_1\|_2^2\text{Tr}(G_1^{1/2}) + (1/2)\sum_{t=2}^T \|\mathring{w} - w_t\|_2^2\text{Tr}(G_t^{1/2} - G_{t-1}^{1/2}) \leq (a/2)\|\mathring{w}\|_2^2 + (1/2)\max_t \|\mathring{w} - w_t\|_2^2\text{Tr}(G_T^{1/2})$, where we used $w_1 = 0$ for A-FOBOS and $\|w\|_A^2 \leq \|w\|_2^2\text{Tr}(A)$ for $A \in S_+^n$. By [2, Lemma 10], $\sum_{t=1}^T \|g_t\|_{R_{1:t}^*}^2 = \sum_{t=1}^T \|g_t\|_{A_t^{-1}}^2 \leq 2\text{Tr}(G_T^{1/2})$.

logarithmic regret, are developed in [15]. The two algorithms can also be unified under the proposed framework.

Proposition 9: FTAL in (8) can be recovered with FTBDL in (28) on setting $v_\tau = w_\tau$, $S_t = [t]$, $A_t = A_t^{a,b}$ and $R_\tau = (1/2)\|w\|_{Q_\tau}^2$, where $a = 0$, $b = 1$, $Q_\tau = g_\tau g_\tau^\top$, $\eta = r$ and $(1/r) = (1/2)\,\min\{(1/4D_2G_2), \varepsilon\}$.

From Proposition 9, FTAL is identical with FTPRL using $R_\tau = (1/2)\|w\|_{(1/r)Q_\tau}^2$, where $Q_\tau = g_\tau g_\tau^\top$. Furthermore, as shown in [15], ONS is equivalent to FTAL but with a more efficient implementation. Then, by Proposition 2, it can be recovered by MD. Formally, we state it as below.

Proposition 10: ONS in (9) can be recovered with FTBDL in (28) by setting $v_\tau = w_\tau$, $S_t = [t]$, $A_t = A_t^{a,b}$, and $R_\tau = (1/2)\|w\|_{Q_\tau}^2$, where $a = 0$, $b = 1$, $Q_\tau = g_\tau g_\tau^\top$, $\eta = r$, and $(1/r) = (1/2)\min\{(1/4D_2G_2), \varepsilon\}$. Equivalently, it can be updated by MD (15) [and then by its closed-form (14)] on using $v_\tau = w_\tau$, $R_{1:t} = (1/2)\|w\|_{A_t}^2$, and $A_t = A_t^{a,b}$, where $a = 0$, $b = 1$, $\eta = r$, and $Q_\tau = g_\tau g_\tau^\top$.

Thanks to the second term on the right-hand side of (32), a logarithmic regret as stated in [15] can also be derived from Proposition 3, with a minor extension to ONS, i.e., generalizing $a = 0$ to $a > 0$. In particular, the regret of ONS is upper bounded by $(aD^2/2r) + (r/2)n\ln((G^2/a)T + 1)$,[9] which is on the order of $O(\ln T)$ given a proper a.

3) A-RDA and AODG: Different to the above algorithms, A-RDA or AODG fixes $v_\tau = 0$, i.e., it is origin-centered.

Proposition 11: A-RDA in (5) is equivalent to FTBDL in (28) with $v_\tau = 0$, $R_\tau = (1/2)\|w\|_{A_\tau - A_{\tau-1}}^2$, and $A_t = A_t^{a,b}$, where $a \geq G_2$, $b = (1/2)$, $S_t = [t]$. Q_τ is identical with that of A-FOBOS.

As pointed out in Section II, AODG [6] in (13) is a simpler version of A-RDA, and in particular, $\eta = 1$, $R_\tau = (1/2)\|w\|_{Q_\tau}^2$, where $Q_\tau = \sigma_\tau I$. Then, it is also equivalent to FTBDL in (28), but with $\eta = 1$ and $A_t = A_t^{a,b}$, where $a = 0$, $b = 1$, and $Q_\tau = \sigma_\tau I$.

Omit the nonpositive term $-U/\eta$ in (21), the regret for A-RDA is upper bounded by[10]

$$\text{Regret}_T \leq \frac{a}{2\eta}\|\mathring{w}\|_2^2 + \frac{\|\mathring{w}\|_2^2}{2\eta}\text{Tr}\big(G_T^{1/2}\big) + \eta\text{Tr}\big(G_T^{1/2}\big) + O(1). \tag{33}$$

The above bound is tighter than that of A-FOBOS as $\|\mathring{w}\|_2^2 < D_2^2$. The advantage comes from the fact that, for A-FOBOS, the upper bound has to cover $(1/\eta)\sum_{t=1}^T \|\mathring{w} - w_t\|_{A_\tau - A_{\tau-1}}^2$, but A-RDA only needs to cover $(1/\eta)\sum_{t=1}^T \|\mathring{w}\|_{A_\tau - A_{\tau-1}}^2$. Here, the analysis is based on the upper bound, and interestingly, the practical aspects of A-RDA versus A-FOBOS also support the conclusion [2], [6].

4) NAROW and SOP: The two algorithms use input correlation matrix for adaption and are designed for classification. NAROW uses hinge loss and SOP is based on perceptron, and the two share similar adaptive matrix.

When $\mathcal{F} = \mathbb{R}^n$, $v_\tau = 0$, and $\eta = 1$, the closed-form for FTBDL in (28) reduces to $w_{t+1} = A_t^{-1}(-\sum_{\tau\in S_t} g_\tau)$. It is identical to NAROW in (7) in that $S_t = \mathcal{M}_t \cup \mathcal{U}_t$ and

$$A_t = I + \sum_{\tau\in S_t\cup\{t+1\}} \frac{x_\tau x_\tau^\top}{r_\tau}. \tag{34}$$

For hinge loss, $\ell_\tau(w) = \max\{0, 1 - y_\tau\langle w_\tau, x_\tau\rangle\}$, where y_τ is the label corresponding to x_τ. We observe that $g_\tau = -y_\tau x_\tau$, where $\ell_\tau(w_\tau) > 0$, and then, $x_\tau x_\tau^\top = g_\tau g_\tau^\top$. To unify the presentation with (29), let

$$\hat{A}_{t+1}^{a_t,b} = a_t I + \left(\sum_{\tau\in S_t} Q_\tau + Q_{t+1}\right)^b$$

then (34) can be represented as $A_t = \hat{A}_{t+1}^{1,1}$, where $Q_\tau = (x_\tau x_\tau^\top / r_\tau)$. Formally, we have the following results.

Proposition 12: NAROW in (7) is equivalent to FTBDL in (28) with $\mathcal{F} = \mathbb{R}^n$, $v_\tau = 0$, $R_\tau = (1/2)\|w\|_{A_\tau - A_{\tau-1}}^2$, and $A_t = \hat{A}_{t+1}^{1,1}$, where $S_t = \mathcal{M}_t \cup \mathcal{U}_t$, $Q_\tau = (x_\tau x_\tau^\top / r_\tau)$.

Note that $t + 1 \in S_{t+1}$ is not always true, and therefore, $\hat{A}_{t+1}^{1,1} \neq A_{t+1}^{1,1}$ in general. SOP in (6) can be viewed as a special case of NAROW, but $S_t = \mathcal{M}_t$, $Q_\tau = x_\tau x_\tau^\top$, and $A_t = \hat{A}_{t+1}^{a_t,1}$, where $a_t \geq 0$. SOP and NAROW are developed for classification and mistake bounds are considered. More details on the bounds and the parameters r_t or a_t can be found in the literature [8], [10].

B. Interesting Cases Where $g_\tau g_\tau^\top = x_\tau x_\tau^\top$

Besides hinge loss, for other losses, such as ϵ-insensitive loss $\ell_t(w) = \max\{0, |y_\tau - \langle w, x_\tau\rangle| - \epsilon\}$, where y_τ is the target output of x_τ, we also have[11] $g_\tau g_\tau^\top = x_\tau x_\tau^\top$ for $\ell_\tau > 0$. Hence, we can redefine A_t by including the current instance x_t into the prediction. As suggested in [8], this may lead to performance improvement in some scenarios.

In particular, consider that A-RDA in (5), and let $R_{1:t} = (1/2)\|w\|_{A_t}^2$ and[12] $A_t = (a^2 I + \sum_{\tau=1}^t g_\tau g_\tau^\top)^{1/2}$. Augment A_t as

$$\hat{A}_{t+1} = \big(A_t^2 + x_{t+1}x_{t+1}^\top\big)^{1/2}$$

and let $\widehat{R}_{1:t} = (1/2)\|w\|_{\hat{A}_{t+1}}^2$, then $\|g_t\|_{R_{1:t-1}^*}^2 = \|g_t\|_{A_{t-1}^{-1}}^2$ in regret (21) will be replaced by $\|g_t\|_{\widehat{R}_{1:t-1}^*}^2 = \|g_t\|_{\hat{A}_t^{-1}}^2$.

[9] In the case of interest, regret (16) becomes $\text{Regret}_T \leq (1/2r)\|\mathring{w} - w_1\|_{aI}^2 + (1/2r)\sum_{t=1}^T \|\mathring{w} - w_t\|_{Q_t}^2 + (r/2)\sum_{t=1}^T \|g_t\|_{A_t^{-1}}^2 - (1/2r)\sum_{t=1}^T \|\mathring{w} - w_t\|_{Q_t}^2 \leq (aD^2/2r) + (r/2)\sum_{t=1}^T \|g_t\|_{A_t^{-1}}^2$. Furthermore, [8, Lemma D.1] gives us that $\sum_{t=1}^T \|g_t\|_{A_t^{-1}}^2 \leq \ln(\det(A_T)/\det(aI))$. In addition, $\ln(\det(A_T)/\det(aI)) = \ln(\prod_{i=1}^n (\lambda_i + a/a)) = \sum_{i=1}^n \ln((\lambda_i/a) + 1) \leq n\ln((\lambda_{\max}/a) + 1) \leq n\ln((\text{Tr}(G_T)/a) + 1) \leq n\ln((G_2^2 T/a) + 1)$, where λ_i is the ith eigenvalue of $G_T = \sum_{\tau\in S_T} Q_\tau$ and $\lambda_{\max}$ is the largest one.

[10] In the case of interest, $R_{1:T}(\mathring{w}) = (1/2)\|\mathring{w}\|_{A_T}^2 \leq (1/2)\|\mathring{w}\|_2^2\text{Tr}(G_T^{1/2}) + (a/2)\|\mathring{w}\|_2^2$. By [2, Lemmas 9 and 10], $\sum_{t=1}^T \|g_t\|_{A_{t-1}^{-1}}^2 \leq 2\text{Tr}(G_T^{1/2})$. Note that $a \geq G_2$ is necessary to draw this conclusion.

[11] Two scenarios for $\ell_\tau > 0$: 1) $y_\tau - \langle w, x_\tau\rangle > \epsilon$, then $\ell_\tau = y_\tau - \langle w, x_\tau\rangle - \epsilon$ and $g_\tau = -x_\tau$ and 2) $y_\tau - \langle w, x_\tau\rangle < -\epsilon$, then $\ell_\tau = \langle w, x_\tau\rangle - y_\tau - \epsilon$ and $g_\tau = x_\tau$. Then, we have $g_\tau g_\tau^\top = x_\tau x_\tau^\top$ in both cases.

[12] The regret upper bound (33) is still suitable for this regularizer. Actually, in this case, $A_t \preceq aI + (\sum_{\tau=1}^t g_\tau g_\tau^\top)^{1/2}$, and then, the first two terms still hold. For the third term, $A_t^{-1} \preceq (\sum_{\tau=1}^{t+1} g_\tau g_\tau^\top)^{1/2}$ when $a \geq G_2$ and [2, Lemma 10] confirms the bound.

Algorithm 3 Simple Augment Adaptive Algorithm

```
Input: $\eta > 0$, $\sigma_{1:t} = \sqrt{\sum_{\tau=1}^{t} \|g_\tau\|_2^2 + \|x_{t+1}\|_2^2}$.
Initialize: $w_1 \in \mathcal{F}$.
for $t = 1, 2, \ldots$ do
    Suffer loss $\ell_t(w_t)$ and compute its subgradient $g_t$;
    $w_{t+1} = \arg\min_{w \in \mathcal{F}} \frac{\sigma_{1:t}}{2}\|w\|_2^2 + \eta\langle g_{1:t}, w\rangle$.
end for
```

As $\hat{A}_t \succeq A_{t-1}$, so $\hat{A}_t^{-1} \preceq A_{t-1}^{-1}$. In general, $\hat{A}_t^{-1} \neq A_{t-1}^{-1}$, and then, it gets an advantage on round t when $g_t \neq 0$, i.e., $\|g_t\|_{\hat{A}_t^{-1}}^2 < \|g_t\|_{A_{t-1}^{-1}}^2$. Actually, the Woodbury identity [21] gives us $(A_{t-1}^2)^{-1} = (1+x_t^\top(A_{t-1}^2)^{-1}x_t)(\hat{A}_t^2)^{-1}$, then $A_{t-1}^{-1} = (1 + x_t^\top(A_{t-1}^2)^{-1}x_t)^{1/2}\hat{A}_t^{-1}$, and we have

$$g_t^\top A_{t-1}^{-1} g_t = \sqrt{1 + x_t^\top\left(A_{t-1}^2\right)^{-1} x_t}\, g_t^\top \hat{A}_t^{-1} g_t.$$

Clearly, $(1 + x_t^\top(A_{t-1}^2)^{-1}x_t)^{1/2} > 1$ for $x_t \neq 0$.

In addition, when $g_t \neq 0$, $\hat{A}_t = (A_{t-1}^2 + g_t g_t^\top)^{1/2} = A_t$, and it is always true (it is trivial when $g_t = 0$) that $\|g_t\|_{\widehat{R}_{1:t-1}^*}^2 = \|g_t\|_{A_t^{-1}}^2$. Hence, the upper bound (33) still holds, but the assumption $a \geq G_2$ can be removed.

Furthermore, based on the observation, we develop a simple augment algorithm using diagonal adaptive matrix for losses that possess the property $g_\tau g_\tau^\top = x_\tau x_\tau^\top$. In particular, the algorithm is summarized in Algorithm 3.

Clearly, Algorithm 3 is a special case of FTBDL in that $v_\tau = 0$ and $R_\tau = (\sigma_\tau/2)\|w\|_2^2$. In addition, regret (21) leads to the following bound.

Proposition 13: The regret for Algorithm 3 with respect to $\mathring{w} \in \mathcal{F}$ is bounded by

$$\text{Regret}_T \leq \frac{\sigma_{1:T}}{2\eta}\|\mathring{w}\|_2^2 - \frac{1}{\eta}U + \frac{\eta}{2}\sum_{t=2}^{T}\frac{\|g_t\|_2^2}{\sigma_{1:t-1}} + O(1). \quad (35)$$

Let $\|\mathring{w}\|_2 \leq D_2/2$ and omit the nonpositive term $-U/\eta$, then setting $\eta = D_2/2\sqrt{2}$ yields

$$\text{Regret}_T \leq \frac{\sqrt{2}}{2}D_2\sqrt{\sum_{t=1}^{T}\|g_t\|_2^2} + O(1) \quad (36)$$

where $O(1)$ corresponds to the term $\langle g_1, w_1 - w_2\rangle$.

We can see that, for losses possessing the property $g_\tau g_\tau^\top = x_\tau x_\tau^\top$, Algorithm 3 gains a factor 1/2 over GD, and meanwhile, $(\sum_{t=1}^{T}\|g_\tau\|_2^2)^{1/2} \leq G_2\sqrt{T}$. In addition, regret (36) gains a factor 1/2 over FTPRL using a coordinate-constant regularizer (3).

In general, based on the above analysis, more adaptive algorithms can be developed in specific scenarios. One may choose different values of v_τs, use the original version of A_t or its augment as Algorithm 3 does, set different a_t, b, or S_t, or more generally, use different BDs as Algorithm 2 does. In addition, it will be interesting to extend adaptive learning to kernel-based online learning in a general way.

V. General Matrix Equation for Kernelization

In this section, we derive a general matrix equation to extend adaptive learning to nonlinear learning with kernels. We first transform adaptive learning to the form that involves inner product $\langle x_i, x_j\rangle$ and, then, generalize it by replacing the inner product with kernel $k(x_i, x_j)$ as has been used in [8].

Without loss of generality,[13] consider A-RDA type online learning with $\mathcal{F} = \mathbb{R}^n$ and the closed-form for FTBDL in (28) reduces to[14]

$$w_{t+1} = -\eta A_t^{-1}\left(\sum_{\tau \in S_t} g_\tau\right) = -\eta A_t^{-1} g_{\sim S_t}. \quad (37)$$

Here, the adaptive matrix A_t involves $x_\tau x_\tau^\top$ or $g_\tau g_\tau^\top$, and we have to transform update (37) into some form which can be represented by inner products.

A. Matrix Equations

First, we restate a matrix equation, which has been used in [8] for kernelizing SOP.

Proposition 14 [21]: Let $A \in \mathbb{R}^{n\times m}$ and $B = A^\top$, then

$$B(aI_n + AB)^{-1} = (aI_m + BA)^{-1}B. \quad (38)$$

Now, we present a lemma that will be used later.

Lemma 1: Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$, and $C \in \mathbb{R}^{m\times m}$. Assume that $BA = CB$, then for any $d \in \mathbb{N}^+$, we have

$$BA^d = C^d B. \quad (39)$$

This can be established by induction. By the assumption, (39) holds when $d = 1$. We assume $BA^{d-1} = C^{d-1}B$, then

$$BA^d = BA^{d-1}A = C^{d-1}BA = C^{d-1}CB = C^d B.$$

The following is the key equation that supports the transformation and, then, the kernelization.

Proposition 15: Let $A \in \mathbb{R}^{n\times m}$ and $B = A^\top$, then

$$B(aI_n + AB)^{-\frac{1}{2}} = (aI_m + BA)^{-\frac{1}{2}}B. \quad (40)$$

The proof is included in Appendix E.

More generally, by Lemma 1, $BA = CB$ implies $Bp(A) = p(C)B$, where $p(\cdot)$ denotes a polynomial.

B. Kernelization

The above propositions make it possible to transform update (37) to be of interest. Let $g_t = [g_\tau]$, $\tau \in S_t$, then $g_{\sim S_t}^\top = 1^\top g_t^\top$ (g_τ can be replaced by x_τ or $(x_\tau/\sqrt{r_\tau})$ as used in SOP or NAROW). When $A_t = aI_n + g_t g_t^\top$. Proposition 14 gives us

$$\begin{aligned} w_{t+1}^\top &= -\eta 1^\top g_t^\top\left(aI_n + g_t g_t^\top\right)^{-1} \\ &= -\eta 1^\top\left(aI_{|S_t|} + g_t^\top g_t\right)^{-1} g_t^\top \\ &=: -\eta 1^\top \mathbb{A}_t^{-1} g_t^\top \end{aligned} \quad (41)$$

[13] It is possible to kernelize adaptive MD (i.e., $v_\tau = w_\tau$), but it does not provide more insights and considering the limit of space we omit it.

[14] By (12), and the closed-form of the update for (28) reduces to $w_{t+1} = \nabla R_{1:t}^*(\sum_{\tau=1}^{t}(\nabla R_\tau(v_\tau) - \eta g_\tau)) = \nabla R_{1:t}^*(-\eta\sum_{\tau\in S_t} g_\tau)$. Then, the fact that $\nabla R_{1:t}^*(w) = A_t^{-1}w$ leads to (37).

Algorithm 4 Simple Adaptive Algorithm With Kernels

```
Input: $\eta > 0$, $\sigma_{1:t} = \sqrt{\sum_{\tau=1}^{t} \|g_\tau\|_2^2}$.
Initialize: $w_1 \in \mathcal{H}$.
for $t = 1, 2, \ldots$ do
    Suffer loss $\ell_t(w_t)$ and compute its subgradient $g_t$;
    $w_{t+1} = \arg\min_{w \in \mathcal{H}} \frac{\sigma_{1:t}}{2} \|w - w_t\|_2^2 + \eta \langle g_t, w \rangle$.
end for
```

where $g_t^\top g_t$ only involves inner product $\langle g_i, g_j \rangle$. Note that for commonly used losses, $g_\tau = \dot{g}_\tau x_\tau$, where $\dot{g}_\tau$ is a scalar.[15] Let $x_t = [x_\tau]$, $\dot{g}_t = [\dot{g}_\tau]$, and $D(\dot{g}_t)$ be the diagonal matrix with diagonal elements $\dot{g}_t$. Then, $g_t = x_t D(\dot{g}_t)$ and $\mathbb{A}_t = a I_{|S_t|} + D(\dot{g}_t) K_t D(\dot{g}_t)$. Here, $K_t := x_t^\top x_t$ is the corresponding Gram matrix. Clearly, $K_t(\tau, \tau) = x_\tau^\top x_\tau$.

As pointed out before, $A_t = (a^2 I_n + g_t g_t^\top)^{1/2}$ makes no change for the regret analysis. In this case, Proposition 15 gives us

$$\begin{aligned} w_{t+1}^\top &= -\eta 1^\top \left(a^2 I_{|S_t|} + g_t^\top g_t\right)^{-\frac{1}{2}} g_t^\top \\ &=: -\eta 1^\top \mathbb{A}_t^{-1} g_t^\top . \end{aligned} \tag{42}$$

Using Sherman–Morrison–Woodbury formula [21] (or the matrix inversion lemma [44]), $\mathbb{A}_t^{-1}$ can be updated recursively and costs us $O(|S_t|^2)$ each round. It is common to use the diagonal version [2], [5], replacing $a I_{|S_t|} + g_t^\top g_t$ (or $a^2 I_{|S_t|} + g_t^\top g_t$) with its diagonal matrix reduces the cost to $O(|S_t|)$, which is on the same order of nonadaptive online learning. In particular, let $c_\tau = \dot{g}_\tau / (a + \dot{g}_\tau^2 K_t(\tau, \tau))$ (or $c_\tau = \dot{g}_\tau / (a^2 + \dot{g}_\tau^2 K_t(\tau, \tau))^{1/2}$ in case where square root operation is used), then (41) [or (42)] reduces to

$$w_{t+1}^\top = -\eta \sum_{\tau \in S_t} c_\tau x_\tau^\top =: \sum_{\tau \in S_t} \alpha_\tau x_\tau^\top \tag{43}$$

whose computation cost scales on the same order as that of the nonadaptive online learning.

Alternatively, it is possible to use inequality (18) to get a simple adaptive version with kernels with improvable guarantee. In particular, let $R_\tau = (\sigma_\tau/2)\|w\|_2^2$, $v_\tau = w_t$ and FTBDL reduces to Algorithm 4 ($\sigma_{1:t}$ only involves inner product). Its closed-form update is equal to that of GD, but with an adaptive stepsize, $\eta_t = (\eta/\sigma_{1:t})$, and its regret upper bound is improved to $\sqrt{2} D_2 (\sum_{t=1}^{T} \|g_t\|_2^2)^{1/2}$.

C. Other Topics

A bottleneck for kernelized online learning is that the support set may keep increasing with learning. Therefore, budget strategies, which keep $|S_t|$ under control [45]–[50], or sparse update methods [51], [52], which perform random update, can be incorporated with adaptive online learning. An alternative way to kernelization is using the duality between kernels and random processes [53], [54] to approximate the kernel with the inner products of m randomized features. Then, adaptive online learning can be performed on the m-dimensional randomized features. In fact, by applying this approximating idea to the kernel least mean square algorithm, Singh *et al.* [55] show a constant computational complexity with no observable loss in performance. Explicitly, mapping the input onto the complex feature space (Euler representation) is another interesting clue for keeping the kernel under control, as shown in [39], where the authors successfully introduce a robust incremental PCA called Euler-PCA.

VI. Conclusion

This paper proposed a framework, FTBDL, which covers most popular adaptive algorithms that use the second-order information. With the proposed framework, a deep unification of existing algorithms is presented, and some new insights are revealed. Several new simple adaptive algorithms with improvable guarantee are developed. Furthermore, this paper derived a matrix equation that provides a general way to extend adaptive online linear learning to nonlinear cases via kernelization. Then, a simple adaptive algorithm applicable to kernelized online learning is presented.

Developing new algorithms under the proposed framework, specifically using other forms of BDs, such as KL divergence, is interesting. Parameter free algorithms, such as V-SGD in [18], but with clear regret, guarantees deserve further study.

Appendix

A. Proof of Proposition 1

Proof: Let $\widetilde{w}_{t+1} = \arg\min_{w \in \mathbb{R}^n} \ell_{1:t}^{R_\tau}(w)$, and $w'_{t+1} = \Pi_{\ell_{1:t}^{R_\tau}, \mathcal{F}}(\widetilde{w}_{t+1})$. By definition, we have

$$\ell_{1:t}^{R_\tau}(w_{t+1}) \leq \ell_{1:t}^{R_\tau}\left(w'_{t+1}\right).$$

In addition, $\nabla \ell_{1:t}^{R_\tau}(\widetilde{w}_{t+1}) = 0$ as $\widetilde{w}_{t+1}$ minimizes $\ell_{1:t}^{R_\tau}$ over $\mathbb{R}^n$. Then

$$\ell_{1:t}^{R_\tau}\left(w'_{t+1}\right) - \ell_{1:t}^{R_\tau}(\widetilde{w}_{t+1}) = B_{\ell_{1:t}^{R_\tau}}\left(w'_{t+1}, \widetilde{w}_{t+1}\right).$$

Furthermore

$$\begin{aligned} B_{\ell_{1:t}^{R_\tau}}\left(w'_{t+1}, \widetilde{w}_{t+1}\right) &\leq B_{\ell_{1:t}^{R_\tau}}(w_{t+1}, \widetilde{w}_{t+1}) \\ &= \ell_{1:t}^{R_\tau}(w_{t+1}) - \ell_{1:t}^{R_\tau}(\widetilde{w}_{t+1}). \end{aligned}$$

In addition, we have $\ell_{1:t}^{R_\tau}(w'_{t+1}) \leq \ell_{1:t}^{R_\tau}(w_{t+1})$. By the assumption of strongly convexity of R, the BD is strictly convex with respect to its first argument. Thus, $\ell_{1:t}^{R_\tau}$ is strictly convex, and we have $w_{t+1} = w'_{t+1}$.

For the second equivalence, recalling the properties 1) and 2) of BD presented in Section III-A, we have

$$\begin{aligned} B_{\ell_{1:t}^{R_\tau}}(u, v) &= \sum_{\tau=1}^{t} B_{\ell_\tau^{R_\tau}}(u, v) = \sum_{\tau=1}^{t} B_{B_{R_\tau}(\cdot, v_\tau)}(u, v) \\ &= \sum_{\tau=1}^{t} \{B_{R_\tau}(u, v_\tau) - B_{R_\tau}(v, v_\tau) \\ &\qquad - \langle \nabla B_{R_\tau}(v, v_\tau), u - v \rangle\} \end{aligned}$$

[15] Examples: for hinge loss $\ell_t(w_t) = \max\{0, 1 - y_t \langle w_t, x_t \rangle\}$, consider the case when $\ell_t(w_t) > 0$, and we have $g_t = -y_t x_t$; for square loss $\ell_t(w_t) = (1/2)(y_t - \langle w_t, x_t \rangle)^2$, we have $g_t = -(y_t - \langle w_t, x_t \rangle) x_t$.

where $\nabla B_{R_\tau}(u, v)$ is the derivative of $B_{R_\tau}(u, v)$ with respect to the first argument u. Then, the three-point equality of BD presented in Section III-A completes the proof

$$B_{\ell^{R_\tau}_{1:t}}(u, v) = \sum_{\tau=1}^{t} B_{R_\tau}(u, v) = B_{R_{1:t}}(u, v).$$

■

B. Proof of Proposition 2

Proof: By Proposition 1, we only need to prove the equivalency in the case of no projection. That is, we aim to prove that the following two are equivalent:

$$\widehat{w}_{t+1} = \arg\min_{w\in\mathbb{R}^n} \{B_{R_{1:t}}(w, \widehat{w}_t) + \eta\langle g_t, w\rangle\} \tag{44}$$

$$\widetilde{w}_{t+1} = \arg\min_{w\in\mathbb{R}^n} \sum_{\tau=1}^{t} (B_{R_\tau}(w, \widetilde{w}_\tau) + \eta\langle g_\tau, w\rangle). \tag{45}$$

The proof is done by induction, and the convexity of the two objective functions is used.

We start with $\widehat{w}_1 = \widetilde{w}_1$ and assume that $\widehat{w}_t = \widetilde{w}_t$. By the optimality for (45), the gradient of the objective function with respect to $\widetilde{w}_t$ is zero, that is, $\sum_{\tau=1}^{t-1}(\nabla R_\tau(\widetilde{w}_t) - \nabla R_\tau(\widetilde{w}_\tau) + \eta g_\tau) = 0$. Then, we have

$$\sum_{\tau=1}^{t-1} \nabla R_\tau(\widetilde{w}_t) = -\eta g_{1:t-1} + \sum_{\tau=1}^{t-1} \nabla R_\tau(\widetilde{w}_\tau). \tag{46}$$

On the other hand, by (44)

$$\begin{aligned}
\widehat{w}_{t+1} &= \arg\min_{w\in\mathcal{R}^n} \sum_{\tau=1}^{t} B_{R_\tau}(w, \widehat{w}_t) + \eta\langle g_t, w\rangle \\
&= \arg\min_{w\in\mathcal{R}^n} \sum_{\tau=1}^{t} (R_\tau(w) - R_\tau(\widehat{w}_t) \\
&\qquad - \langle \nabla R_\tau(\widehat{w}_t), w - \widehat{w}_t\rangle) + \eta\langle g_t, w\rangle \\
&= \arg\min_{w\in\mathcal{R}^n} \sum_{\tau=1}^{t} (R_\tau(w) - \langle \nabla R_\tau(\widehat{w}_t), w\rangle) + \eta\langle g_t, w\rangle \\
&= \arg\min_{w\in\mathcal{R}^n} \sum_{\tau=1}^{t} R_\tau(w) - \sum_{\tau=1}^{t-1} \langle \nabla R_\tau(\widehat{w}_t), w\rangle \\
&\quad - \langle \nabla R_t(\widehat{w}_t), w\rangle + \eta\langle g_t, w\rangle.
\end{aligned}$$

Replacing $\widehat{w}_t$ in the above equation with $\widetilde{w}_t$ and using (46)

$$\begin{aligned}
\widehat{w}_{t+1} &= \arg\min_{w\in\mathcal{R}^n} \sum_{\tau=1}^{t} R_\tau(w) - \langle \nabla R_t(\widetilde{w}_t), w\rangle + \eta\langle g_t, w\rangle \\
&\qquad - \left\langle -\eta g_{1:t-1} + \sum_{\tau=1}^{t-1} \nabla R_\tau(\widetilde{w}_\tau), w \right\rangle \\
&= \arg\min_{w\in\mathcal{R}^n} \sum_{\tau=1}^{t} (R_\tau(w) - \langle \nabla R_\tau(\widetilde{w}_\tau), w\rangle + \eta\langle g_\tau, w\rangle).
\end{aligned}$$

That is

$$\begin{aligned}
\widehat{w}_{t+1} &= \arg\min_{w\in\mathcal{R}^n} \sum_{\tau=1}^{t} (R_\tau(w) - R_\tau(\widetilde{w}_\tau) + \eta\langle g_\tau, w\rangle \\
&\qquad - \langle \nabla R_\tau(\widetilde{w}_\tau), w - \widetilde{w}_\tau\rangle) \\
&= \arg\min_{w\in\mathcal{R}^n} \sum_{\tau=1}^{t} (B_{R_\tau}(w, \widetilde{w}_\tau) + \eta\langle g_\tau, w\rangle) \\
&= \widetilde{w}_{t+1}.
\end{aligned}$$

It completes the proof. ■

C. Proof of Proposition 3

Proof: Due to the equivalence of MD and FTBDL in Proposition 2, it is sufficient to prove regret (16) for MD (15).

By the derivation of [42, Th. 4.1], i.e., (4.21) therein, for any $\mathring{w} \in \mathcal{F}$, we have $\ell_t(w_t) - \ell_t(\mathring{w}) \le (\triangle_t/\eta) + (\eta/2)\|g_t\|^2_{R^*_{1:t}}$, where $\triangle_t := B_{R_{1:t}}(\mathring{w}, w_t) - B_{R_{1:t}}(\mathring{w}, w_{t+1})$. Summing two sides of the inequality, the regret can be upper bounded by

$$\sum_{t=1}^{T} \frac{\triangle_t}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|g_t\|^2_{R^*_{1:t}} =: RS1 + \frac{\eta}{2}\sum_{t=1}^{T} \|g_t\|^2_{R^*_{1:t}}.$$

Furthermore, by the additivity and nonnegativity of BD

$$RS1 \le \frac{1}{\eta} B_{R_1}(\mathring{w}, w_1) + \frac{1}{\eta}\sum_{t=1}^{T-1} \hat{\triangle}_{t+1} = \frac{1}{\eta}\sum_{t=1}^{T} B_{R_t}(\mathring{w}, w_t)$$

where $\hat{\triangle}_{t+1} := B_{R_{1:t+1}}(\mathring{w}, w_{t+1}) - B_{R_{1:t}}(\mathring{w}, w_{t+1})$. It completes the proof. ■

D. Proof of Proposition 4

Proof: Let $\widetilde{R}_t := (1/\eta)R_{1:t}(w)$ and rewrite (20) as

$$w_{t+1} = \arg\min_{w\in\mathcal{F}} \{\widetilde{R}_t(w) + \langle g_{1:t}, w\rangle\}. \tag{47}$$

Recall that $\widetilde{R}^*_t(\theta) = \sup_{w\in\mathcal{F}}(\langle\theta, w\rangle - \widetilde{R}_t(w))$, and we have

$$\begin{aligned}
\text{Regret}_T &\le \sum_{t=1}^{T} \langle g_t, w_t - \mathring{w}\rangle - \widetilde{R}_T(\mathring{w}) + \widetilde{R}_T(\mathring{w}) \\
&\le \sum_{t=1}^{T} \langle g_t, w_t\rangle + \sup_{w\in\mathcal{F}} \{-\langle g_{1:T}, w\rangle - \widetilde{R}_T(w)\} \\
&\quad + \widetilde{R}_T(\mathring{w}) \\
&= \widetilde{R}_T(\mathring{w}) + \sum_{t=1}^{T} \langle g_t, w_t\rangle + \widetilde{R}^*_T(-g_{1:T}).
\end{aligned}$$

By the optimality in (47), we have

$$\begin{aligned}
\widetilde{R}^*_T(-g_{1:T}) &= -\langle g_{1:T}, w_{T+1}\rangle - \widetilde{R}_T(w_{T+1}) \\
&= -\langle g_{1:T}, w_{T+1}\rangle - \widetilde{R}_{T-1}(w_{T+1}) - \frac{1}{\eta} R_T(w_{T+1}) \\
&\le \sup_{w\in\mathcal{F}} \{-\langle g_{1:T}, w\rangle - \widetilde{R}_{T-1}(w)\} - \frac{1}{\eta} R_T(w_{T+1}) \\
&= \widetilde{R}^*_{T-1}(-g_{1:T}) - \frac{1}{\eta} R_T(w_{T+1}).
\end{aligned}$$

Note that R_τ is strongly convex and $\widetilde{R}_t$ is $1/\eta$-strongly convex. Then, $\widetilde{R}_t^*(\theta)$ has η-Lipschitz continuous gradients and

$$\left\| \nabla \widetilde{R}_t^*(\theta_1) - \nabla \widetilde{R}_t^*(\theta_2) \right\|_{R_{1:t}} \leq \eta \|\theta_1 - \theta_2\|_{R_{1:t}^*}.$$

Due to the duality of strongly convex and strongly smooth functions [28, Lemma 2.19], we have

$$\widetilde{R}_t^*(\theta_1) \leq \widetilde{R}_t^*(\theta_2) + \left\langle \nabla \widetilde{R}_t^*(\theta_2), \theta_1 - \theta_2 \right\rangle + \frac{\eta}{2} \|\theta_1 - \theta_2\|_{R_{1:t}^*}^2. \tag{48}$$

And, for the gradient of the CD [31, Lemma 2]

$$\nabla \widetilde{R}_t^*(\theta) = \arg\min_{w \in \mathcal{F}} \{ -\langle \theta, w \rangle + \widetilde{R}_t(w) \}. \tag{49}$$

Then, due to (48)

$$\begin{aligned}\widetilde{R}_{T-1}^*(-g_{1:T}) \leq{}& \widetilde{R}_{T-1}^*(-g_{1:T-1}) - \left\langle \nabla \widetilde{R}_{T-1}^*(-g_{1:T-1}), g_T \right\rangle \\ &+ \frac{\eta}{2} \|g_T\|_{R_{1:T-1}^*}^2.\end{aligned}$$

Furthermore, (49) and the optimality of (47) show that

$$\left\langle \nabla \widetilde{R}_{T-1}^*(-g_{1:T-1}), g_T \right\rangle = \langle w_T, g_T \rangle.$$

Hence, we have

$$\begin{aligned}\text{Regret}_T \leq{}& \widetilde{R}_T(\mathring{w}) - \frac{1}{\eta} R_T(w_{T+1}) + \frac{\eta}{2} \|g_T\|_{R_{1:T-1}^*}^2 \\ &+ \sum_{t=1}^{T-1} \langle g_t, w_t \rangle + \widetilde{R}_{T-1}^*(-g_{1:T-1}).\end{aligned}$$

Repeat this process $T - 1$ times (on time index t), and we have

$$\begin{aligned}\text{Regret}_T \leq{}& \widetilde{R}_T(\mathring{w}) - \frac{U'}{\eta} + \sum_{t=2}^{T} \frac{\eta}{2} \|g_t\|_{R_{1:t-1}^*}^2 \\ &+ \langle g_1, w_1 \rangle + \widetilde{R}_1^*(-g_1)\end{aligned}$$

where $U' = \sum_{t=2}^T R_t(w_{t+1})$. Using the fact that $\widetilde{R}_1^*(-g_1) = -\langle g_1, w_2 \rangle - (1/\eta) R_1(w_2)$, we complete the proof. ■

E. Proof of Proposition 15

Proof: It is equivalent to prove $(aI_m + BA)^{1/2} B = B(aI_n + AB)^{1/2}$. For simplicity, let $\hat{A}_n := aI_n + BA$ and $\hat{A}_m := aI_m + AB$. As we know AB and BA share the same nonzero eigenvalues, then $\delta(\hat{A}_n) = \delta(\hat{A}_m)$, where $\delta(\cdot)$ is the set of all the different eigenvalues. Let $s \leq \min\{n, m\}$ be the number of different eigenvalues and denote this set by $\{\lambda_1, \ldots, \lambda_s\}$, where $\lambda_i \neq \lambda_j$ for $i \neq j$. Moreover, assume that there is a function $h(\cdot) = (\cdot)^{1/2}$, and let $D(\cdot)$ be the diagonal matrix with diagonal elements $(\cdot)$.

$\hat{A}_n$ is positive definite, and it has a unique square root

$$(\hat{A}_n)^{\frac{1}{2}} = P_1 D\left(\lambda_{i_1}^{\frac{1}{2}}, \ldots, \lambda_{i_n}^{\frac{1}{2}}\right) P_1^{-1}$$

where P_1 is an orthogonal matrix, and for all $k \in [n]$, $\lambda_{i_k} \in \{\lambda_1, \ldots, \lambda_s\}$ and $\{\lambda_{i_1}, \ldots, \lambda_{i_n}\} \supseteq \{\lambda_1, \ldots, \lambda_s\}$. Similarly, we have

$$(\hat{A}_m)^{\frac{1}{2}} = P_2 D\left(\lambda_{j_1}^{\frac{1}{2}}, \ldots, \lambda_{j_m}^{\frac{1}{2}}\right) P_2^{-1}$$

where P_2 is an orthogonal matrix, and for all $k \in [m]$, $\lambda_{i_k} \in \{\lambda_1, \ldots, \lambda_s\}$ and $\{\lambda_{j_1}, \ldots, \lambda_{j_m}\} \supseteq \{\lambda_1, \ldots, \lambda_s\}$. That is

$$h(\hat{A}_n) = (\hat{A}_n)^{1/2} = P_1 D(h(\lambda_{i_1}), \ldots, h(\lambda_{i_n})) P_1^{-1}$$

and

$$h(\hat{A}_m) = (\hat{A}_m)^{\frac{1}{2}} = P_2 D(h(\lambda_{j_1}), \ldots, h(\lambda_{j_m})) P_2^{-1}.$$

Therefore

$$\hat{A}_n = P_1 D(\lambda_{i_1}, \ldots, \lambda_{i_n}) P_1^{-1}, \ \hat{A}_m = P_2 D(\lambda_{j_1}, \ldots, \lambda_{j_m}) P_2^{-1}.$$

They are all diagonalizable matrices.

Let P_{s-1} be the polynomial of degree $s - 1$. Then, by the properties of diagonalizable matrix [56, Ch. 3.1, Th. 3], there exists a polynomial function $p(\lambda) \in P_{s-1}$, such that

$$p(\lambda_j) = h(\lambda_j), \quad j = 1, 2, \ldots, s$$

and

$$p(\hat{A}_n) = h(\hat{A}_n), \quad p(\hat{A}_m) = h(\hat{A}_m).$$

The function $h(\cdot)$ is unrelated to the choice of P_1 or P_2, and so does $p(\cdot)$. In particular, recall here $h(\cdot) = (\cdot)^{1/2}$.

By Proposition 14, we have $B\hat{A}_n = \hat{A}_m B$, and Lemma 1 gives us $Bp(\hat{A}_n) = p(\hat{A}_m)B$. Then, $B(\hat{A}_n)^{1/2} = (\hat{A}_m)^{1/2} B$, and the result follows. ■

References

[1] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 928–935.

[2] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[3] J. C. Duchi and Y. Singer, “Efficient online and batch learning using forward backward splitting,” *J. Mach. Learn. Res.*, vol. 10, pp. 2899–2934, Dec. 2009.

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

[5] H. B. McMahan and M. Streeter, “Adaptive bound optimization for online convex optimization,” in *Proc. 23th Conf. Learn. Theory*, 2010, pp. 244–256.

[6] H. B. McMahan, “Follow-the-regularized-leader and mirror descent: Equivalence theorems and $L1$ regularization,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 525–533.

[7] P. L. Bartlett, E. Hazan, and A. Rakhlin, “Adaptive online gradient descent,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20. 2007, pp. 65–72.

[8] N. Cesa-Bianchi, A. Conconi, and C. Gentile, “A second-order Perceptron algorithm,” *SIAM J. Comput.*, vol. 34, no. 3, pp. 640–668, 2005.

[9] K. Crammer, A. Kulesza, and M. Dredze, “Adaptive regularization of weight vectors,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 22. 2009, pp. 414–422.

[10] F. Orabona and K. Crammer, “New adaptive algorithms for online classification,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23. 2010, pp. 1840–1848.

[11] J. Wang, P. Zhao, and S. C. H. Hoi, “Exact soft confidence-weighted learning,” in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1–8.

[12] T. M. Cover, “Universal portfolios,” *Math. Finance*, vol. 1, no. 1, pp. 1–29, 1991.

[13] J. Kivinen and M. K. Warmuth, “Relative loss bounds for multidimensional regression problems,” *Mach. Learn.*, vol. 45, no. 3, pp. 301–329, 2001.

[14] V. Vovk, “Competitive on-line statistics,” *Int. Statist. Rev.*, vol. 69, no. 2, pp. 213–248, 2001.

[15] E. Hazan, A. Kalai, S. Kale, and A. Agarwal, “Logarithmic regret algorithms for online convex optimization,” in *Proc. 19th Annu. Conf. Learn. Theory*, 2006, pp. 499–513.

[16] F. Orabona, N. Cesa-Bianchi, and C. Gentile, "Beyond logarithmic bounds in online learning," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, vol. 22. 2012, pp. 823–831.

[17] S. Ross, P. Mineiro, and J. Langford, "Normalized online learning," in *Proc. 29th Conf. Uncertainty Artif. Intell. (UAI)*, Aug. 2013.

[18] T. Schaul, S. Zhang, and Y. LeCun, "No more pesky learning rates," in *Proc. 30th Int. Conf. Mach. Learn. Cycle 3*, 2013, pp. 343–351.

[19] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[20] B. Schölkopf and A. J. Smola, *Learning With Kernels—Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[21] K. B. Petersen and M. S. Pedersen. (2012). *The Matrix Cookbook*. [Online]. Available: http://matrixcookbook.com, accessed Nov. 2012.

[22] K. Crammer, M. Dredze, and F. Pereira, "Exact convex confidence-weighted learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 21. 2008, pp. 345–352.

[23] N. N. Schraudolph, "Local gain adaptation in stochastic gradient descent," in *Proc. 9th Int. Conf. Artif. Neural Netw.*, 1999, pp. 569–574.

[24] S. V. N. Vishwanathan, N. N. Schraudolph, and A. J. Smola, "Step size adaptation in reproducing kernel Hilbert space," *J. Mach. Learn. Res.*, vol. 7, pp. 1107–1133, Dec. 2006.

[25] W. He, "Limited stochastic meta-descent for kernel-based online learning," *Neural Comput.*, vol. 21, no. 9, pp. 2667–2686, 2009.

[26] W. He, J. T. Kwok, J. Zhu, and Y. Liu. (Jun. 2015). *A Note on the Unification of Adaptive Online Learning*. [Online]. Available: http://www.umich.edu/~youngliu/pub/online_full.pdf, accessed Nov. 2015.

[27] A. Rakhlin. (Jan. 2009). *Lecture Notes on Online Learning*. [Online]. Available: http://www-stat.wharton.upenn.edu/~rakhlin/, accessed Apr. 2009.

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

[29] A. Rakhlin and K. Sridharan. (Aug. 2012). *Lecture Notes on Statistical Learning Theory and Sequential Prediction*. http://www-stat.wharton.upenn.edu/~rakhlin/, accessed Aug. 2012.

[30] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 3, pp. 200–217, 1967.

[31] S. Shalev-Shwartz and Y. Singer, "Logarithmic regret algorithms for strongly convex repeated games," Hebrew Univ., Tech. Rep. 2007-42, May 2007.

[32] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006, pp. 294–298.

[33] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Inf. Comput.*, vol. 132, no. 1, pp. 1–63, 1997.

[34] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, Dec. 2005.

[35] R. Iyer and J. Bilmes, "Submodular-Bregman and the Lovász–Bregman divergences with applications," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25. 2012, pp. 2942–2950.

[36] H. Wang and A. Banerjee, "Bregman alternating direction method of multipliers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2816–2824.

[37] J. Abernethy, E. Hazan, and A. Rakhlin, "Competing in the dark: An efficient algorithm for bandit linear optimization," in *Proc. Conf. Learn. Theory*, 2008, pp. 263–274.

[38] B. Kulis, M. A. Sustik, and I. S. Dhillon, "Low-rank kernel learning with Bregman matrix divergences," *J. Mach. Learn. Res.*, vol. 10, pp. 341–376, Dec. 2009.

[39] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Euler principal component analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 498–518, 2013.

[40] M. K. Warmuth and D. Kuzmin, "Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension," *J. Mach. Learn. Res.*, vol. 9, no. 10, pp. 2287–2320, 2008.

[41] M. Collins, S. Dasgupta, and R. E. Schapire, "A generalization of principal components analysis to the exponential family," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 617–624.

[42] 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.

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

[44] L. Csató and M. Opper, "Sparse on-line Gaussian processes," *Neural Comput.*, vol. 14, no. 3, pp. 641–668, Mar. 2002.

[45] K. Crammer, J. Kandola, and Y. Singer, "Online classification on a budget," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16. 2003, pp. 225–232.

[46] N. Cesa-Bianchi and C. Gentile, "Tracking the best hyperplane with a simple budget Perceptron," in *Learning Theory*. Springer, 2006, pp. 483–498.

[47] O. Dekel, S. Shalev-Shwartz, and Y. Singer, "The Forgetron: A kernel-based Perceptron on a budget," *SIAM J. on Comput.*, vol. 37, no. 5, pp. 1342–1372, 2008.

[48] F. Orabona, J. Keshet, and B. Caputo, "Bounded kernel-based online learning," *J. Mach. Learn. Res.*, vol. 10, pp. 2643–2666, Dec. 2009.

[49] W. He and S. Wu, "A kernel-based Perceptron with dynamic memory," *Neural Netw.*, vol. 25, no. 1, pp. 106–113, 2012.

[50] Z. Wang, K. Crammer, and S. Vucetic, "Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale SVM training," *J. Mach. Learn. Res.*, vol. 13, pp. 3103–3131, Oct. 2012.

[51] L. Zhang, J. Yi, R. Jin, M. Lin, and X. He, "Online kernel learning with a near optimal sparsity bound," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 621–629.

[52] W. He and J. T. Kwok, "Simple randomized algorithms for online learning with kernels," *Neural Netw.*, vol. 60, pp. 17–24, Dec. 2014.

[53] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1177–1184.

[54] B. Dai *et al.*, "Scalable kernel methods via doubly stochastic gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3041–3049.

[55] A. Singh, N. Ahuja, and P. Moulin, "Online learning with kernels: Overcoming the growing sum problem," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2012, pp. 1–6.

[56] G. N. Chen, *Matrix Theory and Applications*, (in Chinese). Beijing, China: Science Press, 2007, pp. 88–139.

Wenwu He received the Ph.D. degree in statistics from Central South University, Changsha, China, in 2008.

He visited The Hong Kong University of Science and Technology, Hong Kong, from 2011 to 2012, and the University of Michigan, Ann Arbor, MI, USA, in 2014 and 2015. He is currently an Associate Professor with the School of Mathematics and Physics, Fujian University of Technology, Fuzhou, China. His current research interests include statistical machine learning, online learning, and neural computing.

James Tin-Yau Kwok (SM'07) received the Ph.D. degree in computer science from The Hong Kong University of Science and Technology, Hong Kong, in 1996.

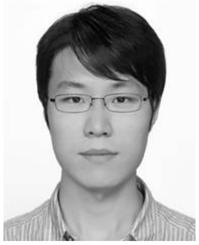
He is currently a Professor with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. His current research interests include kernel methods, machine learning, pattern recognition, and artificial neural networks.

Prof. Kwok received the IEEE Outstanding Paper Award in 2004 and the Second Class Award in Natural Sciences from the Ministry of Education, China, in 2008. He has been a Program Co-Chair for a number of international conferences, and served as an Associate Editor for the IEEE Transactions on Neural Networks and Learning Systems from 2006 to 2012. He is currently an Associate Editor of *Neurocomputing*.

Ji Zhu received the Ph.D. degree in statistics from Stanford University, Stanford, CA, USA, in 2003.

He is currently a Professor with the Department of Statistics, University of Michigan, Ann Arbor, MI, USA, and a Courtesy Professor with the Department of Electrical Engineering and Computer Science. His current research interests include statistical learning, high-dimensional data, and statistical network analysis.

Prof. Zhu was elected as a member of International Statistical Institute in 2010 and a fellow of American Statistical Association in 2013, and received a CAREER Award from the National Science Foundation in 2008. He has been a Program Chair for a number of international conferences. He is currently an Associate Editor of the *Journal of the American Statistical Association*, the *Journal of the Computational and Graphical Statistics*, *Biometrika*, *International Statistical Review*, and *Statistics in Biosciences*.



Yang Liu received the bachelor's degree in information security from Shanghai Jiao Tong University, Shanghai, China, in 2010, the M.S. degree in electrical engineering systems and mathematics from the University of Michigan, Ann Arbor, MI, USA, in 2012 and 2014, respectively, and the Ph.D. degree from the Department of Electrical Engineering Systems, University of Michigan, in 2015.

He is currently a Post-Doctoral Fellow with Harvard University, Cambridge, MA, USA. He has authored or co-authored several technical papers in top journals and conferences of the IEEE/Association for Computing Machinery (ACM). His current research interests include developing learning theory, signal process solutions, and optimal decision making strategies toward processing large scale and potentially noisy data via modeling, and mathematical analysis.

Dr. Liu is an ACM Student Member.