A Note on the Unification of Adaptive Online Learning

Wenwu He, James T. Kwok, Senior Member, IEEE, Ji Zhu, 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 guarantee are derived. Further, a general equation derived from matrix analysis generalizes the adaptive learning to nonlinear case with kernel trick.

Index Terms—Online Learning, Adaptive Gradient Descent, Second order information, Follow the Bregman Divergence Leader

I. INTRODUCTION

Online learning, in which the instances arrive sequentially, is a popular and natural approach in many real-time and lifelong 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 $w_t \in F \subseteq \mathbb{R}^n$ ($F$ 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 $\text{Regret}_F = \sum_{t=1}^T (\ell_t(w_t) - \ell_t(\hat{w}))$, where $\hat{w} \in F$ is a competitor. Note that, this paper is focusing on full information online learning algorithms and that the entire loss function and gradient or (Hessian) are observed and computable, that is, bandit-type algorithms are not considered.

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

In cases where different feature dimensions carry different amounts of information, GD can be significantly improved by incorporating 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 FOBOS [2, 3] and adaptive RDA [2, 4]. They update the weight as

$$w_{t+1} = \Pi_F^A (w_t - \eta A_t^{-1} g_t),$$

where $\Pi_F^A (\cdot) = \arg \min_{w \in F} \| w - \cdot \|_A$ is the projection operator, $\| \cdot \|_A = \sqrt{\cdot, A}$, and the positive definite matrix $A_t$ contains second-order information. Another algorithm that is closely related to A-FOBOS is FTPRL [5], but without the sparseness regularizer. [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 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 NAROW [10], which uses both adaptive and fixed second-order information. A recent algorithm that uses a similar idea is exact soft confidence-weighted learning (SCW) [11]. It improves AROW by adding an adaptive margin.

Interestingly, for exp-concave losses [12], two algorithms, FTAL and ONS also use second-order information of the gradient and obtain the logarithmic regret of $O(\ln T)$ [13]. More recently, Orabona et al. [16] showed that ONS has a regret in the order of $\ln(1 + L_T^*)$ for smooth and exp-concave 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 $O(\ln T)$.

Recent work [17] introduces a normalized adaptive gradient descent (NAG) algorithm that incorporates scale invariance to adaptive gradient. NAG is robust to features scales and collapses the range hyper-parameter search required to achieve good performance. The idea of adaption presented in [17] is similar to that presented in [5] or [2]. A noticeable adaption method which is different to what we discuss here is variance-based SGD [18]. V-SGD uses a per-parameter learning rate

1Examples of exp-concave 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 $\ell_t(w_t) = \langle (w_t, x_t) - y_t \rangle^2$, 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. So exp-concave can be viewed as a relaxation of strongly convex.
proportional to an estimate of gradient squared divided by variance and second derivative. [18] analyses the asymptotic convergence while the rate and the regret bound are not clear.

The theme of the 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, Following The Bregman Divergence Leader, to unify existing second-order online learning algorithms. Also two simple adaptive algorithms, Adaptive Exponential Gradient and Simple Augment Adaptive Algorithm are proposed, presented and analyzed. Finally, existing adaptive algorithms mainly consider linear learning and extending them to nonlinear cases is non-trivial. We propose and prove a matrix equation which can provide a general way to extend linear adaptive learning to nonlinear learning.

Kernel trick is popular for nonlinear learning [19], [20]. The input $x_t$ is first mapped to a RKHS (Reproducing Kernel Hilbert Space) feature space $H$, i.e., $x_t \rightarrow \phi(x_t) \in H$, then replace the inner product $\langle x_t, x_j \rangle$ with $k(x_t, x_j) = \langle \phi(x_t), \phi(x_j) \rangle$. Due to the representation theorem [20], $w_t = \sum_{t \in S_t} \alpha_t \phi(x_t)$, where $S_t$ is the set of support. By the property of reproduction, $f_t(x) = \sum_{t \in S_t} \alpha_t k(x, x_t)$ in kernelized case. 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, [18] uses a matrix equation [21] to kernelize SOP. Another approach in [22] and [29] use a representor 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 also are other types of adaptation for the stepsize, such as stochastic meta-descent (SMD) [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 bound 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, Follow The Bregman Divergence Leader. The simple adaptive algorithm Adaptive Exponential Gradient is also presented there. 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. Full version can be found in [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 semi-definite matrices. Moreover, $A > 0$ (resp. $A \succeq 0$) when $A \in S_n^{++}$ (resp. $A \in S_n^{+}$). For a matrix $A$, $\text{det}(A)$ is its determinant, $\text{Tr}(A)$ its trace and $\text{diag}(A)$ its diagonal matrix. For $A \in S_n^{++}$, $\|x\|_A = \sqrt{x^T A x}$, where $(\cdot, \cdot)$ is the inner product. Moreover, $\Pi_A^2(v) = \arg\min_{u \in \mathbb{R}^n} \|u - v\|_A$, where $u, v \in \mathbb{R}^n$, is the projection of $v$ based on $\| \cdot \|_A$.

**II. REVIEW**

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

Denote $B_{\mathcal{F}}(\cdot, \cdot)$ as the measure Bregman Divergence (BD) and $\Pi_{\mathcal{F}}(\cdot, \cdot) = \arg\min_{v \in \mathcal{F}} B_{\mathcal{F}}(u, v)$ is the projection based on it. For a strongly convex and differentiable function $R(\cdot)$, $B_{\mathcal{F}}(u, v) = R(u) - R(v) - \langle \nabla R(v), u - v \rangle$ (more details found in Section III-A). The conjugate dual (CD) of $R$ is $\tilde{R}(\theta) = \sup_{w \in \mathcal{F}} \langle w, \theta \rangle - R(w)$.

Moreover, let $\{t\} = \{1, \ldots, t\}$, $\cdot|_{t}$ be the shorthand for $\sum_{t=1}^t (\cdot)_t$, and $(\cdot)|_{S_t}$ be the partial sum, $S_t \subseteq \{t\}$. 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 $\ell_t(\cdot)$ is $\varepsilon$-exp-concave if for all $w \in \mathcal{F}$ 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 $w_{t+1} = \arg\min_{w \in \mathcal{F}} \ell_t(w)$. FTRL is based on empirical risk minimization and relies entirely on the observed history [29]. As such, its $w_t$ solution may shift drastically from round to round. An interesting example in which FTL fails can be found in [28].

A natural modification of FTL is to add a regularizer, leading to the Follow The Regularized Leader (FTRL) algorithm [27], [28], [29]. Given a strongly convex regularizer $R(\cdot)$, FTRL updates $w_t$ as

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

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

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

B. Follow The Proximal-Regularized Leader

As mentioned in Section II 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 second-order information of the loss’s (sub)gradient. Given

2Examples: For $q > 1$, let $R(w) = \frac{1}{2} \|w\|_2^2$, then $R^*(\theta) = \frac{1}{2} \|\theta\|_2^2$, where $\frac{1}{p} + \frac{1}{q} = 1$. For $A \in S_n^{++}$, let $R(w) = \frac{1}{2} \|w\|_A^2$, then $R^*(\theta) = \frac{1}{2} \|\theta\|_2^2$.

3Interesting examples can be found in [5], [3] or references in Section I.
By using the input correlation matrix, Second-Order Perceptron (SOP) in \ref{[8]} updates \( w_t \) as
\begin{equation}
    w_t = (aI + \sum_{\tau \in M_{t-1} \cup t} x_\tau x_\tau^\top)^{-1} \sum_{\tau \in M_{t-1}} y_\tau x_\tau,
\end{equation}
where \( a \geq 0 \). 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), Narrow Adaptive Regularization of Weights (NAROW) in \ref{[10]} updates \( w_t \) as
\begin{equation}
    w_t = \left( I + \sum_{\tau \in M_{t-1} \cup t} x_\tau x_\tau^\top \right)^{-1} \sum_{\tau \in M_{t-1}} y_\tau x_\tau, \tag{7}
\end{equation}
where \( r_\tau > 0 \).

For an exp-concave loss Follow The Approximate Learner (FTAL) \ref{[15]} updates \( w_t \) as
\begin{equation}
    w_{t+1} = \arg \min_{w \in \mathcal{F}} \sum_{\tau = 1}^t \tilde{\ell}_\tau (w), \tag{8}
\end{equation}
where \( \tilde{\ell}_\tau (w) \triangleq \ell_\tau (w) + g_\tau^\top (w - w_\tau) + \frac{1}{2\tau} g_\tau^\top g_\tau (w - w_\tau) + \text{nd} \frac{1}{2} \min \left\{ \frac{1}{1 + \epsilon \tau^2}, \epsilon \right\} \). It is understandable that \( \tilde{\ell}_\tau (w) \) is a second-order approximations of \( \ell_\tau (w) \). As showed in \ref{[15]}, FTAL is equivalent to the following algorithm, called Online Newton Step (ONS) for its close connection to the Newton method, which update \( w_t \) as
\begin{equation}
    w_{t+1} = \Pi_{t} \left\{ A_t^{-1} b_t \right\}, \tag{9}
\end{equation}
where \( A_t = \sum_{\tau = 1}^t g_\tau g_\tau^\top, b_t = \sum_{\tau = 1}^t (g_\tau^\top w_\tau - r_\tau g_\tau) \) and \( \frac{1}{r} = \frac{1}{2} \min \left\{ \frac{1}{1 + \epsilon \tau^2}, \epsilon \right\} \). We will see that in the later part of the paper, \ref{[9] can be updated equivalently with the closed-form \ref{[14] where \( \eta = r \) and \( A_t \) are the same as in \ref{[9].

III. ADAPTIVE ONLINE LEARNING FRAMEWORK

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

A. Bregman Divergence

A core theme of the section is that of the measure, i.e. Bregman Divergence, which is first introduced in \ref{[30]}. Recall the definition of BD, for a strongly convex and differentiable function \( R(u) \), \( B_R (u, v) = R(u) - R(v) - \langle \nabla R(v), (u - v) \rangle \). 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

\footnotetext{Here \( M_t \) is the index set of the mistake rounds, and \( k_t \) is the index set of rounds whose prediction is correct but \( \ell_t (w_t) > 0 \).}
nonnegative. Also, by Lemma 1 of [31], BD is strongly convex w.r.t. its first argument.

We introduce several properties of BD. \( B_R(u, v) = 0 \) 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 the paper:

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

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

In particular, when \( R(w) = \frac{1}{2}||w||_2^2, B_R(u, v) = \frac{1}{2}||w - v||_2^2 \). When \( R(w) = \sum_{i=1}^{n} \left( w[i] \ln \frac{w[i]}{v[i]} + v[i] - w[i] \right) \), where \( w_i \geq 0 \) \( \forall i \), \( B_R(w) = \sum_{i=1}^{n} \left( w[i] \ln \frac{w[i]}{v[i]} + v[i] - w[i] \right) \) is the generalized KL divergence [33], [32], [27]. For the two BDs, Kivinen et al. [33] developed the linear gradient descent algorithm and exponential gradient 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 (ADMM) [36].

Interestingly, BD is also helpful in online learning with limited feedback (bandit), e.g., in [37] where a curious link between the notion of BDs and self-concordant barriers is limited feedback (bandit), e.g., in [37] where a curious link between the notion of BDs and self-concordant barriers is.

B. Follow The Bregman Divergence Leader

In the following, we propose the Follow The Bregman Di-
vergence Leader (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

\[
w_{t+1} = \arg \min_{w \in \mathcal{F}} \sum_{\tau=1}^{T} \left( B_{R_\tau}(w, v_\tau) + \eta \ell_\tau(w) \right),
\]

where \( R_\tau(\cdot) \) is strongly convex. Hereafter, we replace \( \ell_\tau(w) \) with \( \langle g_\tau, w \rangle \) unless otherwise specified. In the sequel, we either set \( v_\tau = w_\tau \) so that the learner tries to keep \( w \) close to the \( w_\tau \) learned in the previous round; or set \( v_\tau \) 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 \( v \).

Remark 1: When \( R_\tau(w) = \frac{\tau}{2} \| Q^{\tau} w \|_2^2 \) and \( v_\tau = w_\tau \), FTBDL reduces to FTRL in Section II-B.

Remark 2: When \( Q^\tau = g_\tau g_\tau^T \), \( R_\tau(w) = \frac{\tau}{2} \| Q^\tau w \|_2^2 \), 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.

**Algorithm 1** Follow The Bregman Divergence Leader.

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

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

\[
w_{t+1} = \arg \min_{w \in \mathcal{F}} \sum_{\tau=1}^{T} \ell^*_{R_\tau}(w) = \arg \min_{w \in \mathcal{F}} \ell^*_{R_{1:t}}(w),
\]

where \( \ell^*_{R_{1:t}}(w) = \sum_{\tau=1}^{t} \ell^*_{R_\tau}(w) \). The following Proposition will show that \( w_{t+1} \) can be computed in closed-form, as:

\[
w_{t+1} = \Pi_{R_{1:t}, \mathcal{F}} \left( \nabla R_{1:t}^{-1} \left( \sum_{\tau=1}^{T} (\nabla R_\tau(v_\tau) - \eta g_\tau) \right) \right),
\]

Proof is in Appendix A.

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

\[
w_{t+1} = \Pi_{R_{1:t}, \mathcal{F}} \left( \arg \min_{w \in \mathcal{F}} \ell^*_{R_{1:t}}(w) \right),
\]

and that

\[
\Pi_{R_{1:t}, \mathcal{F}}(v) = \Pi_{R_{1:t}, \mathcal{F}}(v).
\]

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

**Remark 3:** When \( R_\tau = \frac{\eta}{2} \| w \|_2^2, v_\tau = 0 \) and \( \eta = 1 \), (11) becomes a simple version of A-RDA.

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

and its closed-form solution is

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

C. Theoretical Properties

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

\[
w_{t+1} = \Pi_{A_t}(w_t - \eta A_t^{-1} g_t).
\]

Generally speaking, 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 substrates. 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. [15] provides 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 and \( O(\sqrt{T}) \) for GD.
Actually, for A-FOBOS, $B_{R_t}(w, w_t) = \frac{1}{2}\|w - w_t\|_{Q_t}^2$, and the closed-form solution of (4) leads to (14). Clearly, when $A_t = \sqrt{t}$ it reduces to standard GD. When

As will be revealed in Section 4.4, different adaptive algorithms mainly differ in the choice of the matrix $A_t$, and on whether $v_t$ is origin-centered (i.e., $v_t = 0$ as in A-RDA) or updated iteratively (e.g., $v_t = w_t$ as in A-FOBOS). Here we summarize several special cases and connect them back to existing methods.

1) $v_t = w_t$: Let $R_t(w) = \frac{1}{2}\|w\|_{A_t - A_{t-1}}^2$, then A-FOBOS is a specific case of the following update rule, called mirror descent (MD) [43], [44],

\[ w_{t+1} = \arg \min_{w \in F} B_{R_t}(w, w_t) + \eta \langle g_t, w \rangle. \]  \hspace{1cm} (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. [15] can be computed in closed-form with (12), and as showed in proposition [3] it also has a regret guarantee.

The following Proposition will show the equivalence between MD and FTBDL, and then FTBDL [10] covers A-FOBOS [2]. Proof is in Appendix B.

Proposition 2: Assume that $\bar{w}_t = \bar{w}_{t-1}$. The mirror descent update

\[ \bar{w}_{t+1} = \arg \min_{w \in F} \{ B_{R_t}(w, \bar{w}_t) + \eta \langle g_t, w \rangle \} \]

and FTBDL update

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

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

By the equivalence, regret bounds can be derived with ease from existing results. For example, using Lemma 16 in [2] or Theorem 4.1 in [42], the regret for FTBDL ($v_t = w_t$) can be bounded as follows. Proof can be found in Appendix C.

Proposition 3: When $v_t = w_t$, the regret for Algorithm [1] w.r.t. $w \in F$ is bounded by

\[ \text{Regret}_T \leq \frac{1}{\eta} \sum_{t=1}^{T} B_{R_t}(\tilde{w}, w_t) + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2_{R_t^{-1}}, \]  \hspace{1cm} (16)

This allows us to derive specific regrets that correspond to different BD using different $R$'s.

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

\[ \text{Regret}_T \leq \sum_{t=1}^{T} \frac{1}{2}\|w - w_t\|_{Q_t}^2 + \frac{1}{2} \sum_{t=1}^{T} \|g_t\|^2_{Q_t^{-1}}. \]  \hspace{1cm} (17)

The above regret gains a factor $\frac{1}{2}$ in the second right-hand side term, compared with that of Theorem 2 in [5]. Using the diagonal $Q_t$ in [3], [17] becomes

\[ \text{Regret}_T \leq \sum_{t=1}^{T} \frac{1}{2}\|w - w_t\|_{Q_t}^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2_{Q_t^{-1}}. \]  \hspace{1cm} (18)

Further, use Lemma 7 in [5], i.e., for any non-negative real sequence $s_1, \ldots, s_T$,

\[ \sum_{r=1}^{T} \frac{s_r}{\sqrt{\sum_{r=1}^{T} s_r}} \leq 2 \sqrt{\sum_{r=1}^{T} s_r}, \]

the regret reduces to $\sqrt{D_2} \sqrt{\sum_{r=1}^{T} \|g_t\|^2_2} \leq \sqrt{D_2} G_2 \sqrt{T}$ and then it has a tighter regret than GD.

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

\[ w_{t+1} = \arg \min_{w \in F} \{ B_{R_t}(v, w) + \eta \langle g_{1:t}, w \rangle \}. \]  \hspace{1cm} (19)

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

\[ w_{t+1} = \arg \min_{w \in F} \{ R_{1:t}(w) + \eta \langle g_{1:t}, w \rangle \}. \]

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

\[ w_{t+1} = \arg \min_{w \in F} \{ R_{1:t}(w) + \eta g_{1:t}, w \}. \]  \hspace{1cm} (20)

For update (20) we have the following regret (proof is 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 w.r.t. $\tilde{w} \in F$ is bounded by

\[ \text{Regret}_T \leq \frac{1}{\eta} R_{1:T}(\tilde{w}) - \frac{1}{\eta} U + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2_{R_t^{-1}} + O(1), \]  \hspace{1cm} (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).

\[ \text{Regret}_T \leq \frac{1}{\eta} R_{1:T}(\tilde{w}, v) - \frac{1}{\eta} U + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2_{R_t^{-1}} + O(1), \]

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

The above bounds allow for more refined results to be derived in specific cases. Consider simple A-RDA [3] and let $\sigma_{1:t} = 2\sqrt{G_2 T \sqrt{D_2}}$. Bound (21) provides the regret $\sqrt{D_2} G_2 \sqrt{T} - U + O(1)$, which gains a factor $\frac{1}{2}$ (meanwhile $U > 0$) over standard GD ($F = \{w \mid \|w\|_2 \leq D_2/2\}$).

D. Adaptive Exponential Gradient

In this subsection we will present a new adaptive algorithm. Consider the normalized KL divergence, i.e., $R(w) = \sum_{i=1}^{n} w[i] \ln w[i], w \in F$ where $F = \{w \mid \|w\|_1 = 1, w[i] \geq 0\}$, then $B_{R_t}(w, v) = \sum_{i=1}^{n} w[i] \ln w[i]$ is strongly convex w.r.t. $\|\cdot\|_1$ [28] and the dual norm is $\|\cdot\|_{\infty}$. Also the conjugate dual $R^*(\theta) = \ln(\sum_{i=1}^{n} e^{\theta[i]})$ [28]. The resulting algorithm is often called normalized exponential gradient (EG) [33], [28].

Now we design a simple adaptive EG (Algorithm [2]). Let $v = \left[\frac{v_1}{\sqrt{n}}, \frac{v_2}{\sqrt{n}}, \ldots, \frac{v_n}{\sqrt{n}}\right]^T$, $\sigma_1 > 0$, $\sigma_t \geq 0$ for $t > 1$, $R_{1:t}(w) =$
Further, setting \( \eta_k = \frac{\eta}{\sigma_{i,t}} \). Using the fact that \( B_{R_{i,t}} (w,v) = \sigma_{i,t} B_R (w,v) \), we update \( w_t \) (\( w_1 = v \)) with
\[
w_{t+1} = \arg \min_{w \in \mathcal{F}} \{ B_R (w,v) + \eta_t g_{1:t}, w \}.
\]

**Algorithm 2: Adaptive Exponential Gradient.**

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

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

**Proposition 5:** The closed-form update for adaptive EG algorithm (22) is
\[
w_{t+1}[i] = \frac{w_t[i] e^{-\eta_t g_t[i]}}{\sum_{j=1}^n w_t[j] e^{-\eta_t g_t[j]}},
\]
and it is equivalent to the following update
\[
w_{t+1} = \arg \min_{w \in \mathcal{F}} \{ B_R (w,v) + \eta_t g_{1:t}, w \},
\]
where \( w_1 = v \).

Algorithm 2 is similar to normalized EG but has an adaptive stepsize \( \eta_t \). For its regret we have the following upper bound.

**Proposition 6:** The regret for Algorithm 2 w.r.t. \( \dot{w} \in \mathcal{F} \) is bounded by
\[
\text{Regret}_T \leq \sigma_{1:T} \eta \ln n + \frac{n}{2} \sum_{t=1}^T \| g_t \|_2^2 \sigma_{1:t}.
\]
In particular, setting \( \sigma_{1:t} = \sqrt{t}, \eta = \sqrt{\ln n/G_\infty} \) yields
\[
\text{Regret}_T \leq 2G_\infty \sqrt{T \ln n}.
\]
Further, setting \( \sigma_{1:t} = \sqrt{\sum_{t=1}^T \| g_t \|_2^2}, \eta = G_\infty \) yields
\[
\text{Regret}_T \leq 2 \sqrt{\ln n \sum_{t=1}^T \| g_t \|_2^2} \leq 2G_\infty \sqrt{T \ln n}.
\]
Regret bound (26) is identical to that of Corollary 2.14 of [28] by setting \( \tilde{\eta} = \sqrt{\ln n/T}/G_\infty \) therein, but \( \eta \) 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, etc. The regret (27) is even tighter than (26) and \( \eta \) is also free w.r.t. \( T \).

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

**IV. UNIFICATION OF ADAPTIVE ALGORITHMS**

FTBDL proposed in Section III provides a basis for the unification of different adaptive algorithms and preliminarily it has been shown 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 will be presented here: 1) the algorithms involved will be cleared up, 2) new observations on the connection within them will be discussed, and 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 specific form
\[
w_{t+1} = \arg \min_{w \in \mathcal{F}} \sum_{\tau \in S_t} (B_{R_t} (w,v_\tau) + \eta (g_\tau, w)),
\]
where \( R_t = \frac{1}{2} \| w \|_2^2 A_t - A_{t-1} \) (then \( R_{t, t} = \frac{1}{2} \| w \|_2^2 A_t \)) and \( A_t \in S_t^n \). \( S_t \) excludes those rounds in which \( A_t \) is not updated, e.g., SOP [8] or NARROW [10] uses \( S_t \neq \emptyset \), \( v_t = v_t' \) or \( v_t = 0 \) as discussed in Section III-C. In particular, let \( A_t = A_t^{a,v,b} \) and
\[
A_t^{a,v,b} = a_t I + \left( \sum_{\tau \in S_t} Q_\tau \right)^b,
\]
for some \( a_t \geq 0, b \in \{ \frac{1}{2}, 1 \}, Q_t \in S_t^n \), and \( S_t \subseteq [t] \).

With (28) and (29) at hand, adaptive algorithms presented in Section III 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 \( v_\tau = v_\tau' \), \( A_t = A_t^{a} \) and \( R_t = \frac{1}{2} \| w \|_2^2 A_t - A_{t-1} \), where \( a \geq 0, b = \frac{1}{2}, S_t = [t] \) and \( Q_\tau = g_\tau g_\tau' \) (or \( Q_\tau \) is diagonal and \( (Q_\tau)_{i,i} = (g_\tau, g_\tau')^2 \)).

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 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 \( A_t^{a} \) can be calculated. In practice, optimal setting of \( a \) can be tuned by validation.

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

\[
\text{Regret}_T \leq \frac{a}{2\eta} \| \dot{w} \|_2^2 + \frac{D_2^2}{2\eta} \Tr(G^{1/2}) + \eta \Tr(G^{1/2}),
\]
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.,

8\ In regret (16) \( \sum_{\tau=1}^T B_R (\dot{w}, w_\tau) = \frac{1}{2} \| \dot{w} - w_1 \|_2^2 A_1 + \frac{1}{2} \sum_{\tau=2}^T (\| \dot{w} - w_\tau \|_2^2 A_{\tau-1} - \| \dot{w} - w_\tau \|_2^2 A_{\tau-1} + \frac{1}{2} \| \dot{w} - w_\tau \|_2^2 A_{\tau-1}) \leq \frac{1}{2} \| \dot{w} - w_1 \|_2^2 + \frac{1}{2} \max_{\tau} \| \dot{w} - w_\tau \|_2^2 \| G_\tau \|_2^2, \) where we used \( w_1 = 0 \) for A-FOBOS and \( \| w \|_2 \leq \| w \|_2 \Tr(A) \) for \( A \in S_t^n \). By the Lemma 10 in [2]. \( \sum_{\tau=1}^T \| g_\tau \|_{A_t^{-1}}^2 = \sum_{\tau=1}^T \| g_\tau \|_{A_t^{-1}}^2 \leq 2T \| G \|_2 \).
\[ \text{Regret}_T \leq \sqrt{2D_2 \text{Tr}(G_T^{1/2})} = \sqrt{2D_2 \sum_{i=1}^{n} \lambda_i^{1/2}}, \text{ where } \lambda_i \text{ is the } i\text{th eigenvalue of } G_T. \]

Note that, \( G_T \in S_+^n \) and \( \lambda_i > 0 \), then
\[ \left( \sum_{i=1}^{n} \lambda_i^{1/2} \right)^2 > \sum_{i=1}^{n} \lambda_i = \text{Tr}(G_T). \quad (30) \]

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

Generally, 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 FTPDL.

\textbf{Proposition 8:} FTPRL in \([2]\) is a specific case of FTBRL in \([28]\) with \( \eta = 1, v_r = w_r, A_t = A_t^{ab} \) and \( R_t = \frac{1}{2}\|w\|_{Q_r^T} \), where \( a = 0, b = 1, Q_1 \in S_+^n, Q_r > 1 \in S_+^n \) 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 \( L_\infty \)-ball. In particular, let \( Q_t \) be diagonal and
\[
(Q_{1:t})_{i,i} = \frac{\sqrt{\sum_{i=1}^{T} (g_i[i])^2}}{\text{D}_\infty / \sqrt{2}}. 
\]

It is identical to A-FOBOS using diagonal matrix \( a = 0 \) with \( \eta = D_\infty / \sqrt{2} \). Then regret \([17]\) for FTPRL reduces to
\[ \text{Regret}_T \leq \sqrt{2D_2 \sum_{i=1}^{n} \left( \sum_{t=1}^{T} (g_i[i])^2 \right)} \quad (31) \]

To cover a \( L_\infty \)-ball feasible region, \( D_2 = \sqrt{n}D_\infty \), and then for GD, its upper regret bound has to be \( \sqrt{2nD_\infty G_2 \sqrt{T}} \). Hence the per-coordinate adaption outperforms GD. Actually, by Cauchy-Schwarz Inequality, \( \sum_{i=1}^{n} \left( \sqrt{\sum_{t=1}^{T} (g_i[i])^2} \right) \geq \left( \sum_{i=1}^{n} \sqrt{\sum_{t=1}^{T} (g_i[i])^2} \right) \),

which means that
\[ \sqrt{n} \sum_{i=1}^{T} ||g_i||_2^2 \geq \sqrt{\sum_{i=1}^{n} \sum_{t=1}^{T} (g_i[i])^2}. \]

And, for the coordinate-constant adaption in the case of interest, its upper regret bound is \( \sqrt{2nD_\infty \sum_{i=1}^{T} ||g_i||_2^2} \). Then it is also looser than that of per-coordinate adaption. But, for \( L_\infty \)-ball feasible region, the conclusion is opposite (similar to \([30]\), \( \sum_{i=1}^{n} \sqrt{\sum_{t=1}^{T} (g_i[i])^2} \geq \sqrt{\sum_{i=1}^{n} ||g_i||_2^2} \)).

\textbf{2) ONS and FTAL:} For strongly convex loss function, a regret in scale of \( O(\ln T) \) can be derived, even for GD. And the magic is that, for general convex loss, e.g., linear loss, \( \ell_t(w_t) - \ell_t(\hat{w}) \leq \langle g_t, w_t - \hat{w} \rangle \), but for \( \sigma \)-strongly convex loss, \( \ell_t(w_t) - \ell_t(\hat{w}) \leq \langle g_t, w_t - \hat{w} \rangle - \frac{\sigma}{2} ||w_t - \hat{w}||_2^2 \), where \( w_t, \hat{w} \in \mathcal{F} \). The extra non-positive term improves the regret, and for \( \eta_t = (\sigma t)^{-1} \), the upper bound of the regret for GD is
\[ \frac{G_2^2}{\sigma^2} (1 + \ln T). \quad (32) \]

As discussed in the introduction, some widely used loss functions in learning problems are not strongly convex, but they may still obtain low regrets. In particular, for exp-concave loss functions, a property analogous to that of strongly convex function can be exploited. That is, for \( \frac{1}{\sigma} = \frac{1}{2} \min \left\{ \frac{1}{4D_2^2}, \varepsilon \right\} \)
\[ \ell_t(w_t) - \ell_t(\hat{w}) \leq \langle g_t, w_t - \hat{w} \rangle - \frac{1}{2\sigma} ||w_t - \hat{w}||_2^2, \quad (32) \]

where \( \exp(-\varepsilon \ell_t(\cdot)) \) is concave, \( Q_t = g_t^T g_t^T \) and \( w_t, \hat{w} \in \mathcal{F} \). Based on this, two algorithms, ONS and FTAL which enjoy the logarithmic regret are developed in \([15]\). The two can also be unified under the proposed framework.

\textbf{Proposition 9:} FTAL in \([8]\) can be recovered with FTBRL in \([28]\) on setting \( v_r = w_r, S_t = [t], A_t = A_t^{ab} \) and \( R_t = \frac{1}{2}\|w\|_{Q_r^T}^2 \), where \( a = 0, b = 1, Q_r = g_r^T g_r^T \), \( \eta = r \) and \( \frac{1}{r} = \frac{1}{2} \min \left\{ \frac{1}{4D_2^2}, \varepsilon \right\} \).

From Proposition \[9\], FTAL is identical with FTPRL using \( R_t = \frac{1}{2}\|w\|_{Q_r^T}^2 \), where \( Q_r = g_r^T g_r^T \). Further, as showed in \([15]\), ONS is equivalent to, but an efficient implementation to the FTAL. Then by Proposition \[10\] it can be recovered by MD formally, we state it as bellow.

\textbf{Proposition 10:} Ons in \([9]\) can be recovered with FTBRL in \([28]\) on setting \( v_r = w_r, S_t = [t], A_t = A_t^{ab} \) and \( R_t = \frac{1}{2}\|w\|_{Q_r}^2 \), where \( a = 0, b = 1, Q_T = g_T^T g_T^T \), \( \eta = r \) and \( \frac{1}{r} = \frac{1}{2} \min \left\{ \frac{1}{4D_2^2}, \varepsilon \right\} \). Equivalently it can be updated by MD
\[ (28) \]

with \( FTBDL \) in (28) reduces to

\[ 10 \]

\( RDA \) is upper bounded by \( \sigma_r I \) in particular, \( A-RDA \) in (5) is equivalent to \( FTBDL \) in (28) with \( v_r = 0, \) \( R_r = \frac{1}{2} \| w \|_2^2 \) and \( A_t = A_t^{a,b} \), where \( a \geq G_2, b = \frac{1}{2} \). \( S_t = [t] \), \( Q_r \) is identical to that of A-FOBOS.

As pointed out in Section II, AODG in (13) is a simpler version of A-RDA, and in particular, \( \eta = 1, \) \( R_r = \frac{1}{2} \| w \|_2^2 \) where \( Q_r = \sigma_r I \). 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 \( Q_r = \sigma_r I \).

Omit the non-positive term \(-\frac{\lambda_1}{\eta} \) in (21), the regret for A-RDA is upper bounded by \( \eta \)

\[ \text{Regret}_T \leq \frac{a}{2\eta} \| \hat{w} \|_2^2 + \frac{\| \hat{w} \|_2^2}{2\eta} \text{Tr}(G^2_{T/2}) + \eta \text{Tr}(G^2_{T/2}) + O(1). \]  

The above bound is tighter than that of A-FOBOS since \( \| \hat{w} \|_2^2 < D_2^2 \). The advantage comes from the fact that, for A-FOBOS, the upper bound has to cover \( \frac{1}{\eta} \sum_{t=1}^T \| \hat{w} - w_t \|_2^2, \) but A-RDA only needs to cover \( \frac{1}{\eta} \sum_{t=1}^T \| \hat{w} - \hat{w}_t \|_2^2 = \| \hat{w} \|_2^2, \) here the analysis is based on the upper bound and interestingly, the practical aspects of A-RDA vs. A-FOBOS also support the conclusion.

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

**Proposition 11:** A-RDA in (5) is equivalent to FTBDL in (28) with \( v_r = 0, \) \( R_r = \frac{1}{2} \| w \|_2^2 \) and \( A_t = A_t^{a,b} \), where \( a \geq G_2, b = \frac{1}{2} \). \( S_t = [t] \), \( Q_r \) is identical to that of A-FOBOS.

For hinge loss \( \ell_r(w) = \max(0, |y_r - \langle w, x_r \rangle| - c) \) where \( y_r \) is the target output of \( x_r \), we also have \( g_t g_r^T = x_r^T x_r \) for \( \ell_r > 0 \). Hence, we can redefine \( A_t \) by including the current instance \( x_i \) into the prediction. As suggested in (6), this may lead to performance improvement in some scenarios.

In particular, consider A-RDA in (5) and let \( R_{t,1} = \frac{1}{2} \| w \|_2^2 \) and \( a_t = (a^2 I + \sum_{r=1}^{t} g_t g_r^T)^{-1} \). Augment \( A_t \) as

\[ \hat{A}_{t+1} = (A_t^2 + x_{t+1} x_{t+1}^T)^{-1/2} \]

and let \( \hat{R}_{t,1} = \frac{1}{2} \| w \|_2^2 \) where \( \| g_t \|^2_{R_{t,1}^{-1}} = 1 \) in regret (21) will be replaced by \( \| g_t \|_{\hat{R}_{t,1}^{-1}}^2 = \| g_t \|^2_{A_{t-1}^{-1}} \) and hence \( \hat{A}_{t+1} \leq A_{t+1} \leq A_{t-1}^{-1} \). 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., \( \| g_t \|^2_{\hat{A}_{t+1}^{-1}} < \| g_t \|^2_{A_{t+1}^{-1}} \). Actually, the Woodbury identity gives us \( \hat{A}_{t+1}^{-1} = (1 + x_t^T (A_{t-1}^2)^{-1} x_t) \hat{A}_{t+1}^{-1} - 1 \) and \( A_{t-1}^{-1} = \sqrt{1 + x_t^T (A_{t-1}^2)^{-1} x_t} \hat{A}_{t+1}^{-1} \) and we have

\[ g_t^T \hat{A}_{t+1}^{-1} g_t = \sqrt{1 + x_t^T (A_{t-1}^2)^{-1} x_t} g_t^T \hat{A}_{t+1}^{-1} g_t. \]

Clearly, \( \sqrt{1 + x_t^T (A_{t-1}^2)^{-1} x_t} \neq 1 \) for \( x_t \neq 0 \). Also, when \( g_t \neq 0, \) \( \hat{A}_{t+1} = (A_{t-1}^2 + g_t g_t^T)^{-1/2} = A_{t-1} \), and it is always true (it is trivial when \( g_t = 0 \)) that \( \| g_t \|^2_{R_{t,1}^{-1}} < \| g_t \|^2_{A_{t-1}^{-1}} \). Hence the upper bound still holds but the assumption \( a \geq G_2 \) can be removed.

Further, based on the observation, we develop a simple augment algorithm using diagonal adaptive matrix for losses

\[ 11 \]

\[ 12 \]

Two scenarios for \( \ell_r > 0 \): i) \( y_r - \langle w, x_r \rangle \geq c \), then \( \ell_r = y_r - \langle w, x_r \rangle - c \) and \( g_t = x_r \). ii) \( y_r - \langle w, x_r \rangle < -c \), then \( \ell_r = -c + \langle w, x_r \rangle \) and \( g_t = x_r \). Then we have \( g_t^T g_r^T = x_r^T x_r \) in both cases.

The regret upper bound (33) is still suitable for this regularizer. Actually, in this case \( A_t \leq \sigma_t I + \sum_{r=1}^{t} g_r g_r^T )^{-1/2} \) and then the first two terms still hold. For the third term, \( A_{t-1}^{-1} \leq \left( \sum_{r=1}^{t-1} g_r g_r^T \right)^{1/2} \) when \( a \geq G_2 \) and Lemma 10 in (23) confirms the bound.
that the possess the property $g_r g_r^\top = x_r x_r^\top$. In particular, the algorithm is summarized in Algorithm 3.

**Algorithm 3** Simple Augment Adapative Algorithm.

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

Clearly, Algorithm 3 is a special case of FTBDL in that $v_r = 0$ and $R_r = \frac{\eta}{2} \|w\|^2$. And regret (21) leads to the following bound.

**Proposition 13:** The regret for Algorithm 3 w.r.t. $\hat{w} \in \mathcal{F}$ is bounded by

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

Let $\|w\|^2 \leq D_2/2$ and omit the non-positive term $-\frac{U}{\eta}$, then $\eta = D_2/\sqrt{2}$ yields

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

where $O(1)$ corresponds to the term $\langle x_1, w_1 - w_2 \rangle$.

We can see that, for losses possessing the property $g_r g_r^\top = x_r x_r^\top$, Algorithm 3 gains a factor $\frac{1}{2}$ over GD and meanwhile, $\sqrt{\sum_{t=1}^{T} \|g_r\|^2} \leq G_2 \sqrt{T}$. Also, regret (36) gains a factor $\frac{1}{2}$ 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 $v_r$s, use the original version of $A_t$ or its augment as Algorithm 3 does; set different $a_r$, $b$ or $S_t$; or more generally, use different Bregman Divergences as Algorithm 2 does. Also, 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 will 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 $\langle x, x \rangle$ and then generalize it by replacing the inner product with kernel $k(x, x)$ as been used in [8].

We consider A-RDA type online learning with $\mathcal{F} = \mathbb{R}^n$ and the closed-form for FTBDL in [28] reduces to

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

Here the adaptive matrix $A_t$ involves $x_r x_r^\top$ or $g_r g_r^\top$ 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] to kernelize SOP.

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

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

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

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

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

$$BA^d = BA^{d-1} A = C^{d-1} BA = C^{d-1} CB = C^d B. \quad (40)$$

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

$$B(aI_n + AB)^{-1/2} = (aI_n + BA)^{-1/2} B. \quad (41)$$

The proof is included in Appendix N.

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 $g_t = [g_{t, \tau}] \tau \in S_t$, then $g_{\neg S_t} = 1^\top g_t$ ($g_r$ can be replaced by $x_r$ or $x_r^\top$ as used in SOP or NAROW).

When $A_t = aI_n + g_t g_t^\top$, Proposition 14 gives us

$$w_{t+1} = -\eta (aI_n + g_t g_t^\top)^{-1} \left( aI_n + g_t g_t^\top \right) \left( aI_n + g_t g_t^\top \right)^{-1/2} g_t$$

$$= -\eta (aI_n + g_t g_t^\top)^{-1} g_t. \quad (41)$$

where $g_t g_t^\top$ only involves inner product $\langle g_t, g_t \rangle$. Note that for commonly used losses, $g_r = g_r x_r$, where $g_r$ is a scalar.

Let $x_\tau = [x_{\tau}]$, $g_t = [g_{\tau}]$ and $D(g_t)$ be the diagonal matrix with diagonal elements $g_t$. Then $g_t = x_\tau D(g_t)$ and $A_t = aI_{|S_t|} + D(g_t) K_t D(g_t)$. Here $K_t := x_\tau^\top x_\tau$ 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 update. In this case Proposition 15 gives us

$$w_{t+1} = -\eta (a^2 I_{|S_t|} + g_t g_t^\top)^{-1/2} g_t$$

$$= -\eta (a^2 I_{|S_t|} + g_t g_t^\top)^{-1/2} g_t. \quad (42)$$

Using Sherman-Morrison-Woodbury (SMW) formula [21] (or the matrix inversion lemma [44]), $K_t^{-1}$ can be updated recursively and costs us $O(|S_t|^2)$ each round. It is common to use the diagonal version [5, 2], replacing $aI_{|S_t|} + g_t g_t^\top$ (or $a^2 I_{|S_t|} + g_t g_t^\top$) with its diagonal matrix reduces the cost to $O(|S_t|)$, which is in the same order to that of non-adaptive online learning. In particular, let $c_r = (a + g_r^2 K_t(\tau, \tau))^{-1}$

\[c_r \leq \frac{1}{2} (g_t - \langle w_t, x_t \rangle)^2 \]
(or $c_r = \hat{g}_r / \sqrt{\alpha^2 + \hat{g}_r^2 K_r(\tau, \tau)}$ in case where square root operation is involved), then (41) (or (42)) in the case when diagonal adaptive matrix is used reduces to:

$$w_{t+1}^T = -\eta \sum_{\tau \in S_t} c_r x_r^T =: \sum_{\tau \in S_t} \alpha_r x_r^T,$$

(43)

whose computation cost scales as the same order as that of the non-adaptive online learning.

Alternatively, it is possible to use inequality (18) to get a simple adaptive version with kernels with improvable guarantee. In particular, let $R_{\tau} = \frac{\sigma}{\tau^2} \|w\|_2^2$, $v_\tau = w_t$ 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 $\eta_t = \frac{\eta}{\sigma_{1,t}}$, and its regret upper bound is improved to $\sqrt{2D_2 \sum_{t=1}^n \|g_t\|^2}$.

Algorithm 4 Simple Adaptive Algorithm with kernels.

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

C. Other Topics

A bottleneck for kernelized online learning is that the support set may keep increasing with learning. So budget strategies which keep $|S_t|$ under control [45], [46], [47], [48], [49], [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 $m$-dimension randomized features. Actually, by using the approximating idea, [55] presents a way of overcoming the growing sum problem of the Kernel Least Mean Square (KLMS) algorithm.

VI. CONCLUSION

This paper proposed a framework Follow the Bregman Divergence Leader that covers most popular adaptive algorithms that use 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. Further, the 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 Bregman Divergences 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 2

Proof: Let $w_{t+1} = \arg\min_{w \in \mathbb{R}^n} \ell_{1,t}^R(w)$, and $w_{t+1} = \Pi_{1,t} w_t(w_{t+1})$. By definition, we have

$$\ell_{1,t}^R(w_{t+1}) \leq \ell_{1,t}^R(w_{t+1}).$$

Also, $\nabla \ell_{1,t}^R(w_{t+1}) = 0$ as $\hat{w}_{t+1}$ minimizes $\ell_{1,t}^R$ over $\mathbb{R}^n$. Then,

$$\ell_{1,t}^R(w_{t+1}) - \ell_{1,t}^R(\hat{w}_{t+1}) = B_{1,t}(w_{t+1}, \hat{w}_{t+1}).$$

Further

$$B_{1,t}(w_{t+1}, \hat{w}_{t+1}) \leq B_{1,t}(w_{t+1}, \hat{w}_{t+1}) = \frac{\eta}{2} \ell_{1,t}^R(w_{t+1}).$$

And we have $\ell_{1,t}^R(w_{t+1}) \leq \ell_{1,t}^R(w_{t+1})$. By the assumption of strongly convexity of $R$, the Bregman Divergence is strictly convex w.r.t. its first argument. Thus $\ell_{1,t}^R$ is strictly convex and we have $w_{t+1} = w_{t+1}$.

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

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

$$= \sum_{\tau=1}^t \sum_{\tau=1}^t \{ B_{R_{\tau}}(u, v_\tau) - B_{R_{\tau}}(v, v_\tau)$$

$$- \langle \nabla B_{R_{\tau}}(v, v_\tau), u - v \rangle \},$$

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

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.

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

(44)

$$\tilde{w}_{t+1} = \arg\min_{w \in \mathbb{R}^n} \sum_{t=1}^T \{ B_{R_{t+1}}(w, \tilde{w}_t) + \eta \langle g_t, w \rangle \}. $$

(45)

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

We start with $\tilde{w}_1 = \hat{w}_1$ and assume that $\hat{w}_t = \tilde{w}_t$. By the optimality for (45), the gradient of the objective function w.r.t.
\(\hat{w}_t\) is zero, that is \(\sum_{t=1}^{t-1} (\nabla R_t(\hat{w}_t) - \nabla R_t(\bar{w}_t) + \eta g_t) = 0\).

Then we have
\[
\sum_{t=1}^{t-1} \nabla R_t(\hat{w}_t) = -\eta g_{t:t-1} + \sum_{t=1}^{t-1} \nabla R_t(\bar{w}_r). \tag{46}
\]

On the other hand, by (44),
\[
\hat{w}_{t+1} = \arg \min_{w \in \mathbb{R}^n} \sum_{t=1}^{t} B_{R_t}(w, \hat{w}_t) + \eta (g_t, w)
\]
\[
= \arg \min_{w \in \mathbb{R}^n} \sum_{t=1}^{t} (R_t(w) - R_t(\hat{w}_t) - \langle \nabla R_t(\hat{w}_t), w - \hat{w}_t \rangle) + \eta (g_t, w)
\]
\[
= \arg \min_{w \in \mathbb{R}^n} \sum_{t=1}^{t} (R_t(w) - \sum_{t=1}^{t-1} \nabla R_t(\bar{w}_t), w) + \eta (g_t, w)
\]
\[
= \arg \min_{w \in \mathbb{R}^n} \sum_{t=1}^{t} (R_t(w) - \langle \nabla R_t(\bar{w}_t), w \rangle) + \eta (g_t, w).
\]

Replacing \(\hat{w}_t\) in the above equation with \(\bar{w}_t\) and using (46),
\[
\hat{w}_{t+1} = \arg \min_{w \in \mathbb{R}^n} \sum_{t=1}^{t} (R_t(w) - \langle \nabla R_t(\bar{w}_t), w \rangle) + \eta (g_t, w)
\]
\[
= \arg \min_{w \in \mathbb{R}^n} \sum_{t=1}^{t} \sum_{t=1}^{t-1} \nabla R_t(\bar{w}_r), w) + \eta (g_t, w).
\]

That is
\[
\hat{w}_{t+1} = \arg \min_{w \in \mathbb{R}^n} \sum_{t=1}^{t} (R_t(w) - \langle \nabla R_t(\bar{w}_t), w \rangle) + \eta (g_t, w)
\]
\[
= \arg \min_{w \in \mathbb{R}^n} \sum_{t=1}^{t} (B_{R_t}(w, \bar{w}_t) + \eta (g_t, w))
\]
\[
= \hat{w}_{t+1}.
\]

It completes the proof.

C. Proof of Proposition 2

Proof: Due to the equivalence of MD and FTBDL in Proposition 2, it is sufficient to prove \(\text{Regret}^{\text{MD}}\) for MD (15).

By the derivation of Theorem 4.1 in (42), i.e., (4.21) therein, for any \(w \in \mathcal{F}\) we have \(\ell_t(w) - \ell_t(w) \leq \frac{\Delta_t(w)}{\eta} + \frac{1}{2} \|g_t\|^2 \mathcal{R}^*_t\),

where \(\Delta_t := B_{R_t}(\hat{w}_t, w_t) - B_{R_t}(\hat{w}_t, w_{t+1})\).

Sum the two sides of above inequality then
\[
\text{Regret}^{\text{MD}} \leq \sum_{t=1}^{T} \frac{\Delta_t}{\eta} + \frac{1}{2} \sum_{t=1}^{T} \|g_t\|^2 \mathcal{R}^*_t =: \text{RS1} + \frac{1}{\eta} \sum_{t=1}^{T} \|g_t\|^2 \mathcal{R}^*_t.
\]

By the additivity and non-negativity of BD,
\[\text{RS1} \leq \frac{1}{\eta} B_{R_t}(\hat{w}, w_t) + \frac{1}{\eta} \sum_{t=1}^{T-1} \hat{\Delta}_{t+1} = \frac{1}{\eta} \sum_{t=1}^{T} B_{R_t}(\hat{w}, w_t)\]

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

D. Proof of Proposition 2

Proof: Let \(\tilde{R}_t := \frac{1}{\eta} \mathcal{R}_{t:1}(w)\) and rewrite (20) as
\[
w_{t+1} = \arg \min_{w \in \mathcal{F}} \{ \tilde{R}_t(w) + \langle g_{1:T}, w \rangle \}.
\]

Recall that \(\tilde{R}^*_t(\theta) = \sup_{w \in \mathcal{F}} \{ \langle \theta, w \rangle - \tilde{R}_t(w) \}\), and we have
\[
\text{Regret}^{\text{T}} \leq \sum_{t=1}^{T} \langle g_t, w_t - \hat{w} \rangle - \tilde{R}_t(\hat{w}) + \tilde{R}_t(w)
\]
\[
\leq \sum_{t=1}^{T} \langle g_t, w_t \rangle + \sup_{w \in \mathcal{F}} \{ -\langle g_{1:T}, w \rangle - \tilde{R}_t(w) \}
\]
\[
= \tilde{R}_t(\hat{w}) + \sum_{t=1}^{T} \langle g_t, w_t \rangle + \tilde{R}_t(\hat{w}_{1:T}).
\]

By the optimality in (47) we have
\[
\tilde{R}(g_{1:T}) = -\langle g_{1:T}, w_{1:T} \rangle - \tilde{R}_t(w_{1:T})
\]
\[
= -\langle g_{1:T}, w_{1:T} \rangle - \tilde{R}_{t-1}(w_{1:T-1}) - \frac{1}{\eta} \tilde{R}_t(w_{1:T-1})
\]
\[
\leq \sup_{w \in \mathcal{F}} \{ -\langle g_{1:T}, w \rangle - \tilde{R}_{t-1}(w) \}
\]
\[
= \tilde{R}_{t-1}(\hat{w}_{1:T}) - \frac{1}{\eta} \tilde{R}_t(w_{1:T-1}).
\]

Note that \(R_t\) is strongly convex and \(\tilde{R}_t\) is \(\frac{1}{\eta}\)-strongly convex. Then \(\tilde{R}_t(\theta)\) has \(\eta\)-Lipschitz continuous gradients and
\[
\|\nabla \tilde{R}_t(\theta_1) - \nabla \tilde{R}_t(\theta_2)\|_{\mathcal{R}_t} \leq \eta \|	heta_1 - \theta_2\|_{\mathcal{R}_t}.
\]

Due to the duality of strongly convex and strongly smooth functions (Lemma 2.19 of [28]) we have
\[
\tilde{R}^*_t(\theta_1) \leq \tilde{R}^*_t(\theta_2) + \langle \nabla \tilde{R}^*_t(\theta_2), \theta_1 - \theta_2 \rangle + \frac{\eta}{2} \|	heta_1 - \theta_2\|_{\mathcal{R}_t}^2. \tag{48}
\]

And, for the gradient of the conjugate dual (Lemma 2 of [51]),
\[
\nabla \tilde{R}^*_t(\theta) = \arg \min_{w \in \mathcal{F}} \{ -\langle \theta, w \rangle + \tilde{R}_t(w) \}. \tag{49}
\]

Then, due to (48),
\[
\tilde{R}^*_t(-g_{1:T}) \leq -\tilde{R}^*_{t-1}(-g_{1:T-1}) - \langle \nabla \tilde{R}^*_{t-1}(-g_{1:T-1}), g_t \rangle + \frac{\eta}{2} \|g_t\|^2 \mathcal{R}^*_{t-1}.
\]

Further, (49) and the optimality of (47) show that
\[
\langle \nabla \tilde{R}^*_{t-1}(-g_{1:T-1}), g_t \rangle = \langle w_T, g_t \rangle.
\]

Hence we have
\[
\text{Regret}^{\text{T}} \leq \tilde{R}_t(\hat{w}) - \frac{1}{\eta} \tilde{R}_t(w_{1:T+1}) + \frac{\eta}{2} \|g_T\|^2 \mathcal{R}^*_{t:T-1} + \sum_{t=1}^{T} \langle g_t, w_t \rangle + \tilde{R}_t(\hat{w}_{1:T}).
\]
Repeat this process $T - 1$ times (on time index $t$) and we have

$$
\text{Regret}_T \leq \hat{R}_T(\hat{w}) + \frac{U'}{\eta} + \sum_{t=2}^{T} \frac{\eta}{2} g_t^2 R_{t-1:t} + (g_1, w_1) + \hat{R}_1(-g_1),
$$

where $U' = \sum_{t=2}^{T} R_t(w_{t+1})$. Using the fact that $\hat{R}_1(-g_1) = -(g_1, w_2) - \frac{\eta}{2} R_1(w_2)$ we complete the proof. $\blacksquare$

E. Proof of Proposition 3

Proof: It is easy to verify that $w_{t+1}$ in (22) can be computed in closed-form as

$$
w_{t+1} = \Pi_{R,F} \left( \nabla R^* (\nabla R(v) - \eta g_{t+1}) \right).
$$

(50)

Since $\nabla R^* (\theta)[i] = \sum_{j=1}^{n} e^{\theta[i]}$, we have

$$
w_{t+1}[i] = \frac{e^{-\eta g_{t+1}[i]} \sum_{j=1}^{n} e^{-\eta g_{t+1}[j]} - \sum_{j=1}^{n} e^{-\eta g_t[j]} \sum_{j=1}^{n} e^{-\eta g_t[j]}}{\sum_{j=1}^{n} e^{-\eta g_{t+1}[j]} - \sum_{j=1}^{n} e^{-\eta g_t[j]}},
$$

where $c = \nabla R(v)[i]$. Further, the update can be formulated as

$$
w_{t+1}[i] = \frac{e^{-\eta g_{t+1}[i]} \sum_{j=1}^{n} e^{-\eta g_{t+1}[j]} - \sum_{j=1}^{n} e^{-\eta g_t[j]} \sum_{j=1}^{n} e^{-\eta g_t[j]}}{\sum_{j=1}^{n} e^{-\eta g_{t+1}[j]} - \sum_{j=1}^{n} e^{-\eta g_t[j]}},
$$

(51)

It completes the proof of (23).

Note that $w_{t+1}$ in (23) has the same closed-form update. In the case of interest, (30) is replaced with

$$
w_{t+1} = \Pi_{R,F} \left( \nabla R^* (\nabla R(w_t) - \eta g_t) \right).
$$

Then we have

$$
w_{t+1}[i] = \frac{e^{1+\ln w_t[i] - \eta g_t[i]} \sum_{j=1}^{n} e^{1+\ln w_t[j] - \eta g_t[j]} - \sum_{j=1}^{n} e^{1+\ln w_{t+1}[j] - \eta g_{t+1}[j]} \sum_{j=1}^{n} e^{1+\ln w_{t+1}[j] - \eta g_{t+1}[j]}}{\sum_{j=1}^{n} e^{1+\ln w_t[j] - \eta g_t[j]} - \sum_{j=1}^{n} e^{1+\ln w_{t+1}[j] - \eta g_{t+1}[j]}}.
$$

And $w_1 = v$ completes the proof. $\blacksquare$

F. Proof of Proposition 6

Proof: Due to the equivalence in Proposition 3 we start with (24). By (4.21) in [42], for any $\hat{w} \in F$ we have

$$
\ell_t(w_t) - \ell_t(\hat{w}) \leq \frac{B_R(\hat{w}, w_t) - B_R(\hat{w}, w_{t+1})}{\eta_t} + \frac{\eta_t}{2} g_t^2 R_{t-1:t}.
$$

For the normalized KL divergence, we have

$$
B_R(\hat{w}, w_t) - B_R(\hat{w}, w_{t+1}) = \sum_{i=1}^{n} \hat{w}[i] \ln \frac{w_{t+1}[i]}{w_{t}[i]},
$$

and using the fact that $R(\cdot)$ is strongly convex w.r.t. $\| \cdot \|_2^2$, we get $\| g_t \|_{R^*}^2 = \| g_t \|_2^2$. Hence we have

$$
\text{Regret}_T
\leq \frac{\sum_{t=1}^{T} g_t^2}{\eta_t} \left( \sum_{i=1}^{n} \hat{w}[i] \ln \frac{w_{t+1}[i]}{w_{t}[i]} \right) + \frac{\eta_t}{2} \sum_{t=1}^{T} \| g_t \|_{R^*}^2
\leq \frac{\sum_{t=1}^{T} g_t^2}{\eta_t} \left( \sum_{i=1}^{n} \hat{w}[i] \ln \frac{w_{t+1}[i]}{w_{t}[i]} \right) + \frac{\eta_t}{2} \sum_{t=1}^{T} \| g_t \|_{R^*}^2
\leq \frac{\sum_{t=1}^{T} \hat{w}[i] \ln \frac{w_{t+1}[i]}{w_{t}[i]} + \eta_t}{2} \sum_{t=1}^{T} \| g_t \|_{R^*}^2
\leq \frac{\sum_{t=1}^{T} \hat{w}[i] \ln \frac{w_{t+1}[i]}{w_{t}[i]} + \eta_t}{2} \sum_{t=1}^{T} \| g_t \|_{R^*}^2
\leq \frac{\sum_{t=1}^{T} \| g_t \|_{R^*}^2}{2} \sum_{t=1}^{T} \frac{1}{\eta_t}.
$$

Using the fact that $\sum_{t=1}^{T} \frac{1}{\eta_t} \leq 2\sqrt{T}$ and setting $\eta = \sqrt{\ln n}/G$, we obtain the bound (26). Similarly, use (18) once more and we get the regret (27). $\blacksquare$

G. Proof of Proposition 7

Proof: It directly follows from the equivalence in Proposition 2. A-FOBOS in (4) is equivalent to FTPRL in (28) with $v_T = v_T$ and $S_t = [t]$. $\Psi_t = \frac{1}{2} \| v_t \| A_t = R_{t-1:t}(w)$ where $A_t = aI + (\sum_{t=1}^{T} g_T g_T^T)^{1/2}$. Then $b = \frac{1}{2}, \ Q_T = g_T g_T^T$ and $R_T(w) = \frac{1}{2} \| v_T \|^2_{A_t A_t^{-1}}$. $\blacksquare$

H. Proof of Proposition 8

Proof: When $R_T(w) = \frac{1}{2} \| v_t \|^2_{Q_T}$, $B_R(w, w_t) = \frac{1}{2} \| w - w_t \|^2_{Q_t}$, then FTPRL in (2) is a specific case of FTBRL in (28) with $\eta = 1, v_T = w_T$ and $S_t = [t]$. $\blacksquare$

I. Proof of Regret (37)

Proof: In the case of interest the first term in regret (17) reduces to

$$
\frac{1}{2} \sum_{t=1}^{T} \sum_{i=1}^{n} (w[i] - w_t[i])^2 [(Q_{1:t-1})_{i,i} - (Q_{1:t})_{i,i}],
$$

Further, by the definition $D_\infty = \max_{x,y \in F} \| x - y \|_{\infty}$,

$$
\frac{D_\infty^2}{2} \sum_{t=1}^{T} \sum_{i=1}^{n} (Q_{1:t})_{i,i} - (Q_{1:t-1})_{i,i} = \frac{D_\infty^2}{2} \sum_{t=1}^{T} \sum_{i=1}^{n} (g_t[i])^2.
$$

Using inequality (18), the second term in regret (17) becomes

$$
\frac{D_\infty^2}{2} \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{(g_t[i])^2}{\sum_{t=1}^{T} (g_t[i])^2} = \frac{D_\infty^2}{2} \sum_{t=1}^{T} \sum_{i=1}^{n} (g_t[i])^2.
$$

Then add two terms together and we complete the proof. $\blacksquare$
J. Proof of Proposition 9

Proof: It can be seen that $\ell_r(w_r)$ and $g_t^\top w_r$ in $\mathbb{E}(w)$ of (8) are independent of the solution $w$. Then it can be removed and $\ell_r(w)$ reduces to

$$\frac{1}{2}w - w_r\|Q_w^* + g_r, w\|,$$

where $Q_w = g_r g_r^\top$. Hence, (8) can be replaced with

$$w_{t+1} = \max_{w \in \mathcal{F}} \left( \frac{1}{2}w - w_r\|Q_w^* + r(g_r, w) \right),$$

which is identical with (28) on setting $v_r = w_r$, $S_t = [t]$, $\eta = r$ and $R_r = \frac{1}{2}w\|Q_r^*.$

K. Proof of Proposition 10

Proof: As showed in [15], ONS is equivalent to FTAL and then, by Proposition 9, it can be recovered with FTDL in (28). Further, by the equivalence of MD and FTDL, ONS can be recovered with MD in [15].

To solve MD [15], we set $\nabla R_{1t}(w) = \nabla R_{1t}(w_t) - \eta g_r$, to get the optimal solution before projection. Using the fact that $\nabla R^*(\cdot) = (\nabla R(\cdot))^{-1}$, it can be updated in closed-form

$$w_{t+1} = R_{1t}, \nabla (\nabla R_{1t}(w_t) - \eta g_r).$$

When $R_{1t}(w) = \frac{1}{2}w\|A_t, R_{1t}(w) = \frac{1}{2}w\|A_t^{-1}$ and then

$$w_{t+1} = \Pi_{A_t} \left\{ w_t - \eta A_t^{-1} g_r \right\}.$$

L. Proof of Proposition 11

Proof: It is equivalent to prove $(aI_m + BA)^{1/2} = B(aI_n + AB)^{1/2}$. For simplicity, let $\tilde{A}_n := aI_n + AB$ and $\tilde{A}_m := aI_m + AB$. As we know $AB$ and $BA$ share the same nonzero eigenvalues, then $\delta(A_n) = \delta(A_m)$, where $\delta(\cdot)$ is the set of all the different eigenvalues. Let $s \leq \min(m, n)$ 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 $(\cdot)$.

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

$$(\tilde{A}_n)^{1/2} = P_1D(\lambda_1^2, \ldots, \lambda_s^2)P_1^{-1},$$

where $P_1$ is an orthogonal matrix, and for all $n \in \{\lambda_1, \ldots, \lambda_s\}$ and $\{\lambda_1, \ldots, \lambda_s\} \supseteq \{\lambda_1, \ldots, \lambda_s\}$. Similarly, we have

$$(\tilde{A}_m)^{1/2} = P_2D(\lambda_1^2, \ldots, \lambda_m^2)P_2^{-1},$$

where $P_2$ is an orthogonal matrix, and for all $m \in \{\lambda_1, \ldots, \lambda_m\}$ and $\{\lambda_1, \ldots, \lambda_m\} \supseteq \{\lambda_1, \ldots, \lambda_s\}$. That is,

$$h(\tilde{A}_n) = (\tilde{A}_n)^{1/2} = P_1D(h(\lambda_1), \ldots, h(\lambda_s))P_1^{-1},$$

and

$$h(\tilde{A}_m) = (\tilde{A}_m)^{1/2} = P_2D(h(\lambda_1), \ldots, h(\lambda_m))P_2^{-1}.$$