PROJECTION FREE DYNAMIC ONLINE LEARNING

Deepak S. Kalhan[*]*, Amrit S. Bedi*[†]*, Alec Koppel*[†]*, Ketan Rajawat*[*]*, Abhishek Gupta*[*]*, and Adrish Banerjee*[*]

[*]Department of Electrical Engineering, IIT Kanpur, Kanpur, India
[†] CISD, U.S. Army Research Laboratory, Adelphi, Maryland, USA

ABSTRACT

Projection based algorithms are popular in the literature for online convex optimization with convex constraints and the projection step results in a bottleneck for the practical implementation of the algorithms. To avoid this bottleneck, we propose a projection-free scheme based on Frank-Wolfe: where instead of online gradient steps, we use steps that are collinear with the gradient but guaranteed to be feasible. We establish performance in terms of dynamic regret, which quantifies cost accumulation as compared with the optimal at each individual time slot. Specifically, for convex losses, we establish $\mathcal{O}(T^{1/2})$ dynamic regret up to metrics of non-stationarity. We relax the algorithm's required information to only noisy gradient estimates, i.e., partial feedback and derived the dynamic regret bounds. Experiments on matrix completion problem and background separation in video demonstrate favorable performance of the proposed scheme.

***Index Terms*—** Online learning, Frank-Wolfe algorithm, convex optimization, gradient descent.

1. INTRODUCTION

Many learning problems may be formulated as complex data-dependent optimization problems, as in the design of methods for speech recognition [1], perception [2], and locomotion [3]. These technologies upend several orthodoxies in the design of optimization algorithms: finite time performance is prioritized, updates must be memory-efficient despite the scale of training sets, and drift in data distributions must be mitigated. Recently, online optimization has gained popularity as a way to meet these specifications in disparate contexts such as non-parametric regression [4, 5], portfolio management [6], control in robotics [7]. The framework of online optimization decomposes a complex problem into a sequence of sub-problems, which inherently arises when one operates on subsets of data per step due to the sheer scale of full training sets. Alternatively, in many problems, the cost is an expectation of a collection of loss functions parameterized by data only accessible via samples [8, 9].

In literature, central to online optimization is online gradient descent [14], whose static regret is $\mathcal{O}(T^{1/2})$. Improvements are possible for strongly convex losses [15], for a detailed review, see [13]. The constraint satisfaction at each time slot poses challenges: methods based on Lagrangian relaxation such as ADMM [16] or saddle point [17] cannot ensure feasibility of individual actions. In contrast, projections do so but require a quadratic problem to be solved at each step [18]. Frank-Wolfe (conditional gradient) method moves in a feasible direction that is collinear with the gradient through the solution of a linear program [19], and has gained attention recently as a way to avoid projections in online constrained settings [10, 20]. We build upon these successes to characterize the behavior of Frank-Wolfe method in non-stationary settings. For non-stationary learning problems, several works characterize sublinear growth of dynamic regret up to factors depending on V_T, D_T, and W_T, as detailed in Table 1. i.e., $\mathcal{O}(T^{1/2}(1 + W_T))$ for OGD or mirror descent with convex losses [14, 21], expressions that depend on multiple metrics of non-stationarity [22, 23], and improved rates $\mathcal{O}(1 + W_T)$ in strongly convex cases [24, 25]. Here, V_T, D_T, and W_T are the metrics of non-stationarity where V_T denotes the rate of change of objective function, D_T describes the rate of change of objective function gradient, and W_T defines the rate of change of the optimal value. However, all existing works in the literature execute projections, which owing to the complexity requirements, may prohibit them from yielding solutions in a timely fashion when data drifts, in contrast to Frank-Wolfe [10]. Furthermore, in practice, exact online gradients may be unavailable [26]. In this work, we take inspiration from [26] to propose such methods that may operate effectively in the presence of non-stationarity.

Contributions: In this work, we put forth a collection of online optimization schemes that obviate the need for projection and are robust to gradient estimation error, leveraging recently developed averaging techniques [27], and characterize their performance amidst non-stationarity. In particular: (i) We generalize Frank-Wolfe method to non-stationary problems (Sec. 3) and establish $\mathcal{O}(T^{1/2})$ dynamic regret upto metrics of non-stationarity when losses are convex (Sec. 4). (ii) We generalize the algorithm to the setting where we only have access to noisy estimates of online gradients (partial feedback, Sec. 3.1), and establish that its dynamic regret (Sec. 4.1). (iii) To show the efficacy of the proposed algorithm experimentally, we observe that Online Frank-Wolfe attain favorable performance relative to alternatives [13] on non-stationary matrix completion and background extraction in video (Sec. 5). In particular, Frank-Wolfe yields a significant reduction in the computational time, while attaining comparable performance, to existing approaches. In the experiments, we have also included a variance reduced version of the proposed algorithm

Reference	Loss function.	Step-size	Batch	Regret definition	Rate
[10]	$(L/D)t^{-1/4}$-strongly convex	diminishing	$\mathcal{O}(t)$	$\sum_{t=1}^{T} F(\mathbf{x}_t) - F(\mathbf{x}^\star)$	$\mathcal{O}\left(T^{3/4}\right)$
[11]	convex	diminishing	$\mathcal{O}(1)$	$\mathbb{E}\big[F(\mathbf{x}_T) - F(\mathbf{x}^\star)\big]$	$\mathcal{O}(1/T^{1/3})$
[12]	convex	depends on σ_2 & C_T	-	$\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T}\left[F_t(\mathbf{x}_{i,t}) - F_t(\mathbf{x}_t^\star)\right]$	$\mathcal{O}\left(\sqrt{\frac{(1+C_T)T}{1-\sigma_2(W)}}\right)$
[13]	1-strongly convex	diminishing	$\mathcal{O}(1)$	$\sum_{t=1}^{T}\left[F_t(\mathbf{x}_t) - F_t(\mathbf{x}^\star)\right]$	$\mathcal{O}\left(T^{3/4}\right)$
This work	convex	constant	$\mathcal{O}(1)$	$\sum_{t=1}^{T} F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^\star)$	$\mathcal{O}\left(\sqrt{T}\left(1 + V_T + \sqrt{D_T}\right)\right)$
This work	convex (partial feedback)	constant	$\mathcal{O}(1)$	$\sum_{t=1}^{T}\mathbb{E}\big[F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^\star)\big]$	$\mathcal{O}\left(1 + T^{\frac{5}{6}} + \sqrt{T}V_T + T^{\frac{5}{6}}\sqrt{D_T}\right)$

Table 1: Summary of the related works compared to the present work.

named as "Meta Frank-Wolfe" algorithm similar to the one proposed in [26] but for dynamic settings.

2. PROBLEM FORMULATION

In online convex optimization (OCO), at each time t, a learner selects an action $\mathbf{x}_t$ after which an arbitrary convex cost F_t is revealed. The standard performance metric for this setting is to compare the action sequence $\{\mathbf{x}_t\}_{t=1}^{T}$ up to some time-horizon T with a *single* best action in hindsight, defined as the regret $\mathbf{Reg}_T^S = \sum_{t=1}^{T} F_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} F_t(\mathbf{x})$. However, whenever training data defines trajectories, as is the case in increasingly salient learning problems in dynamical systems or reinforcement learning [28, 29], then hypothesizing that samples come from a stationary distribution is invalid. While the use of buffers experimentally sidestep this issue [30], rigorously addressing it requires treating learning as non-stationary stochastic optimization [22].

In general, this perspective requires tuning algorithms to mixing rates of the data distribution [31, 32], which substantially impact performance but mixing rates are typically unknown. Online optimization in the presence of non-stationarity avoids these difficulties by instead defining an alternative quantifier of performance called *dynamic regret*: the difference between the instantaneous cost accumulation and the cost of the best action at each time slot [23]

$$\mathbf{Reg}_T^D = \sum_{t=1}^{T} F_t(\mathbf{x}_t) - \sum_{t=1}^{T} \min_{\mathbf{x}\in\mathcal{X}} F_t(\mathbf{x}). \tag{1}$$

OCO concerns the design of methods such that $\mathbf{Reg}_T^D$ grows sublinearly in horizon T for a given sequence of loss function F_t, i.e., the average regret goes to null with T (no-regret [14]). Unfortunately, exactly tracking the optimizer defined by an arbitrarily varying optimization problem is impossible [22, 33], and the best one may hope for is to be competitive up to metrics of non-stationarity such as the loss variation V_T and gradient variation D_T defined as [22, 24]

$$V_T := \sum_{t=1}^{T} \sup_{\mathbf{x}\in\mathcal{X}} \mid F_t(\mathbf{x}) - F_{t-1}(\mathbf{x}) \mid, and$$
$$D_T := \sum_{t=1}^{T} \left\|\nabla F_t(\mathbf{x}_t) - \nabla F_{t-1}(\mathbf{x}_{t-1})\right\|^2. \tag{2}$$

Our goal in this work is the design of algorithms such that dynamic regret grows sublinearly in T up to multiplicative factors of the variable and gradient variations defined as V_T and D_T, i.e., $\mathbf{Reg}_T^D = o(T(V_T + D_T))$. Next, we describe the algorithm we proposed to minimize the dynamic regret defined in (1) for the online constrained optimization problems.

3. ONLINE FRANK-WOLFE ALGORITHM

We begin by deriving standard Frank-Wolfe (conditional gradient) algorithm adapted to the setting of online optimization. For time t, assuming that action $\mathbf{x}_t$ has been chosen and the instantaneous cost F_t is revealed, we may evaluate the online gradient as $\nabla F_t(\mathbf{x}_t)$. Based upon this information, we define directional vector d_t by the recursion:

$$\mathbf{d}_t = (1-\rho)\mathbf{d}_{t-1} + \rho\nabla F_t(\mathbf{x}_t) \tag{3}$$

with initial vector $\mathbf{d}_0 = 0$, and $\rho \in (0, 1]$ is a constant momentum parameter. The smoothing step (3) permits us to gracefully apply the algorithm to the more challenging setting of partial feedback or non-convex losses discussed later [11]. Then, we seek a direction $\mathbf{v}_t$ that is parallel to $\mathbf{d}_t$ inside feasible set $\mathcal{X}$, the source of the name *conditional* gradient. This is accomplished by solving the following linear program (LP)

$$\mathbf{v}_t = \arg\min_{\mathbf{v}\in\mathcal{X}} \langle \mathbf{d}_t, \mathbf{v}\rangle. \tag{4}$$

Then, the action $\mathbf{x}_{t+1}$ for subsequent time $t+1$ is given by

$$\mathbf{x}_{t+1} = (1-\gamma)\mathbf{x}_t + \gamma\mathbf{v}_t, \tag{5}$$

where $\gamma < 1$ is a time-invariant step-size. In the following subsection, we discuss a generalization to partial feedback. The method is summarized as Algorithm 1.

3.1. Partial Feedback

To implement the Algorithm 1, the exact gradient $\nabla F_t(\mathbf{x}_t)$ must be computed at each iteration t. In practice, this computation may be unavailable or prohibitively costly to obtain. For instance, in expected risk minimization [8], $\nabla F_t(\mathbf{x}_t)$ denotes the *full batch* gradient, which, if the number N of samples $\{\mathbf{z}_n\}_{n=1}^{N}$ in the training set is large, is costly to evaluate

Algorithm 1 Online Frank-Wolfe Algorithm (OFW)

Require step sizes $0 < \rho < 1$ and $0 < \gamma < 1$.
Initialize $t = 0$, $d_0 = 0$ and choose $x_0 \in \mathcal{X}$.
for t = 1,2.....**do**.
  Update gradient estimate $\mathbf{d}_t = (1-\rho)\mathbf{d}_{t-1} + \rho\nabla F_t(\mathbf{x}_t)$
  Compute $\mathbf{v}_t = \arg\min_{\mathbf{v}\in\mathcal{X}}\langle \mathbf{d}_t, \mathbf{v}\rangle$
  Update $\mathbf{x}_{t+1} = (1-\gamma)\mathbf{x}_t + \gamma\mathbf{v}_t$
end for

[34,35]. Alternatively, one may simply receive only noisy samples of the gradient, but not its true value, as is the case with received signal strength-based localization [36] or learned models of mismatched kinematics in optimal control [37]. For such situations, only a noisy estimate $\nabla f_t(\mathbf{x}_t, \mathbf{z}_t)$ of the online gradient $\nabla F_t(\mathbf{x}_t)$ is available such that $\nabla F_t(\mathbf{x}) = \mathbb{E}\left[\nabla f_t(\mathbf{x}, \mathbf{z}_t)\right]$. Here $\mathbf{z}_t$ denotes a realization of random variable $\mathbf{z}$ that parameterizes the noisy online gradient.

For example, consider the problem of online matrix completion, which seeks the best possible low rank approximation of a given matrix $\mathbf{M_t} \in \mathbb{R}^{m\times n}$. Denote as $\mathbf{X}_t \in \mathbb{R}^{m\times n}$ the low-rank approximation. In each round, the entries of matrix dented as $(i,j) \in \text{OB}_t$ where OB_t is the batch of new entries, are updated. The problem is then defined as [13, Chap. 7]

$$\min_{\mathbf{X}_{ij}} \sum_{(ij)\in \text{OB}_t} (\mathbf{X}_{ij} - \mathbf{M}_{ij,t})^2 \quad \text{such that } \|\mathbf{X}\|_* \leq k. \tag{6}$$

In practice, one observes the entries estimates $\{\hat{\mathbf{M}}_{ij,t}\}$ in an online manner such that $\hat{\mathbf{M}}_{ij,t} = \mathbf{M}_{ij,t} + \mathbf{z}_t$, where $\mathbf{z}_t$ is the stochastic error in the matrix entries estimation. The true value $\mathbf{M}_{ij,t}$ is unknown, and hence only partial feedback is available. The parameter $\rho \in (0,1)$ is used to track the gradient estimate for the partial feedback settings. The error in gradient estimate over the entire time horizon is bounded asymptotically if we choose ρ properly. We note that if we select $\rho = 1$, i.e., use stochastic gradients, then the gradient estimation error over entire time horizon diverges due to the variance of estimates.

4. DYNAMIC REGRET ANALYSIS

In this section, we characterize the performance of Algorithm 1 in the presence of non-stationarity as quantified by dynamic regret. First, we state some required technical assumptions.

[**A1**.] The set $\mathcal{X}$ is convex and compact with diameter D, i.e., for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, it holds that $\|\mathbf{x} - \mathbf{y}\| \leq D$.

[**A2**.] The gradient of loss $\nabla F_t(\cdot)$ is Lipschitz with parameter L_1, which implies that $\|\nabla F_t(\mathbf{x}) - \nabla F_t(\mathbf{y})\| \leq L_1 \|\mathbf{x} - \mathbf{y}\|$ for all t and $(\mathbf{x}, \mathbf{y}) \in \mathcal{X}$.

The Assumptions **A1**-**A2** are standard in online learning [11,24]. Assumption **A1** ensures constrained set $\mathcal{X}$ is compact. Assumption **A2** bounds the loss function gradient. Next, we present the dynamic regret result of Algorithm 1.

Theorem 1 *Under the Assumptions **A1**-**A2**, for the iterates generated by Algorithm 1, under step-size selection $\gamma = \frac{1}{\sqrt{T}}$, it holds that*

$$\textbf{\textit{Reg}}_T^D \leq \mathcal{O}\left(\sqrt{T}\left(1 + V_T + \sqrt{D_T}\right)\right). \tag{7}$$

Theorem 1 (see [38] for proof) establishes convergence of Algorithm 1 for non-stationary problems in terms of dynamic regret up to factors depending on V_T and D_T, as defined in Section 1. This is the first time a projection-free scheme has been demonstrated as theoretically effective for dynamic learning problems, which paves the way for use in applications with data drift across time. Note, however, that Algorithm 1 requires exact gradient information at each step, which in applications to learning online with estimation errors such as in (6), may be unavailable, motivates the partial feedback setting which we analyze next.

4.1. Regret Analysis under Partial Feedback

To analyze performance in when feedback is partial, before proceeding, we state an additional required assumption that limits the variance of stochastic approximation error.

[**A3**.] The variance of the unbiased stochastic gradients $\nabla\tilde{F}_t(\mathbf{x},\ \mathbf{z})$ is bounded above by σ^2, implies that $\mathbb{E}\left[\ \|\nabla f_t(\mathbf{x}, \mathbf{z}) - \nabla F_t(\mathbf{x})\|^2\ \right] \leq \sigma^2$ for all t.

Theorem 2 *Under the Assumptions **A1**-**A3**, for the iterates generated by Algorithm 1, the following expected dynamic regret bounds hold:*

$$\sum_{t=1}^{T}\Big[\mathbb{E}\left[F_t(\mathbf{x}_t)\right] - F_t(\mathbf{x}_t^\star)\Big] \leq \mathcal{O}\left(1 + T^{\frac{5}{6}} + \sqrt{T}V_T + T^{\frac{5}{6}}\sqrt{D_T}\right) \tag{8}$$

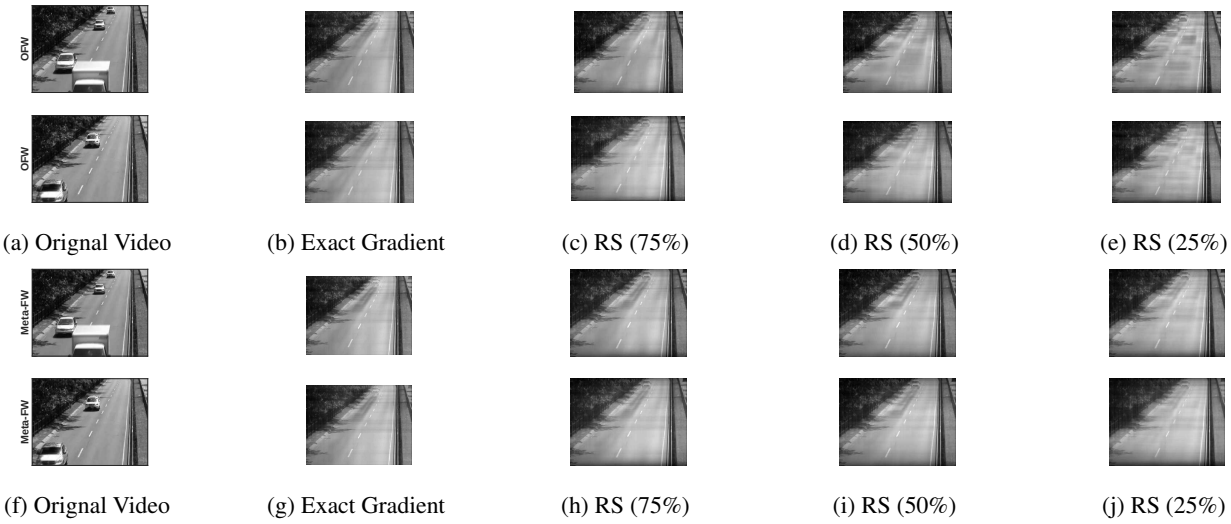
under step-size and inertia selections $\gamma = \frac{1}{\sqrt{T}}$, $\rho = \frac{1}{T^{1/3}}$.

Theorem 2 (see [38] for proof) establishes that the dynamic regret for Algorithm 1 is sublinear despite only having access to noisy estimates of online gradients, given appropriate stepsize and averaging parameter selections.

5. EXPERIMENTS

In this section, we experimentally evaluate the proposed algorithm on matrix completion and background subtraction in video, both of which demonstrate the merits of online Frank-Wolfe. In particular, we observe a favorable tradeoff between complexity and accuracy by virtue of avoiding computationally costly projections.

Online Matrix Completion: We solve (6) using Algorithms 1 and compare performance with alternatives such as OGD which requires projections in Fig. 2a. We have presented the results for both exact as well as partial gradient feedback (we call it inexact gradient). For the experiments, we have also included a variance reduced version of the proposed algorithm called "Meta-Frank Wolfe" which improves the result as



(a) Orignal Video (b) Exact Gradient (c) RS (75%) (d) RS (50%) (e) RS (25%)

(f) Orignal Video (g) Exact Gradient (h) RS (75%) (i) RS (50%) (j) RS (25%)

Fig. 1: Background Extraction Problem: $1st$ (OFW) and $3rd$ (Meta-Frank Wolfe) row represents results for instant 1 of the video; the $2nd$ (OFW) and 4th (Meta-Frank Wolfe) row represent instant 2 of the video, which is clear from the $1st$ column. In the figure, RS denotes random sampling and the percentage denotes how many samples from the full gradient are utilized for the algorithm updates. The proposed algorithm performs really well for this application since the cars are effectively removed from the frame, as clear from 2nd to 5th columns. Meta-Frank Wolfe is a variance reduced version of the proposed OFW algorithm.

Algorithm	Exact Gradient	RS(75%)	RS(50%)	RS(25%)
OFW	4.6436	4.2325	3.1949	3.1396
Meta-FW	26.5808	26.5810	22.8794	21.1206

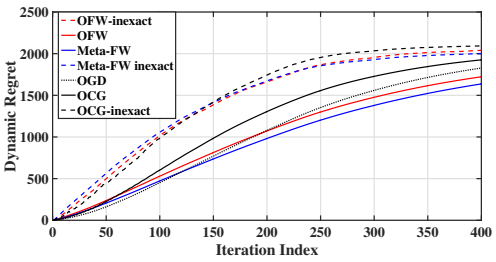
Table 2: Summary of computation time in seconds.

given in Fig. 2a. For OFW-inexact and Meta-FW-inexact, we have considered only 25% of the samples full gradient from random locations at each iteration. As presented in Fig. 2a, OFW performs better than online conditional gradient (OCG), a projection-free algorithm of [13] when full gradient information is available. We remark that the Meta-FW algorithm performs best among all the algorithms when full gradient is available. A similar behavior is observed with the partial information availability too. Also Fig. 2b shows that OGD is the slowest as compared to all the algorithms due to the required projection. For the experiments, we have considered $m = n = 20$. To implement the Meta-Frank Wolfe algorithm, we fix $K = 30$.

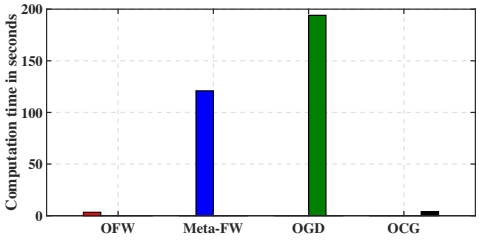
Background extraction problem: In this experiment, we extend the matrix completion problem on real dataset from [39]. At each instant we observe a video frame and collect it into matrix $\mathbf{M}_t$. The goal of the problem is to extract the background from the video which is conceptually the low-rank estimate $\mathbf{M}_t$ of the data matrix. The problem is then given as:

$$\min_{\mathbf{L}_t} \|\mathbf{M}_t - \mathbf{L}_t\|_F^2 + \frac{1}{2}\|\mathbf{L}_t\|_F^2 \text{ such that } \|\mathbf{L}_t\|_* \leq k. \quad (9)$$

The results in Fig. 1 are generated using OFW with different samples of gradient at different instants. Note that online Frank-Wolfe yields effective performance for this application as demonstrated in Fig. 1 – the cars are removed from the frame. We describe the effectiveness of using random sampling (RS) which makes the gradient inexact by summarizing the execution times in Table 2, where we observe thatn increasing RS yields quick completion. The proposed OFW algorithm performs well for the background subtraction application since the cars are completely removed from the frames, as clear from 2nd to 5th columns. Please see the video appended to the submission to observe Frank-Wolfe implementing online background subtraction.



(a)



(b)

Fig. 2: (a) Comparison of dynamic regrets of different algorithms for the matrix completion. (b) Runtime comparison of Frank-Wolfe and Meta Frank-Wolfe compared to alternatives on matrix complication.

6. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[4] D. Calandriello, A. Lazaric, and M. Valko, "Second-order kernel online convex optimization with adaptive sketching," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 645–653.

[5] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 83–126, 2019.

[6] A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire, "Algorithms for portfolio management based on the newton method," in *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 2006, pp. 9–16.

[7] N. Wagener, C.-A. Cheng, J. Sacks, and B. Boots, "An online learning approach to model predictive control," *arXiv preprint arXiv:1902.08967*, 2019.

[8] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer Series in Statistics New York, 2001, vol. 1.

[9] A. Shapiro, D. Dentcheva *et al.*, *Lectures on Stochastic Programming: Modeling and Theory*. Siam, 2014, vol. 16.

[10] E. Hazan and S. Kale, "Projection-free online learning," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ser. ICML'12. USA: Omnipress, 2012, pp. 1843–1850. [Online]. Available: http://dl.acm.org/citation.cfm?id=3042573.3042808

[11] A. Mokhtari, H. Hassani, and A. Karbasi, "Stochastic conditional gradient methods: From convex minimization to submodular maximization," *arXiv preprint arXiv:1804.09554*, 2018.

[12] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2018.

[13] E. Hazan *et al.*, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[14] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. on Machine Learning*, vol. 20, no. 2, Washington DC, USA, Aug. 21-24 2003, pp. 928–936.

[15] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, 2007.

[16] H. Wang and A. Banerjee, "Online alternating direction method," in *Proceedings of the 29th International Conference on Machine Learning*. Omnipress, 2012, pp. 1699–1706.

[17] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 13, no. Sep, pp. 2503–2528, 2012.

[18] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[19] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[20] E. Hazan and H. Luo, "Variance-reduced and projection-free stochastic optimization." in *Proceesing of ICML*, vol. 16, 2016, pp. 1263–1271.

[21] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[22] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations Research*, vol. 63, no. 5, pp. 1227–1244, 2015. [Online]. Available: https://doi.org/10.1287/opre.2015.1408

[23] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Artificial Intelligence and Statistics*, 2015, pp. 398–406.

[24] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 7195–7201.

[25] A. S. Bedi, A. Koppel, K. Rajawat, and B. M. Sadler, "Nonstationary nonparametric online learning: Balancing dynamic regret and model parsimony," *arXiv preprint arXiv:1909.05442*, 2019.

[26] L. Chen, C. Harshaw, H. Hassani, and A. Karbasi, "Projection-free online optimization with stochastic gradient: From convexity to submodularity," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmssan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 814–823. [Online]. Available: http://proceedings.mlr.press/v80/chen18c.html

[27] A. Mokhtari, H. Hassani, and A. Karbasi, "Conditional gradient method for stochastic submodular maximization: Closing the gap," in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1886–1895.

[28] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[29] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

[30] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[31] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*. Springer, 2009, vol. 48.

[32] M. Mohri and A. Rostamizadeh, "Stability bounds for stationary φ-mixing and β-mixing processes," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 789–814, 2010.

[33] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4576–4591.

[34] A. Mokhtari, A. Koppel, and A. Ribeiro, "A class of parallel doubly stochastic algorithms for large-scale learning," *Journal of Machine Learning Research (submitted)*, 2016.

[35] A. Koppel, A. Mokhtari, and A. Ribeiro, "Parallel stochastic successive convex approximation method for large-scale dictionary learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2771–2775.

[36] G. Wang, H. Chen, Y. Li, and M. Jin, "On received-signal-strength based localization with unknown transmit power and path loss exponent," *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 536–539, 2012.

[37] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6059–6066.

[38] "Technical report for "projection free dynamic online learning"." [Online]. Available: https://tinyurl.com/y2e9ntxo

[39] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection. net: A new change detection benchmark dataset," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 1–8.