Online convex optimization in wireless networks and beyond: The feedback -performance trade-off

Elena Veronica Belmega, Panayotis Mertikopoulos, Romain Negrel

Online convex optimization in wireless networks and beyond: The feedback – performance trade-off

E. Veronica Belmega[*†], Panayotis Mertikopoulos[‡§], and Romain Negrel[*]
[*] Univ. Gustave Eiffel, CNRS, LIGM, F-77454, Marne-la-Vallée, France
[†] ETIS UMR 8051, CY Cergy Paris Université, ENSEA, CNRS, F-95000, Cergy, France
[‡] Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France
[§] Criteo AI Lab, Grenoble, France

***Abstract*—The high degree of variability present in current and emerging mobile wireless networks calls for mathematical tools and techniques that transcend classical (convex) optimization paradigms. The aim of this short survey paper is to provide a gentle introduction to online learning and optimization algorithms that are able to provably cope with this variability and provide policies that are asymptotically optimal in hindsight – a property known as *no regret*. The focal point of this survey will be to delineate the trade-off between the information available as feedback to the learner, and the achievable regret guarantees – starting with the case of gradient-based (first-order) feedback, then moving on to value-based (zeroth-order) feedback, and, ultimately, pushing the envelope to the extreme case of a single bit of feedback. We illustrate our theoretical analysis with a series of practical wireless network examples that highlight the potential of this elegant toolbox.**

***Index Terms*—Online optimization; online learning; regret minimization; multi-armed bandits; feeback reduction.**

I. Introduction

Motivated by the highly dynamic nature of future and emerging wireless networks (e.g., B5G, 6G, Internet of Things), online optimization methods have been successfully exploited to design resource allocation policies for problems ranging from signal covariance optimization in multi-antenna terminals [1], channel selection and cognitive medium access [2] to network design and management [3].

In classic optimization, the core underlying assumption is that the objective to be optimized is known by the optimizing agent and remains fixed for the entire runtime of the algorithm computing a solution. Stochastic optimization provides an extension of this framework to problems where the objective function may also depend on a stationary stochastic process. Game theory takes an alternative, multi-agent view of such problems, often revolving around worst-case guarantees against an adversary. However, all these extensions rely on strong assumptions regarding the variability of the problem's objective, the agents' rationality and common knowledge of rationality (in games), the information at the agents' disposal.

By contrast, online optimization provides an elegant toolbox which goes beyond the above by allowing for variations in the problem that are *completely arbitrary* – typically accounting for exogenous (stationary or otherwise) parameters affecting it (cf. Fig. 1). An example of this framework is encountered in wireless communication networks, where traditional static/stationary paradigms no longer suffice to capture and cope with issues such as non-random fluctuations in the wireless medium, the unpredictable behavior of the communicating devices in distributed networks, Internet of things (IoT), etc. [1, 4]. Likewise, in multimedia indexing problems, the online framework allows to exploit distributed annotations from multiple sources (such as Internet users), as opposed to relying on centralized annotation of very large data sets, and also to relax all the stochastic assumptions among annotations that are common in offline metric learning [5].

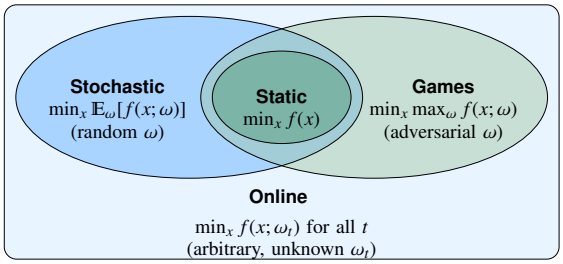

Fig. 1: A bird's eye view of the links between online optimization and other optimization paradigms.

In addition to its wide scope, another major advantage of the online framework is that the derived algorithms – referred here as *online policies* – come with **provable theoretical guarantees in the face of uncertainty**. These guarantees are based on Hannan's seminal notion of regret [6], which compares an algorithm's performance to that of the best fixed policy in hindsight. As such, the aim of online optimization and related literature is to derive causal, online policies leading to **no regret** and achieving the best possible regret minimization rates in a broad range of problems and with the fewest possible assumptions.

This paper aims to provide a light overview of online optimization and no-regret learning, including the theory's lower bounds, the algorithms that achieve them, and their applications in wireless communications and beyond. We pay specific attention here on algorithms with reduced feedback requirements, and the trade-offs involved in their performance. To that end, we begin by introducing the online first-order or gradient-based descent algorithms. Then, we move to zeroth-

This work has been supported in part by the ELIOT ANR-18-CE40-0030 and FAPESP 2018/12579-7 project.

order algorithms that only require the observed value of the objective (and not its gradient). Based on this value, a stochastic gradient approximation is built to enable the use of gradient-like methods that attain no regret at the cost of a slower regret minimization rate. Finally, we discuss a series of cases where it is possible to learn with a *single* bit of feedback that is jointly exploited by a multi-armed bandit algorithm to derive efficient online policies in wireless networks.

II. Online Convex Optimization: preliminaries

The core online optimization framework is as follows: At each stage $t = 1, 2, \ldots$, the optimizing agent selects an action x_t from some compact convex set $\mathcal{X} \subseteq \mathbb{R}^d$ and incurs a loss of $\ell_t(x_t)$ based on some (a priori unknown) loss function $\ell_t \colon \mathcal{X} \to \mathbb{R}$ (as in the bandit case, this loss function could be determined stochastically, adversarially, or otherwise). Subsequently and based on some *strictly causal feedback* information v_t about the incurred loss, the agent selects a new action x_{t+1} for the next stage and the process repeats as shown below:

Process 1: Generic online decision process

Require: action set $\mathcal{X}$, sequence of loss functions $\ell_t \colon \mathcal{X} \to \mathbb{R}$
1: **for** $t = 1$ **to** T **do**
2: play $x_t \in \mathcal{X}$ # action selection
3: incur $\ell_t(x_t)$ # incur loss
4: observe v_t # receive feedback
5: update $x_{t+1} \leftarrow x_t, v_t$ # update action based on feedback
6: **end for**

In this paper, we will focus on a fundamental class of online optimization problems: *online convex optimization*, in which $\ell_t \colon \mathcal{X} \to \mathbb{R}$ is convex for all t. In particular, we make *no assumptions in terms of the underlying dynamics* of the loss function: $\ell_t(x)$ can vary in a completely arbitrary manner with respect to time t (including non-stationary or adversarial), going beyond the scope of stochastic optimization or *Markov decision process* (MDP) reinforcement learning.

A. Regret performance metric

When designing an online policy $x_t, \forall t \in \{1, \ldots, T\}$ an ideal goal would be to track and close the gap with the optimal instantaneous policy:

$$\ell_t(x_t) - \min_{x \in \mathcal{X}} \ell_t(x), \quad \forall t. \tag{1}$$

Of course, this is not achievable in general by strictly causal online processes, unless some restrictive assumptions are made regarding the temporal variability of the losses. For this reason, we will instead focus on a less ambitious target, the agent's *regret* [6]

$$R_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x), \tag{2}$$

defined as the difference between the aggregate loss incurred by the agent after T stages and that of the best action in hindsight. In other words, the agent's regret contrasts the performance of the agent's online policy x_t to that of the optimum fixed action $x^* \in \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x)$, which minimizes the total incurred loss over the given horizon of play.

Of course, this optimum action x^* cannot be calculated if the loss functions encountered by the optimizer are not known in advance. Consequently, the figure of merit in an online optimization problem is to design a strictly causal, online policy that achieves *no regret*, i.e., $R_T = o(T)$.

Online optimization focuses on designing algorithms that attain the best possible regret minimization rate, not only in terms of the horizon of play T, but also in terms of the multiplicative constants that come into play, and which depend on the geometry and dimensionality of the optimizer's feasible set $\mathcal{X}$, as detailed in the next sections.

Before, we discuss below the links with static and stochastic optimization and also with *multi-armed bandits* (MABs) from reinforcement learning.

B. Link with static optimization

A static or classic optimization problem can be stated as

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{X}. \end{aligned} \tag{Opt}$$

Viewed as an *online* optimization problem, this corresponds to facing the same loss function $\ell_t = f$ at each stage. In this case, if x_t is a no-regret policy and f is convex, an immediate application of Jensen's inequality shows that the so-called "ergodic average" $\bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t$ enjoys the guarantee

$$f(\bar{x}_T) - \min f \leq \frac{1}{T} \sum_{t=1}^{T} [f(x_t) - \min f] \leq \frac{R_T}{T}. \tag{3}$$

Thus, $\bar{x}_T$ converges to the solution set of (Opt) as $T \to \infty$; moreover, the rate of this convergence is controlled by the regret minimization rate of x_t. This key property of no-regret policies has been the cornerstone of a vast literature on fast optimization algorithms; for an overview, see [7].

As an application, in Section IV-B, online policies are compared to classical iterative water-filling algorithms in static multiple-input and multiple-output (MIMO) systems composed of rate-driven multiple interfering users [8, 9].

C. Link with stochastic optimization

A stochastic optimization problem is

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}_\omega f(x; \omega) \\ \text{subject to} \quad & x \in \mathcal{X} \end{aligned} \tag{Opt-S}$$

where $f(x; \omega)$ is a stochastic objective function that depends on a random variable ω. The expectation in (Opt-S) is usually very difficult to compute so, when designing algorithms to solve (Opt-S), it is assumed that f is sampled at an independent and identically distributed (i.i.d.) realization ω_t of ω at each iteration. For instance, in offline metric learning for multimedia indexing (see also Section VI), all examples are readily available in a training dataset, but they cannot be exploited simultaneously because of their prohibitive number and size [10]. Instead, much smaller random population samples are drawn and exploited iteratively, despite the fact that

the stochastic average of this sampling procedure cannot be computed.

From an online viewpoint, this corresponds to a sequence of loss functions of the form $\ell_t(x) = f(x;\omega_t)$. Similarly to static optimization, we can use again the Jensen's inequality argument to see that see that $\mathbb{E}[f(\bar{x}_T;\omega)] - \min \mathbb{E}[f] \leq \frac{\bar{R}_T}{T}$, where $\bar{R}_T = \sum_{t=1}^{T} \mathbb{E}[f(x_t;\omega_t)] - \min \mathbb{E}[f]$ denotes the pseudo-regret and represents the figure of merit in stochastic settings.

III. Multi-armed bandits

The generic online decision process in Section II includes a fundamental *discrete choice* problem, i.e., *multi-armed bandits* (MABs) from reinforcement learning, that has been the focus of a very active and vigorous literature and also provides a gentle introduction to the exploration vs. exploitation trade-off that underlies much of online optimization, hence, deserves a dedicated section.

Introduced by Thompson [11] and Robbins [12], a multi-armed bandit problem can be described as follows: At each stage $t = 1, 2, \ldots$, of a repeated decision process, an optimizing *agent* (the decision-maker) selects an *action* a_t from some finite set $\mathcal{A} = \{1, \ldots, A\}$. Based on this choice, the agent incurs a reward (or loss) $u_t(a_t) = -\ell_t(a_t)$, they select a new action a_{t+1} and the process repeats. [1] Originally the agent was a gambler choosing a slot machine in a casino (a "one-armed bandit"), and its reward was the amount of money received minus the cost of playing [12]. In clinical trials [11], the choice of action represents the drug administered to a test patient and the incurred loss is the patient's time to recovery. Wireless communication examples are provided in Section V-B.

It is easy to see that the optimizer faces a trade-off between **exploration and exploitation.** On the one hand, by "*exploring*" more arms, the agent obtains more information and can make better choices in the future. On the other hand, in so doing, the agent fails to "*exploit*" arms that yield better payoffs now, thus lagging behind in terms of performance. Achieving and maintaining a balance between exploration and exploitation is the main objective of the literature on MABs.

Two main classes of MABs are discussed next based on the way the sequence of losses is generated.

A. Stochastic bandits

Assume that the reward at each stage $u_t(a)$ of the a-th arm is an i.i.d. random variable $v_{a,t}$ drawn from a statistical distribution P_a. The arms' reward distributions are not known to the agent, so the objective is to identify and *exploit* the arm with the highest mean reward in as few trials as possible.

A straightforward idea, known as "follow the leader", would be to keep a running average of the losses obtained by each arm and then play the arm with the best past average reward. This *pure exploration* policy is a reasonable first try but it can easily get stuck at a suboptimal arm: if the best arm performs very badly in its first draws, it won't be drawn again in the future. This highlights the need for adding at least *some* exploration into the mix.

Building on this, the landmark idea of [13, 14] was to *retain optimism in the face of uncertainty,* i.e., to construct an *"optimistic"* estimate for the mean payoff of each arm, and then pick the arm with the highest such estimate. The resulting *upper confidence bound* (UCB) algorithm with tuning parameter $\alpha > 2$ is defined via the recursion

$$a_{t+1} = \arg\max_{a\in\mathcal{A}} \left\{ \hat{\mu}_{a,t} + \sqrt{\frac{\alpha \log t}{2n_{a,t}}} \right\}, \tag{4}$$

where

$$\hat{\mu}_{a,t} = \frac{1}{n_{a,t}} \sum_{s=1}^{t} \mathbb{1}(a_s = a)\, v_{a,s} \tag{5}$$

denotes the empirical mean payoff of arm a and $n_{a,t}$ is the number of times arm a has been chosen so far. Heuristically, the first term in (4) drives the agent to exploit the arm with the highest empirical mean while the second one encourages exploration by giving a second chance to arms which have not been played often enough (i.e., $n_{a,t}$ is small relative to t).

For the specific variant considered here with parameter $\alpha > 2$, the analysis of [15] gives a logarithmic pseudo-regret $\bar{R}_T = \mathcal{O}(\log T)$. Importantly, the regret guarantee of UCB is optimal in T and no causal policy played against a bandit with Bernoulli reward distributions can achieve regret lower than $\Omega(\log T)$ [13].

B. Adversarial bandits

When the problem at hand is not purely statistical in nature (or when its statistics are influenced by exogenous, contextual factors), UCB can be brought to a halt. In fact, **Cover's impossibility result** [16] states that: *no deterministic algorithm can hope to achieve sublinear regret against an adversarial bandit.*

Tracing its roots to the "rigged casino" problem of [17], this paradigm is known as *adversarial* because the agent is called to learn against *any* possible sequence of rewards, including those designed by a mechanism that actively tries to minimize the agent's cumulative payoff over time. More precisely, at each step, the rewards $u_t(a) = v_{a,t}$ of each arm $a \in \mathcal{A}$ are determined by the adversary simultaneously with the agent's action a_t, possibly with full knowledge of the decision process employed by the agent at step t.

A key idea in balancing exploration vs. exploitation in this setting is to keep a cumulative score of the performance of each arm and then employ a *random* arm drawn with probability x_t that is exponentially proportional to this score, leading to the so-called *exponential weights* (EW) algorithm:[2]

$$\begin{aligned} y_{t+1} &= y_t + \gamma v_t, \\ x_{t+1} &= \Lambda(y_{t+1}), \end{aligned} \tag{6}$$

[1] As opposed to other reinforcement learning frameworks (e.g., MDP and dynamic programming), in MABs, no explicit notion of environment state (nor stochastic state transitions) is taken into account and the agent's decisions are solely reward-driven.

[2] Other variants of EW are the multiplicative weights (MW), the exponential-weight algorithm for exploration and exploitation (EXP3), etc.

where the *logit choice map* $\Lambda\colon \mathbb{R}^A \to \Delta(\mathcal{A})$ is given by

$$\Lambda(y) = \frac{(\exp(y_a))_{a\in\mathcal{A}}}{\sum_{a\in\mathcal{A}} \exp(y_a)}. \tag{7}$$

By carefully tuning the learning rate which trades off between exploration vs. exploitation: $\gamma = \sqrt{2\log A/T}$, the regret is shown in [17] to be sublinear $\bar{R}_T = \mathcal{O}\Big(\sqrt{T\log A}\Big)$. Moreover, the regret of any causal algorithm with full information (i.e., the vector of all arms' rewards v_t is known and not just the reward of the chosen arm) is bounded from below as $\bar{R}_T = \Omega\Big(\sqrt{T\log A}\Big)$ [18].

C. Link with online optimization

Let $\mathcal{X} = \Delta(\mathcal{A}) = \{x \in \mathbb{R}_+^A : \sum_{a\in\mathcal{A}} x_a = 1\}$ denote the simplex of probability distributions over the set of arms of an MAB. If the agent uses mixed strategy $x_t \in \mathcal{X}$ at stage t and the bandit's reward vector is v_t, the agent's mean reward will be linear $\ell_t(x_t) = -v_t^\top x_t$, falling thus under the umbrella of *online convex optimization*.

To sum up, for MAB problems, deterministic policies suffice to achieve a logarithmic regret in stochastic environments. However, when playing against an informed *adversary*, such algorithms are doomed to fail: only randomized algorithms can attain a no-regret state. Moreover, there is a gap of $\Omega(\sqrt{T}/\log T)$ between stochastic and adversarial bandits, even with full information for the latter. For more in-depth discussions, see our extended online version [19].

IV. First-order (gradient) feedback

As an illustrating first example, consider the distributed IoT network in Fig. 2 composed of multiple transmitters communicating to their receivers over S orthogonal frequency bands (OFDM), investigated in [20]. Each transmitter aims at minimizing their power consumption provided that a minimum target rate is achieved: $\ell_t(p) = \sum_{s=1}^S p(s) + \lambda\,[R_{\min} - R_t(p)]^+$, where $p = (p(1),\dots,p(S))$ is the power allocation vector over the S subcarriers. The Shannon achievable rate equals $R_t(p) = \sum_{s=1}^S \log(1 + w_t(s)p(s))$ where w_t represents the effective channel vector of entries $w_t(s) = \frac{g_t(s)}{\sigma^2 + \sum_j g_{j,t}(s)p_{j,t}(s)}$, $\forall s$. The feasible set of power vectors is $\mathcal{X} = \Big\{p \in \mathbb{R}^S \;\Big|\; p(s) \geq 0, \forall s,\ \sum_{k=1}^S p(k) \leq P_{\max}\Big\}$ as per usual.

This convex optimization problem is relatively easy if w_t is known at the transmitter. However, since w_t encompasses the effects of the wireless medium (noise, pathloss, device mobility) and depends on the transmit characteristics of all interfering users (which may go on- and off-line in an ad-hoc manner), it may completely unpredictable and not known in advance, calling for online optimization algorithms.

A. Online gradient descent

The most popular and straightforward approach for solving classic, offline optimization problems is based on (projected) gradient descent: at each stage, the algorithm takes a step against the gradient of the objective and, if necessary, projects back to the problem's feasible region. Dating back to the seminal work of Zinkevich [21], *online gradient descent* (OGD) is the direct adaptation of this idea to an online context. In particular, writing $v_t = -\nabla\ell_t(x_t)$ for the negative gradient of the t-th loss function sampled at the agent's chosen action (again at round t), OGD can be described via the recursion

$$x_{t+1} = \Pi(x_t + \gamma v_t), \tag{8}$$

where $\Pi(y) = \arg\min_{x\in\mathcal{X}} \|y - x\|^2$ denotes the (Euclidean) projector and $\gamma > 0$ is a step-size parameter (see also Algorithm 2).

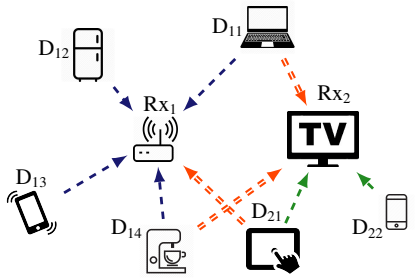

Fig. 2: IoT network composed of six transmit devices (D_{11}, D_{12}, etc.) and two receivers (Rx_1, Rx_2). The blue and green arrows represent the direct links while the red (double-lined) ones are interfering links [20].

Algorithm 2: online gradient/mirror descent (OGD/OMD)

Require: regularizer $h\colon \mathcal{X} \to \mathbb{R}$ # for OMD only
step-size $\gamma > 0$
choose $x \in \mathcal{X}$ # initialization
for $t = 1$ **to** T **do**
  incur loss $\hat{\ell} \leftarrow \ell_t(x)$ # losses revealed
  observe $v \leftarrow -\nabla\ell_t(x)$ # gradient feedback
  play $x \leftarrow \Pi(x + \gamma v)$ or $x \leftarrow P_x(\gamma v)$ # OGD or OMD
end for

Theorem 1 (Worst-case regret of OGD [21])**.** *Against L-Lipschitz convex losses, the OGD algorithm with step-size* $\gamma = (\mathrm{diam}(\mathcal{X})/L)/\sqrt{T}$ *enjoys the regret bound*

$$R_T \leq \mathrm{diam}(\mathcal{X}) L\sqrt{T}, \tag{9}$$

where $\mathrm{diam}(\mathcal{X}) \equiv \max_{x,x'\in\mathcal{X}} \|x' - x\|$ *is the diameter of* $\mathcal{X}$.

In the absence of stronger assumptions on the curvature of the losses, the above bound is tight and $R_T = \Omega(\sqrt{T})$. In the above example, OGD yields $R_T \leq \sqrt{S\,T}$ and the same performance is achieved also in the adversarial MABs setting of Section III.

B. Online mirror descent

In adversarial MABs, the above highlights an important hidden gap in the problem's geometry. Indeed, the EW algorithm yields $R_T = \mathcal{O}(\sqrt{\log A\ T})$, which scales much better than OGD in terms of the problem dimension A. Recovering logarithmic scalability is crucial in many Big Data and wireless communications applications. For instance, in massive MIMO networks the problem dimension is proportional to

the (potentially very high) number of transmit antennas (see Section IV-C).

A systematic way to exploit the geometry of the problem is via the method of *online mirror descent* (OMD) [22]. To illustrate this method (which can be traced back to the seminal work of Nemirovski and Yudin [23] for offline problems), it is convenient to rewrite the Euclidean update (8) of OGD as

$$\begin{aligned} x_{t+1} = \Pi(x_t + \gamma v_t) &= \arg\min_{x\in\mathcal{X}} \ \tfrac{1}{2}\|x_t + \gamma v_t - x\|^2 \\ &= \arg\min_{x\in\mathcal{X}} \ \{\gamma v_t^\top (x_t - x) + D(x, x_t)\}, \end{aligned} \tag{10}$$

where we have defined

$$D(p,x) = \tfrac{1}{2}\|p-x\|^2 = \tfrac{1}{2}\|p\|^2 - \tfrac{1}{2}\|x\|^2 - x^\top(p-x). \tag{11}$$

The key novelty of mirror descent is to replace this quadratic term by the so-called *Bregman divergence*: $D_h(p,x) = h(p) - h(x) - \nabla h(x)^\top (p-x)$, where $h\colon \mathcal{X} \to \mathbb{R}$ is a smooth K-strongly convex function (usually referred to as a *regularizer*). In so doing, we obtain the *online mirror descent* (OMD) algorithm

$$x_{t+1} = P_{x_t}(\gamma_t v_t) \tag{12}$$

where the *mirror-prox operator* P is defined as

$$P_x(v) = \arg\min_{x'\in\mathcal{X}} \left\{ v^\top (x - x') + D_h(x', x) \right\}, \tag{13}$$

As before, $v_t = -\nabla \ell_t(x_t)$ denotes the negative gradient of the loss function of the t-th sampled at x_t.

Example 1 (Euclidean regularization). Of course, the regularizer $h(x) = \frac{1}{2}\|x\|^2$ yields the archetypal OGD algorithm (8).

Example 2 (Entropic regularization). Another important instance of the OMD method is when the problem's feasible region $\mathcal{X}$ is the unit simplex of $\mathbb{R}^d$ and the regularizer is the (negative) Gibbs–Shannon entropy $h(x) = \sum_{j=1}^d x_j \log x_j$. This yields the EW algorithm (6) for MABs with full information (of the gradient v_t). Remarkably, despite their very different origins, exponential weights and gradient descent are simply different sides of **mirror descent.**

Moreover, the similarity of the feasible allocation vectors with the probability simplex ($d \equiv S$) can be efficiently exploited to propose a tailored entropic regularizer yielding the *online exponential learning* (OXL) algorithm in [20] and enjoying $R_T = \mathcal{O}(\sqrt{\log S \cdot T})$.

Theorem 2 (Worst-case regret of OMD [15, 24, 25])**.** *Against L-Lipschitz convex losses, the OMD algorithm based on a C-strongly convex regularizer h enjoys the regret bound*

$$R_T \le 2L\sqrt{\frac{\max h - \min h}{2K}\,T}, \tag{14}$$

achieved by the step-size $\gamma = L^{-1}\sqrt{2C(\max h - \min h)/T}$.

The main take-away from this main result is that OMD enjoys the same $\mathcal{O}(\sqrt{T})$ rate as OGD, but the multiplicative constants are optimized relative to the dimension and geometry of the problem. This logarithmic reduction is of immense value to real-world Big Data problems that suffer from the curse of dimensionality. As a result, the systematic design of tailor-made OMD algorithms for arbitrary problem geometries has attracted considerable interest in the literature and remains a vigorously researched question.

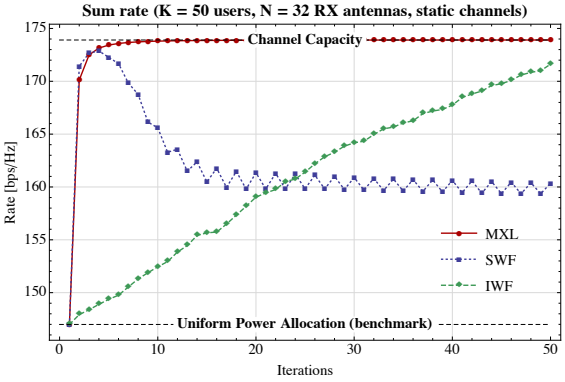

Fig. 3: Comparison between our MXL and iterative and simultaneous water-filling (IWF and SWF) in a static uplink setting with multiple rate-driven users [9]: IWF converges slowly because only one user updates its input covariance per iteration; SWF is faster but does not always converge due to cycles in the update process. On the contrary, MXL converges within a few iterations.

C. Extensions to multi-user MIMO systems

Consider a dynamic MIMO wireless network [9, 26, 27] composed of multiple autonomous devices equipped with multiple antennas. Each user seeks to maximize the Shannon rate or the incurred loss $\ell_t(\mathbf{Q}_t) = -\log\det(\mathbf{I} + \tilde{\mathbf{H}}\mathbf{Q}_t\tilde{\mathbf{H}}^\dagger)$ against any possible sequence of dynamically varying effective channel matrices $\tilde{\mathbf{H}}_t$. Energy efficiency maximization [1] and cognitive medium access [4, 28] are other important applications in MIMO systems that can be formulated as online optimization problems.

In such settings, the control variable is the transmit signal covariance matrix $\mathbf{Q} \equiv \mathbf{X}$, and the problem's feasible region can often be casted into a spectrahedron of the form $\boldsymbol{\mathcal{X}} = \{\mathbf{X} \in \mathbb{R}^{d\times d} : \mathbf{X} \succcurlyeq 0, \operatorname{tr}\mathbf{X} \le 1\}$, where d is the number of transmit antennas. A tailor-made regularizer for this type of constraints is given by the (negative) *von Neumann entropy*: $h(\mathbf{X}) = \operatorname{tr}[\mathbf{X}\log\mathbf{X}] + (1 - \operatorname{tr}\mathbf{X})\log(1 - \operatorname{tr}\mathbf{X})$. As was shown in [9, 27], this choice yields the *matrix exponential learning* (MXL) algorithm

$$\begin{aligned} \mathbf{Y}_{t+1} &= \mathbf{Y}_t + \gamma \mathbf{V}_t \\ \mathbf{X}_{t+1} &= \frac{\exp(\mathbf{Y}_{t+1})}{1 + \operatorname{tr}[\exp(\mathbf{Y}_{t+1})]} \end{aligned} \tag{15}$$

where $\mathbf{V}_t = -\nabla_{\mathbf{X}_t}\ell_t(\mathbf{X}_t)$ is the matrix gradient of the t-th round loss function.[3] A very appealing property of the exponential update in (15) is that it ensures positive-definiteness in an elegant and lightweight manner compared to the Euclidean projection on the positive-definite cone (which requires solving a convex optimization problem at each stage).

For an illustration of MXL performance, see Fig. 3, which highlights the interest of online optimization tools, especially

[3] Since ℓ_t is a real function of a Hermitian matrix, its gradient is also a Hermitian matrix, so $\exp(\mathbf{Y}_t)$ is positive-definite.

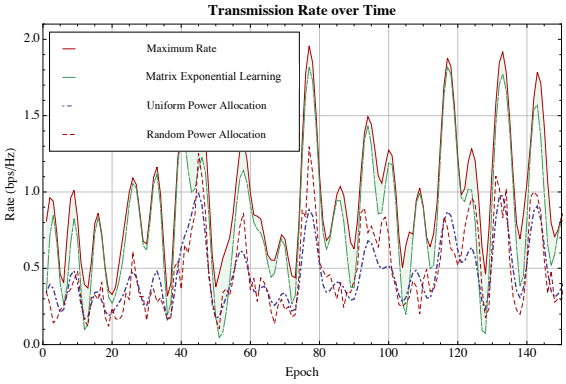

Fig. 4: Rate achieved by MXL in a dynamic MIMO system with mobile users (average velocity 15 km/h) [4]: the adaptive online policy induced by MXL allows users to track their instantaneous optimal rates remarkably well.

for their theoretical guarantees and convergence properties[4]. Indeed, providing convergence guarantees for water-filling algorithms [9, 27] may be very challenging. For instance, the sufficient conditions in [8] roughly require all interfering links to be dominated by the direct ones, which is quite restrictive and even impossible in multiple access networks. In dynamic MIMO systems where water-filling algorithms are brought to a halt (in the absence of non-causal feedback), MXL performs remarkably close to the optimal instantaneous policy in (1) as illustrated in Fig. 4.

V. Reducing the feedback information

So far, the underlying assumption has been that a perfect gradient feedback $v_t = -\nabla \ell_t(x_t)$ is made available to the optimizer at each stage. In this section, we focus on reducing the quality and the amount of feedback information. In the above MIMO systems, the gradient is a (potentially large) matrix and its feedback to the transmitter at each stage can lead to prohibitive overhead [29]; hence, reducing the feedback is crucial in such applications.

Imperfect gradient feedback: Let's start by considering imperfections in the optimizer's feedback where only a noisy estimate: $\hat{v}_t$ of the true gradient is available at each stage. Remarkably, under quite standard statistical assumptions: $\mathbb{E}[\hat{v}_t \mid \mathcal{F}_{t-1}] = v_t$ *unbiased estimator*, and $\mathbb{E}[\|\hat{v}_t\|^2 \mid \mathcal{F}_{t-1}] \leq V^2$ *bounded mean square* ($\mathcal{F}_t$ denotes the history of play up to stage t), the first-order algorithms are able to retain similar performance as in the perfect gradient case. More precisely, the mean regret upper bounds remain the same by simply replacing the maximum gradient norm L with the second moment factor V [15, 25].

A. Zeroth-order feedback

The noisy feedback case can be seen as the precursor to the more challenging question: **can the optimizer attain a no-regret state without gradient feedback?** This question is sometimes referred to as "*bandit online optimization*" (in reference to MABs in which only the reward of the chosen arm is known as opposed to the full information case) or "*online optimization with zeroth-order feedback*" since the only available feedback is the actual incurred loss.

One-point stochastic gradient approximation: In adversarial MABs, this is achieved by means of the *importance sampling* technique [19, 24]. In online convex optimization problems, the key idea is to exploit the scalar value of a function to build an estimator of its gradient; this is precisely achieved by Spall in [30] by sampling the function not at the point of interest, but at a nearby, randomly chosen point.

To illustrate the idea, consider the one-dimensional case, in which we seek to estimate the derivative of a function $f\colon \mathbb{R} \to \mathbb{R}$ at some target point $\hat{x}$ using a single evaluation thereof. By definition, the derivative can be approximated by $f'(\hat{x}) \approx \frac{f(\hat{x}+\delta)-f(\hat{x}-\delta)}{2\delta}$, for sufficiently small δ. Of course, this estimate requires two function evaluations, but it also suggests the following approach: simply make a (uniform) random draw of $z \in \{\pm 1\}$ and sample f at $\hat{x} + \delta z$, yielding the one-point estimator: $\hat{v} = \delta^{-1} f(\hat{x} + \delta z)\, z$ such that

$$\frac{f(\hat{x}+\delta) - f(\hat{x}-\delta)}{2\delta} = \frac{1}{\delta}\, \mathbb{E}[f(\hat{x}+\delta z)\, z]. \tag{16}$$

This can be extended to multi-dimensional problems by taking z to be a uniformly random vector drawn from the unit d-dimensional sphere [19, 20].

Although first-order algorithms (OGD, OMD) exploiting this estimator are able to retain the no-regret property, two major remarks are in order. First, if the pivot point $\hat{x}_t$ lies too close to the boundary of $\mathcal{X}$, the chosen action $x_t = \hat{x}_t + \delta z_t$ may lie outside the feasible set, which is extremely problematic in practice (e.g., in the power allocation problem in Section IV). To account for this feasibility issue, one can keep the method's pivots away from the boundary of $\mathcal{X}$ by re-projecting them onto a "δ-shrunk" sub-region of $\mathcal{X}$, as in [20, 29]. Second, the one-point estimator has an extremely poor *bias vs. variance* trade-off: the bias scales as $\mathcal{O}(\delta)$, while the variance scales as $\mathcal{O}(1/\delta)$ and, hence, δ has to be optimally tuned.

The above leads to the non-trivial compromise when reducing the feedback to one scalar in a much poorer regret decay rate: $\bar{R}_T = \mathcal{O}(\mathrm{poly}(d)\ T^{3/4})$, compared to the optimal $\mathcal{O}(\sqrt{T})$, which also scales polynomially with the problem dimension. This is precisely illustrated in Fig. 5, in which the OXL algorithm (see also Section IV) with gradient feedback is compared to its zeroth-order counterpart [20] for $d \equiv S = 4$. When the problem dimension increases, this performance drop becomes problematic even in *static* settings [29].

Improved stochastic gradient approximation: Motivated by the fact that EW with importance sampling is capable to retrieve the optimal regret rate $\mathcal{O}(\sqrt{T})$ in adversarial MABs [24], a new and improved stochastic gradient approximation was introduced in [29] for *static* convex optimization problems in multi-user MIMO systems.

By exploiting the current sample of the loss function jointly with that of the previous stage (the so-called *callback mechanism*), a two-point stochastic gradient estimator can be built such that $\hat{v}_t = \delta^{-1}[f(\hat{x}_t + \delta z_t) - f(\hat{x}_{t-1} + \delta z_{t-1})]\, z_t$, which has the

[4]Note that the water-filling methods require the exact same amount of feedback as our MXL.

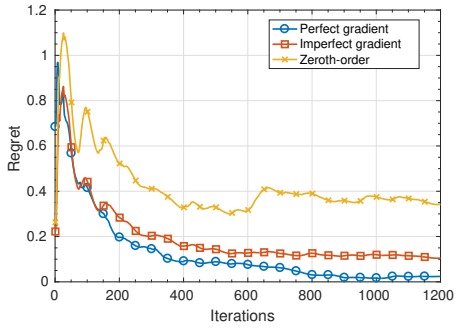

Fig. 5: Impact of reducing the amount of feedback for K users and $S = 4$ subcarriers [20]. Although zeroth-order OXL retains no regret, its average regret exhibits a much slower decay compared with gradient-based OXL.

same bias but a *bounded variance* compared to the one-point estimator. The resulting zeroth-order MXL algorithm is shown to convergence at an optimal rate $\mathcal{O}(\text{poly}(d)\sqrt{T})$. Although this methodology can be easily transposed to other *static* convex problems, in the general *online* convex optimization framework, this remains an open and not trivial issue.

B. One bit of feedback

Let's push to the extreme the reduction of the feedback and ask: **what can be achieved with a single bit of feedback?** Of course, building a relevant one-bit gradient estimator to exploit first-order algorithms does not seem realistic. Instead, we simplify the problem formulation by *quantizing* the feasible set and then exploiting MABs in Section III at the cost of an optimality loss.

One-bit feedback MAB-based adaptive policies: The beam-alignment problem in mmWave MIMO networks has been investigated in [31] via MABs with one bit of feedback. At each stage, the bit of information is of ACK/NACK-type and conveys whether a certain quality of service has been met at the receiver (e.g, in terms of minimum rate), as a result of the chosen arm. The discrete arms represent the possible beam-directions from a predefined and optimized codebook. The unknown expectations of the arm rewards coincide with the opposite of their outage probabilities.

The performance of the resulting online policies (EW or UCB-based) is quite surprising. As recently shown in [32], one-bit feedback MABs remain competitive even with *deep learning* neural networks that have been trained offline. And this, in spite of: their quantization loss; their lack of any *a priori* knowledge, but learning *on-the-fly*; their single bit of strictly causal information; and their lower computational load (compared only to the running phase of the neural networks, excluding the training).

In [33], exploiting similar one-bit feedback MABs showed that non-orthogonal multiple access (NOMA) can be performed efficiently and outperform its OMA counterpart without any prior channel state nor distribution information, as commonly assumed.

VI. Applications beyond wireless

Spurred by the enthusiasm surrounding the Big Data paradigm, online optimization has found vast applications in

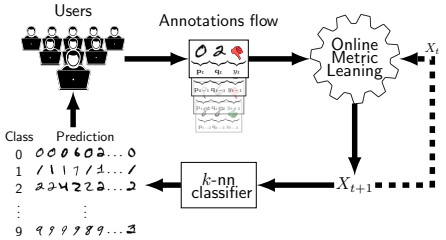

Fig. 6: Online metric learning for MNIST digit classification using the k-nearest neighbors classifier.

problems where the trade-off between data *exploration* and *exploitation* is crucial. In signal processing, examples include sparse coding and dictionary learning [34], data classification and filtering [35], matrix completion and prediction [36], Poisson inverse problems in tomography [37], etc. As a concrete application, we consider here supervised metric learning for image similarity search and classification [10, 38]. The aim is to learn a positive-definite matrix $\mathbf{X}$ shaping the *Mahalanobis distance*: $d_X(\mathbf{p},\mathbf{q}) = (\mathbf{p}-\mathbf{q})^\top \mathbf{X}(\mathbf{p}-\mathbf{q})$ that best captures the similarity of two images $\mathbf{p}$ and $\mathbf{q}$, based upon existing annotations or examples (supervised learning).

An example in the dataset is a triplet $(\mathbf{p},\mathbf{q},y)$ where $\mathbf{p}$, $\mathbf{q}$ are two images and y represents their similarity score (e.g., $y = +1$ if the images are similar and $y = -1$ otherwise). Intuitively, the Malahanobis distance performs a linear transformation of the data and computes the distance $d_X(\mathbf{p},\mathbf{q}) = \|\mathbf{X}^{1/2}\mathbf{p} - \mathbf{X}^{1/2}\mathbf{q}\|^2$ in the transformed space. The idea is to learn the best transformation that brings closer the similar images and separates the dissimilar ones.

This application enables us to highlight the range of the MXL algorithm, which can be exploited here directly. Specifically, we have compared the Euclidean metric with the online metric that was learned by the MXL algorithm, and a different online metric based on the mirror descent for metric learning (MDML) algorithm of [5], which exploits the Frobenius norm regularization. Our MXL algorithm performs best in terms of classification test errors on three well-known datasets: Iris, Wine and MNIST. Both online metrics always outperform the Euclidean one and MXL provides the best online learned metric. In Fig. 7, we plot the principal components in the Euclidean and the transformed space via MXL in the Wine dataset, in which the Euclidean representation is quite ill-suited for classification. For the complete experimental setup and results, we refer the reader to the online report [39].

VII. Conclusions

This paper provides an introduction to *online convex optimization*, to its online policies along with their neat theoretical guarantees, and to its links with other classic frameworks. Application-wise, future wireless networks provide a particularly suitable playground; the derived online policies are: distributed and reinforcing, come with theoretical guarantees, and, most remarkably, are able to adapt *on-the-fly* to arbitrarily and unpredictable changes in the wireless environment, relying on strictly causal and *limited feedback*.

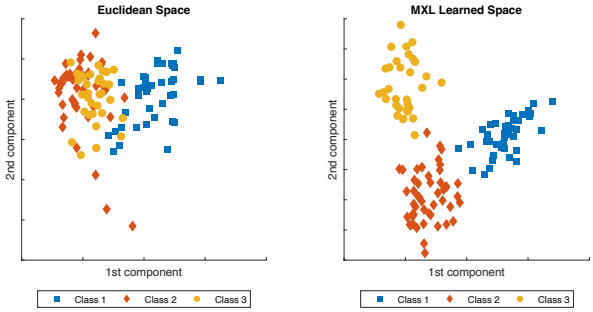

Fig. 7: Principal components in the Euclidean and the transformed space via MXL in the Wine dataset. The MXL algorithm provides the best linear transformation, which successfully separates the images into the three classes.

References

[1] P. Mertikopoulos and E. V. Belmega, "Learning to be green: Robust energy efficiency maximization in dynamic MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 743 – 757, April 2016.

[2] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 731–745, April 2011.

[3] T. Chen, S. Barbarossa, X. Wang, G. B. Giannakis, and Z.-L. Zhang, "Learning and management for internet of things: Accounting for adaptivity and scalability," *Proc. of the IEEE*, vol. 107, no. 4, 2019.

[4] P. Mertikopoulos and E. V. Belmega, "Transmit without regrets: online optimization in MIMO–OFDM cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 1987–1999, November 2014.

[5] G. Kunapuli and J. Shavlik, "Mirror descent for metric learning: A unified approach," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 859–874.

[6] J. Hannan, "Approximation to Bayes risk in repeated play," in *Contributions to the Theory of Games, Volume III*, ser. Annals of Mathematics Studies, M. Dresher, A. W. Tucker, and P. Wolfe, Eds. Princeton, NJ: Princeton University Press, 1957, vol. 39, pp. 97–139.

[7] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical Programming*, vol. 120, no. 1, pp. 221–259, 2009.

[8] G. Scutari, D. P. Palomar, and S. Barbarossa, "The MIMO iterative waterfilling algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1917–1935, May 2009.

[9] P. Mertikopoulos and A. L. Moustakas, "Learning in an uncertain world: MIMO covariance matrix optimization with imperfect feedback," *IEEE Trans. Signal Process.*, vol. 64, no. 1, pp. 5–18, January 2016.

[10] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," *arXiv:1306.6709*, 2013.

[11] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, December 1933.

[12] H. Robbins, "Some aspects of the sequential design of experiments," *Bull. Amer. Math. Soc.*, vol. 58, no. 5, pp. 527–535, 1952.

[13] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, pp. 4–22, 1985.

[14] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, 2002.

[15] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[16] T. M. Cover, "Behavior of sequential predictors of binary sequences," in *Trans. of the 4th Prague Conf. on Inf. Theory, Statistical Decision Functions, and Random Processes*, 1965, pp. 263–272.

[17] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[18] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *Journal of the ACM*, vol. 44, no. 3, pp. 427–485, 1997.

[19] E. V. Belmega, P. Mertikopoulos, R. Negrel, and L. Sanguinetti, "Online convex optimization and no-regret learning: Algorithms, guarantees and applications," *arXiv:1804.04529*, 2018.

[20] A. Marcastel, E. V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Online power optimization in feedback-limited, dynamic and unpredictable IoT networks," *IEEE Trans. Signal Process.*, vol. 67, no. 11, pp. 2987–3000, 2019.

[21] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *ICML '03: Proceedings of the 20th International Conference on Machine Learning*, 2003, pp. 928–936.

[22] S. Shalev-Shwartz, "Online learning: Theory, algorithms, and applications," Ph.D. dissertation, Hebrew University of Jerusalem, 2007.

[23] A. S. Nemirovski and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. New York, NY: Wiley, 1983.

[24] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[25] J. Kwon and P. Mertikopoulos, "A continuous-time approach to online optimization," *Journal of Dynamics and Games*, vol. 4, no. 2, pp. 125–148, April 2017.

[26] G. Scutari, D. P. Palomar, F. Facchinei, and J.-S. Pang, "Convex optimization, game theory, and variational inequality theory in multiuser communication systems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 35–49, May 2010.

[27] P. Mertikopoulos, E. V. Belmega, R. Negrel, and L. Sanguinetti, "Distributed stochastic optimization via matrix exponential learning," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2277–2290, May 2017.

[28] G. Scutari and D. P. Palomar, "MIMO cognitive radio: A game theoretical approach," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 761–780, Feb. 2010.

[29] O. Bilenne, P. Mertikopoulos, and E. V. Belmega, "Fast optimization with zeroth-order feedback in distributed, multi-user MIMO systems," *IEEE Trans. Signal Process.*, vol. 68, pp. 6085–6100, 2020.

[30] J. C. Spall, "A one-measurement form of simultaneous perturbation stochastic approximation," *Automatica*, vol. 33, no. 1, pp. 109–112, 1997.

[31] I. Chafaa, E. V. Belmega, and M. Debbah, "One-bit feedback exponential learning for beam alignment in mobile mmWave," *IEEE Access*, vol. 8, pp. 194 575–194 589, 2020.

[32] I. Chafaa, R. Negrel, E. V. Belmega, and M. Debbah, "Self-supervised deep learning for mmWave beam steering exploiting Sub-6 GHz channels," *IEEE Trans. Wireless Commun.*, 2022.

[33] H. El Hassani, A. Savard, and E. V. Belmega, "Adaptive NOMA in time-varying wireless networks with no CSIT/CDIT relying on a 1-bit feedback," *IEEE Wireless Commun. Lett.*, vol. 10, no. 4, pp. 750–754, 2020.

[34] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, March 2010.

[35] D. Garber and E. Hazan, "Adaptive universal linear filtering," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1595–1604, April 2013.

[36] O. Shamir and S. Shalev-Shwartz, "Matrix completion with the trace norm: Learning, bounding, and transducing," *Journal of Machine Learning Research*, vol. 15, pp. 3401–3423, October 2014.

[37] K. Antonakopoulos, E. V. Belmega, and P. Mertikopoulos, "Online and stochastic optimization beyond lipschitz continuity: A riemannian approach," in *Intl. Conf. on Learning Representations (ICLR)*, 2019.

[38] R. Negrel, D. Picard, and P.-H. Gosselin, "Web-scale image retrieval using compact tensor aggregation of visual descriptors," *IEEE Magazine on MultiMedia*, vol. 20, no. 3, pp. 24–33, 2013.

[39] E. V. Belmega, P. Mertikopoulos, R. Negrel, and L. Sanguinetti, "Matrix exponential learning in multimedia classification problems: Experimental setup and results," *Tech. Rep.*, Mar. 2018. [Online]. Available: https://perso.esiee.fr/~negrelr/conf/MLX2018_ML_setup.pdf