Online Learning and Optimization: Algorithms and Applications

fexxee
计算机科学与技术学院
xxx@xxx.xxx.cn

摘要

随着网络大数据时代的到来，日益增长的数据量对于在线学习 *(Online Learning)* 算法提出了更高的要求。受限于数据存储以及设备计算性能，传统的离线优化与学习算法难以直接被使用，更多新的算法在探索 *(Exploration)*与利用 *(Exploitation)* 之间做出平衡。本文将对在线优化和在线学习相关算法进行介绍，包括在线优化算法的发展、衡量指标、不同优化算法的理论性能分析以及应用举例。最后，我们将列举一些在线优化的前沿进展并预测该领域未来的发展方向。

1. 背景介绍

在线优化 (Online Optimization) 的基本形式如下：在一个重复的决策过程 (Decision Process) 的每一个阶段，优化算法需要依据之前阶段累计的经验以及当前输入 x_t 做出预测 p_t，并得到该预测与当前时刻标准输出 y_t 之间的损失代价，用于描述算法预测的准确性；在线优化算法将持续进行决策并得到反馈，在过程中对预测模型进行修改，目的是随着决策轮次的增长提高预测性能。在这样的问题条件下，在线优化理论寻求该类问题算法的理论性能，并且给出相应的算法实现。

随着数据科学及统计学习相关算法的发展，在线优化算法已经被广泛应用于在线学习及顺序决策问题中，包括模型控制 [31]，云计算基础设施服务 (LAAS) [21]，通信领域的多天线终端信道优化 [24]，信号处理中的数据过滤 [8] 以及矩阵分解和稀疏编码 [22] 等领域。

在传统优化问题 (Static Optimization) 中，核心的假设是优化目标 (比如，求取函数 $f(x)$ 的最小值) 已知并且不随时间推移而改变。随机优化 (Stochastic Optimization) 在此基础上放宽了对目标函数的要求，不再要求其固定形式，而是可能依赖于一个平稳随机过程 (Stationary Stochastic Process) 的假设。更进一步地，在线优化引入了多智能体 (multi-agent) 的视角，采用对抗博弈的

模式对优化过程进行分析。

在线优化模型具有以下优势。第一，在线学习模型允许从多输入源分布式地获取输入数据，而传统优化模型的性能依赖于中心化的大规模数据集。第二，基于 Hannan 提出的最小化 'Regret' 概念 [10] —— 即一个描述算法性能与理论固定最优策略之间差距的损失量，以在线优化理论为基础推导出的相关算法对问题的不确定性有理论性能保证。在线优化的目标是在已知尽可能少的模型假设的情况下，推导出尽可能最接近理论优化解的在线学习算法。第三，在线优化有很强的实用性，近 20 年来，在线学习在多个大数据相关领域的问题中被广泛使用且取得了很好的效果。

我们将在第 2 章节中介绍在线优化的基本内容；在第 3 章节中，我们将介绍多种在线优化算法类别以及他们的理论性能，包括在线梯度下降 (OGD) [35]，在线镜像下降 (OMD) [30, 29]；最后，在第 4 章，我们将介绍在线优化在线学习的前沿算法并列举一些最新进展。本文的大体目标概括如下：

- 对在线优化问题进行概述并给出在线优化算法与其他优化算法之间的联系。
- 介绍在线优化算法并对其理论性能进行分析。
- 列举在线优化的前沿课题与最新进展。

2. 在线优化介绍

在本章节，我们将介绍在线优化相关的基本内容。具体地，我们将首先介绍凸优化 (Convex Optimization) 的相关内容；其次，我们将介绍在线学习的概念以及具体形式；接着，我们给出在线学习的具体实例；最后，我们将通过在线学习的核心概念, Regret 以及基于此概念的在线学习性能分析。

2.1. 凸问题相关概念

本文关于在线优化问题的讨论大多在凸问题条件下，传统的优化算法也多针对凸问题，这里首先介绍凸问题分析优化的相关内容。

凸集: 一个 n 维集合 $S \subset \mathbb{R}^n$ 被称为凸集 (Convex Set) 当且仅当 S 满足对于 $\forall\, x,\, y \in S$, $0 \leq \alpha \leq 1$, 式 1 恒成立：

$$\alpha \cdot x + (1-\alpha) \cdot y := z \in S \tag{1}$$

凸函数: 函数 $f : \mathbb{R}^n \to \mathbb{R}$ 被称为凸函数 (Convex Function) 当且仅当对于 $\forall\, x, y \in S, 0 \leq \alpha \leq 1$, 式 2 恒成立：

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) \tag{2}$$

在一般定义的基础上，若 f 一阶、二阶连续可微，则式 3、4、5 也是 f 为凸函数的等价条件。

$$\langle f'(x) - f'(y), x - y \rangle \geq 0 \tag{3}$$

$$f(x) + \langle f'(x), x - y \rangle \leq f(y) \tag{4}$$

$$f''(x) \succeq 0 \tag{5}$$

其中，$f''(x)$ 为函数 f 在 x 处的的Hessian矩阵。特别地，如果上式中的不等号均为严格大于 (或小于)，则称为严格凸 (Strictly Convex) 函数。在此基础上，为了研究梯度方法、二阶方法在拥有特定凸性问题上的性能，对强凸性 (Strong Convexity) 定义如下: 若函数 $f: \mathbb{R}^n \to \mathbb{R}$ 满足 $\exists\, \mu > 0, f(x) - \frac{1}{2}\mu||x||^2$ 为凸函数，或者等价地，式 6 成立，则称 f 为强凸函数

$$\alpha f(x) + (1-\alpha) f(y) \geq f(\alpha x + (1-\alpha) y) + \alpha(1-\alpha)\frac{\mu}{2}||x-y||^2 \tag{6}$$

类似于凸性，对于高阶可微函数，有式 7、8 作为强凸性质的等价定义：

$$\langle f'(x) - f'(y), x - y \rangle \geq \mu||x-y||^2 \tag{7}$$

$$f''(x) \succeq \mu I_n \tag{8}$$

其中，I_n 为n维单位矩阵。讨论函数的强凸性有利于分析不同优化方法的理论收敛性能。本文接下来将着重介绍在线优化的相关内容，在此只对凸性做简单的说明，对凸问题的传统优化方法，可以参阅 Hindi 的说明资料 [17]。

2.2. 在线优化

接下来，我们将介绍在线优化的核心流程：在每个优化阶段 $t = 1, 2, \ldots$，负责优化的智能体 (Agent) 从备选决策凸集 $K \in \mathbb{R}^n$ 中选择一个行为 (Action) x_t 作为当前阶段的决策，接着，系统反馈一个针对该决策的损失 $\ell_t(x_t)$，其中 $\ell_t: K \to \mathbb{R}$ 是损失函数（可能对智能体已知，也可能由随机、对抗产生）。接下来在 $t+1$ 阶段，智能体依据已有的知识更新其策略，选择一个新的行动 x_{t+1} 并持续至结束，整个流程可以被表示为：

Algorithm 1 在线优化算法流程

必要条件： 备选决策集 $K \in \mathbb{R}^n$，每阶段的损失函数 $\ell_t: K \to \mathbb{R}$

1: **for** t = 1 **to** T **do**
2: 根据已有信息，选择当前阶段的行为 $x_t \in K$
3: 计算并反馈当前损失 $\ell_t(x_t)$
4: **end for**

上文中，我们并没有对损失函数 ℓ_t 形式做出任何假设。实际上，在线学习主要处理以下的凸问题类：

- 在线凸优化 (Online Convex Optimization) : $\forall\, t, \ell_t : K \to \mathbb{R}$ 是凸函数。
- 在线强凸优化 (Online Strongly Convex Optimization) : $\forall\, t, \ell_t : K \to \mathbb{R}$ 是强凸函数。
- 在线线性优化 (Online Linear Optimization) : $\forall\, t, \ell_t(x_t) = -v_t^T x_t$。

其中线性问题和强凸问题均为凸问题的子集，由于其分别具有独特的性质，产生了不同的针对性优化方法。对于其他类别问题的在线学习也有相关算法 [5]，但本文主要介绍上述三类目前学界主要研究的问题。

2.3. 应用 —— 在线度量学习

接下来，我们将介绍一个具体的在线学习的例子 —— 图像相似性检测与图片分类中的度量学习问题 [32, 1, 33, 20]。度量学习的主要目的是在图像或文本聚类过程中自动学习出针对某个特定任务的度量距离的函数，在该函数的作用下，原始数据被映射到一个新的空间，使得在该空间下样本密度分布更加合理。具体地，假设 $p,\ q$ 是两个图像或词向量样本，度量学习 (线性) 的目的是学习一个正定矩阵 X ,用来尽量准确地刻画 $p,\ q$ 在经 X 变换后空间中的马氏距离 $d_X(p,q) = (p-q)^T X(p-q) = ||X^{1/2}p - X^{1/2}q||^2$，我们期望标签相似的样本在变换后的空间中距离更近，反之距离更远。

在许多现实场景中，比如近年来常见的分布式网络线上标注任务，后台研究人员无法一次性得到所有的数据样本的对应标注而进行离线训练，数据和标注在过程中不断更新。这种场景下，每一轮接收用户对样本的新标注作为反馈，计算损失函数来惩罚已有的度量矩阵，从而对模型参数进行调整以达到更好的度量效果。

2.4. 在线学习中的 Regret

接下来，我们介绍与在线学习算法理论性能分析密切相关的概念 ——Regret (遗憾) 。Regret 用于描述在线优化的累计损失与传统批量优化 (Batch-based Optimization) 得到的最优解对应的累计损失之间的差距，式 9 是 Regret 的数学定义：

$$R_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_{x^* \in K} \sum_{t=1}^{T} \ell_t(x^*) \tag{9}$$

其中，前半部分为在线优化的累计损失，后半部分为最优离线解产生的累计损失。在赌博机问

题 [3] 中，智能体在 t 阶段的最优选择为:

$$x_t \in \underset{x\in K}{arg\,min} \sum_{\tau=1}^{t-1} \ell_\tau(x) \tag{10}$$

可以看到，该在线问题最优解是已基于所有已有数据进行离线优化得到的最优解。

当然，如果损失函数对于优化智能体不可见，那么上述最优解也无法被计算。因此，在线优化的核心优化目标是设计一个随时间增长而不会产生 Regret 的算法，如式 11 所示：

$$\lim_{T\to\infty} \frac{R_T}{T} = 0 \iff R_T = o(T) \tag{11}$$

显然，如果上式能够满足，那么当 T 足够大时，在线优化算法的性能将足够接近最优的离线优化算法性能。为此，在优化过程中，算法集中于最小化 Regret。No Regret 的思想为在线学习的理论性能提供了非常好的保障，并促成了一系列快速优化算法 [26] 产生。具体来讲，假设损失函数不随时间改变且为凸函数： $\ell_t \equiv f$ ，则根据琴生不等式以及式 11 ，有以下结论：

$$f(\frac{1}{T}\sum_{t=1}^{T} x_t) - \underset{x^*\in K}{min}\, f(x^*) \le \frac{1}{T}\sum_{t=1}^{T}[f(x_t) - \underset{x^*\in K}{min}\, f(x^*)] \le \frac{R_T}{T} \overset{T\to\infty}{\longrightarrow} 0 \tag{12}$$

公式 12 为传统凸优化问题提供了顺序处理的优化方案，我们可以把在线优化算法得到的顺序解的平均值 $\bar{x_T} = \frac{1}{T}\sum_{t=1}^{T} x_t$ 作为最终凸优化问题的解，从而极大地减少离线优化中的运算，同时理论性能得到充分保证 ($\bar{x_T}$ 与 x^* 具有相同的准确度) 。接下来，我们将介绍多种在线优化算法以及他们的理论上界分析。

3. 在线优化算法与理论性能

接下来，我们要解决上一章中提出的核心问题 —— 设计怎样的在线优化算法达到 No Regret 的目标，以及分析这些算法的收敛率。本章节列举经典常用的在线优化算法，更多详细的算法介绍可以参考 [30, 29, 13, 4, 27, 11]。

3.1. 在线梯度下降 —— Online Gradient Descent

在传统离线优化问题中，最常用的算法都基于梯度下降：在某一个优化阶段，优化算法向当前位置梯度的反方向移动一段距离；如果可行域 K 受限，再将梯度下降后的结果进行投影回可行空间。在在线学习领域，Zinkevich提出的在线梯度下降 (Online Gradient Descent, OGD) [35] 同样是最基本的算法。

OGD的基本算法如下：在第 t+1 阶段，利用 t 阶段的损失函数 ℓ_t 进行梯度下降，如下面两

个公式所示：

$$y_{t+1} = x_t - \gamma f'(x_t) \tag{13}$$

$$x_{t+1} = \Pi(y_{t+1}) = \underset{x \in K}{arg\ min}\, ||y_{t+1} - x||^2 \tag{14}$$

其中 $\gamma > 0$ 为步长， y_{t+1} 为直接梯度下降之后得到的结果， Π 为投影函数。算法流程如下：

Algorithm 2 Online Gradient Descent (OGD)

必要条件: 步长 $\gamma > 0$, 损失函数 ℓ_t

1: 初始化 $x_0 \in K$
2: **for** t = 0 **to** $T-1$ **do**
3: 　计算梯度下降点 $y_{t+1} = x_t - \gamma \ell_t'(x_t)$
4: 　计算投影点 $x_{t+1} = arg\ min_{x \in K} ||y_{t+1} - x||^2$
5: 　计算当前损失 $\ell_t(x_{t+1})$ 及其导数 $\ell_t'(x_{t+1})$
6: **end for**

接下来，我们分析 OGD 的收敛性能。根据投影点的对应关系，我们可以得到：

$$||x_{t+1} - x^*||^2 \leq ||y_{t+1} - x^*||^2 = ||x_t - x^*||^2 - 2\gamma \langle\, \ell_t'(x_t),\ (x^* - x_t)\,\rangle + \gamma^2 ||\ell_t'(x_t)||^2 \tag{15}$$

由此式变形后，可以得到：

$$\langle\, \ell_t'(x_t),\ (x^* - x_t)\,\rangle \leq \frac{1}{2\gamma}(||x_t - x^*||^2 - ||x_{t+1} - x^*||^2) + \frac{\gamma}{2}||\ell_t'(x_t)||^2 \tag{16}$$

另一方面，根据损失函数的凸性可以得到：

$$R_T = \sum_{t=1}^{T} [\ell_t(x_t) - \ell_t(x^*)] \leq \sum_{t=1}^{T} \langle\, \ell_t'(x_t),\ (x^* - x_t)\,\rangle \tag{17}$$

合并上述两式可得：

$$R_T \leq \frac{1}{2\gamma}||x_1 - x^*||^2 + \frac{\gamma}{2}\sum_{t=1}^{T} ||\ell_t'(x_t)||^2 \tag{18}$$

再由损失函数的Lipschiz连续性 (假设常数为 L) 可知， $\forall\, t, ||\ell_t'(x_t)||^2 \leq L$，因此得到 R_T 上界的最终形式如下：

$$R_T \leq \frac{1}{2\gamma}||x_1 - x^*||^2 + \frac{\gamma L^2 T}{2} \tag{19}$$

观察不等式 19 的形式，当且仅当 $\frac{1}{2\gamma}||x_1 - x^*||^2 = \frac{\gamma L^2 T}{2}$ 时，即 $\gamma = \frac{||x_1 - x^*||}{L\sqrt{T}}$ 时，式右边取最小值，即 $R_T \leq ||x_1 - x^*|| L\sqrt{T}$，因此，我们推出 OGD 方法的性能最差情况是 $\mathcal{O}(\sqrt{T})$ 。之后， Hazan

等人 [12] 提出，如果损失函数 ℓ_t 满足 β-强凸性，则略微改变算法 2 的步长 (从定步长 $\gamma_t = \gamma$ 改为 $\gamma_t = \frac{\beta}{t}$) 可以使算法的 Regret 达到对数级别 $\mathcal{O}(\log(T))$ 。

3.2. 在线镜像下降 —— Online Mirror Descent

尽管 OGD 的 Regret 上界已经足够紧凑，但在多臂赌博机 (MABs) 问题中， OGD 算法的最差情况 $\mathcal{O}(\sqrt{AT})$ 与 Exponential Weights (EW) 算法的上界 $\mathcal{O}(\sqrt{T\log(A)})$ 仍有一定差距，从几何的视角来看，二者系数的差距主要来自不同空间距离 (l^∞ $v.s.$ l^2) 。在离线优化算法中，有一类被称为镜像下降 (Mirror Descent) [25] 的算法，用来描述更具有普遍性的空间距离 —— 以 Bregman divergence 来代替传统的欧氏距离，受此启发，Shalev 等人 [30] 提出了在线镜像下降 OMD 方法。在 OGD 中，我们按以下方式定义距离：

$$D(x,y) = ||y-x||^2 = ||y||^2 - ||x||^2 - 2x^T(y-x) \tag{20}$$

作为对比， OMD 的创新点在于用 Bregman divergence 替换欧式距离：

$$D_h(x,y) = h(y) - h(x) - h'(x)^T(p-x) \tag{21}$$

其中， $h: K \to \mathbb{R}$ 是一个 $\beta-$强凸函数。 OMD 算法流程如下：

Algorithm 3 Online Mirror Descent (OMD)

必要条件: 步长 $\gamma > 0$, 损失函数 ℓ_t , $\beta-$强凸函数 h

1: 初始化 $x_0 \in K$
2: **for** t = 0 **to** $T-1$ **do**
3: 计算当前选取点 $x_{t+1} = arg\, min_{x' \in K} \{\gamma \ell_t'(x_t)^T(x_t - x') + D_h(x', x)\}$
4: 计算当前损失 $\ell_t(x_{t+1})$ 及其导数 $\ell_t'(x_{t+1})$
5: **end for**

这里，我们给出 OMD 算法的最坏情况，与 OGD 中类似，我们假设损失函数满足 $L-Lipschitz$ 连续，距离函数 h 满足 $\beta-$强凸，OMD算法的 Regret 上界为：

$$R_T \leq 2L\sqrt{\frac{\max\ h - \min h}{2K}T} \tag{22}$$

注意到 OMD 与 OGD 算法的上界主要成分都是 $\mathcal{O}(\sqrt{T})$ ，但二者的常系数不同，且依赖于问题的几何维度。对于多臂赌博机问题 MABs ， OMD 算法可以达到和 EW 算法相同的理论性能。目前，对于高维度问题，如何针对性设计更有效的 OMD 算法仍是研究热点。除了 OGD 和 OMD，常用的在线学习算法还有基于二阶微分信息的在线牛顿法 [28]。

4. 在线学习前沿进展

介绍完基本的在线优化算法之后，在本章节，我们将介绍一些在线优化领域的研究热点以及近年来的新进展。主要包括 FTL 系列算法，无需投影的在线优化算法，以及未来的研究方向。

4.1. Follow The Leader, FTL

接下来，本文将介绍一系列特殊的算法。这些算法始于一个直觉算法。我们回顾 Regret 的定义：

$$R_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_{x^* \in K} \sum_{t=1}^{T} \ell_t(x^*) \tag{23}$$

很自然地，我们想到可以使用已有的数据进行离线优化，来作为当前轮次我们给出的解，即在第 t 阶段：

$$x_t = \arg\min_{x \in K} \sum_{\tau=1}^{t-1} \ell_\tau(x) \tag{24}$$

这种基于直觉的方法由 Kalai 等人 [19] 提出并命名为 Be-The-leader 算法。该算法最大的问题是具有潜在的不稳定性，不能保证 $\ell_t(x_t)$ 与 $\ell_t(x_{t+1})$ 之间的的距离。

为了解决不稳定性问题， Follow-The-Regularized-Leader (FTRL) 方法被提出，具体来讲，首先用一个线性函数替代原损失函数：

$$\ell_t(x) \rightarrow \ell_t'(x_t)^T x \tag{25}$$

其次，在优化目标中加入正则项，一个 $\beta-$强凸函数 $R(x)$：

$$x_t = \arg\min_{x \in K} \sum_{\tau=1}^{t-1} \ell_\tau'(x_\tau)^T x + \frac{1}{\beta} R(x) \tag{26}$$

这样的优化目标可以保证 $\ell_t'(x_t)^T \cdot x_t - \ell_t'(x_t)^T \cdot x_{t+1} \approx \mathcal{O}(\beta)$，从而解决了原 FTL 算法的不稳定问题。在此基础上， Kalai 等人 [19] 还进一步分析了在线线性优化 (OLO) 中的随机正则化方法，提出了 Follow-The-Perturbed-Leader 方法，分析得到对于 x_t 属于非凸集的情况下，仍有 $R_T = \mathcal{O}(\sqrt{T})$ 。之后，Duchi 等人 [6] 和 McMahan 等人 [23] 进一步研究了适应性的正则化方法，并结合次梯度下降方法提出了 Adagrad 算法。

4.2. Projection-free algorithms

我们在之前章节中讲的主流在线优化算法 OGD, OMD, FTRL 均包含投影 (Projection) 步骤，事实上，这一步骤需要昂贵的计算成本 ($\mathcal{O}(n^3)$) ，因此，有越来越多的工作 [15, 18, 16, 9, 34] 把

重点放在无需投影的算法研究中。基于 Frank 等人的工作 [7]，Hazan 等人注意到线性优化的简单性，尝试把投影过程转变为线性优化, 提出了在线条件梯度优化 [15]，其上界 $R_T = \mathcal{O}(T^{3/4})$ 。之后，Garber 提出线性收敛的条件梯度方法 [9]，优化上界至 $\mathcal{O}(\sqrt{T})$ 。近两年，还出现了关注分布式在线优化中免投影方法的研究 [34] 。

4.3. 未来研究方向

综合近年来在线优化的研究工作 [14, 2]，未来该领域的研究方向主要可能集中于以下几个方面：

- 对维度更敏感的高效在线凸优化算法。
- 更优 Regret 和更快算法之间的平衡问题。
- 对更一般问题集的免投影在线优化算法
- 理论上下限分析及最优解的寻找
- 实际问题场景中凸条件逐渐放宽以及对 OCO 算法的调整

5. 总结

本文对在线优化问题、相关算法、前沿知识做了总结与介绍，分析了在线优化与传统离线优化相比的特点、优势；对已有在线优化方法做了理论性能分析与推导；并列举了一些前沿算法。由于在线优化天然地适合大数据时代的分布式数据处理流程，未来在线优化必将成为机器学习与人工智能领域的重要研究课题，希望本文能够给读者简单清晰的介绍，为更深入地了解在线优化提供帮助。

参考文献

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

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

[3] D. Bergemann and J. Valimaki. Bandit problems. 2006.

[4] S. Bubeck, N. Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[5] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[6] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[7] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[8] D. Garber and E. Hazan. Adaptive universal linear filtering. *IEEE Transactions on Signal Processing*, 61(7):1595–1604, 2013.

[9] D. Garber and E. Hazan. Playing non-linear games with linear oracles. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 420–428. IEEE, 2013.

[10] J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

[11] E. Hazan. 10 the convex optimization approach to regret minimization. *Optimization for machine learning*, page 287, 2012.

[12] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[13] E. Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[14] E. Hazan and S. Kale. Online convex optimization: A game-theoretic approach to learning. http://www.cs.princeton.edu/~ehazan/tutorial/tutorial.htm. Accessed June, 2016.

[15] E. Hazan and S. Kale. Projection-free online learning. *arXiv preprint arXiv:1206.4657*, 2012.

[16] E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

[17] H. Hindi. A tutorial on convex optimization. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 3252–3265. IEEE, 2004.

[18] M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

[19] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[20] J. Lee, S. Abu-El-Haija, B. Varadarajan, and A. P. Natsev. Collaborative deep metric learning for video understanding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 481–490. ACM, 2018.

[21] J. Li, M. Qiu, Z. Ming, G. Quan, X. Qin, and Z. Gu. Online optimization for scheduling preemptable tasks on iaas cloud systems. *Journal of Parallel and Distributed Computing*, 72(5):666–677, 2012.

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[23] H. B. McMahan and M. Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010.

[24] P. Mertikopoulos and E. V. Belmega. Learning to be green: Robust energy efficiency maximization in dynamic mimo–ofdm systems. *IEEE Journal on Selected Areas in Communications*, 34(4):743–757, 2016.

[25] A. S. Nemirovsky and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983.

[26] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[27] A. Rakhlin and K. Sridharan. Statistical learning and sequential prediction. *Book Draft*, 2014.

[28] N. N. Schraudolph, J. Yu, and S. Günter. A stochastic quasi-newton method for online convex optimization. In *Artificial intelligence and statistics*, pages 436–443, 2007.

[29] S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[30] S. Shalev-Shwartz and Y. Singer. Online learning: Theory, algorithms, and applications. 2007.

[31] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IFAC Proceedings Volumes*, 41(2):6974–6979, 2008.

[32] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003.

[33] J. Yu, X. Yang, F. Gao, and D. Tao. Deep multimodal distance metric learning using click constraints for image ranking. *IEEE transactions on cybernetics*, 47(12):4014–4024, 2017.

[34] W. Zhang, P. Zhao, W. Zhu, S. C. Hoi, and T. Zhang. Projection-free distributed online learning in networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4054–4062. JMLR. org, 2017.

[35] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.