Introduction to Online Convex Optimization

Elad Hazan
Princeton University
ehazan@cs.princeton.edu

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Optimization

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

E. Hazan . *Introduction to Online Convex Optimization*. Foundations and Trends® in Optimization, vol. 2, no. 3-4, pp. 157–325, 2015.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-68083-171-9

The full text is available at: http://dx.doi.org/10.1561/2400000013

Foundations and Trends® in Optimization

Volume 2, Issue 3-4, 2015

Editorial Board

The full text is available at: http://dx.doi.org/10.1561/2400000013

Editorial Scope

Topics

Foundations and Trends® in Optimization publishes survey and tutorial articles on methods for and applications of mathematical optimization, including the following topics:

- Algorithm design, analysis, and implementation (especially on modern computing platforms)
- Models and modeling systems
- New optimization formulations for practical problems
- Applications of optimization in:
  - Machine learning
  - Statistics
  - Data analysis
  - Signal and image processing
  - Computational economics and finance
  - Engineering design
  - Scheduling and resource allocation
  - and other areas

Information for Librarians

Foundations and Trends® in Optimization, 2015, Volume 2, 4 issues. ISSN paper version 2167-3888. ISSN online version 2167-3918. Also available as a combined paper and online subscription.

Foundations and Trends® in Optimization
Vol. 2, No. 3-4 (2015) 157–325
© 2016 E. Hazan
DOI: 10.1561/2400000013

Introduction to Online Convex Optimization

Elad Hazan
Princeton University
ehazan@cs.princeton.edu

Contents

Abstract

This manuscript portrays *optimization as a process.* In many practical applications the environment is so complex that it is infeasible to lay out a comprehensive theoretical model and use classical algorithmic theory and mathematical optimization. It is necessary as well as beneficial to take a robust approach, by applying an optimization method that learns as one goes along, learning from experience as more aspects of the problem are observed. This view of optimization as a process has become prominent in varied fields and has led to some spectacular success in modeling and systems that are now part of our daily lives.

E. Hazan . *Introduction to Online Convex Optimization.* Foundations and Trends® in Optimization, vol. 2, no. 3-4, pp. 157–325, 2015.
DOI: 10.1561/2400000013.

The full text is available at: http://dx.doi.org/10.1561/2400000013

1

Introduction

This manuscript concerns the view of *optimization as a process*. In many practical applications the environment is so complex that it is infeasible to lay out a comprehensive theoretical model and use classical algorithmic theory and mathematical optimization. It is necessary as well as beneficial to take a robust approach, by applying an optimization method that learns as one goes along, learning from experience as more aspects of the problem are observed. This view of optimization as a process has become prominent in various fields and led to spectacular successes in modeling and systems that are now part of our daily lives.

The growing literature of machine learning, statistics, decision science and mathematical optimization blur the classical distinctions between deterministic modeling, stochastic modeling and optimization methodology. We continue this trend in this book, studying a prominent optimization framework whose precise location in the mathematical sciences is unclear: the framework of *online convex optimization*, which was first defined in the machine learning literature (see bibliography at the end of this chapter). The metric of success is borrowed from game theory, and the framework is closely tied to statistical learning theory and convex optimization.

We embrace these fruitful connections and, on purpose, do not try to fit any particular jargon. Rather, this book will start with actual problems that can be modeled and solved via online convex optimization. We will proceed to present rigorous definitions, background, and algorithms. Throughout, we provide connections to the literature in other fields. It is our hope that you, the reader, will contribute to our understanding of these connections from your domain of expertise, and expand the growing literature on this fascinating subject.

1.1 The online convex optimization model

In online convex optimization, an online player iteratively makes decisions. At the time of each decision, the outcomes associated with the choices are unknown to the player.

After committing to a decision, the decision maker suffers a loss: every possible decision incurs a (possibly different) loss. These losses are unknown to the decision maker beforehand. The losses can be adversarially chosen, and even depend on the action taken by the decision maker.

Already at this point, several restrictions are necessary for this framework to make any sense at all:

- The losses determined by an adversary should not be allowed to be unbounded. Otherwise the adversary could keep decreasing the scale of the loss at each step, and never allow the algorithm to recover from the loss of the first step. Thus we assume the losses lie in some bounded region.

- The decision set must be somehow bounded and/or structured, though not necessarily finite.

  To see why this is necessary, consider decision making with an infinite set of possible decisions. An adversary can assign high loss to all the strategies chosen by the player indefinitely, while setting apart some strategies with zero loss. This precludes any meaningful performance metric.

Surprisingly, interesting statements and algorithms can be derived with not much more than these two restrictions. The Online Convex Optimization (OCO) framework models the decision set as a convex set in Euclidean space denoted $\mathcal{K} \subseteq \mathbb{R}^n$. The costs are modeled as bounded convex functions over $\mathcal{K}$.

The OCO framework can be seen as a structured repeated game. The protocol of this learning framework is as follows:

At iteration t, the online player chooses $\mathbf{x}_t \in \mathcal{K}$. After the player has committed to this choice, a convex cost function $f_t \in \mathcal{F} : \mathcal{K} \mapsto \mathbb{R}$ is revealed. Here $\mathcal{F}$ is the bounded family of cost functions available to the adversary. The cost incurred by the online player is $f_t(\mathbf{x}_t)$, the value of the cost function for the choice $\mathbf{x}_t$. Let T denote the total number of game iterations.

What would make an algorithm a good OCO algorithm? As the framework is game-theoretic and adversarial in nature, the appropriate performance metric also comes from game theory: define the **regret** of the decision maker to be the difference between the total cost she has incurred and that of the best fixed decision in hindsight. In OCO we are usually interested in an upper bound on the worst case regret of an algorithm.

Let $\mathcal{A}$ be an algorithm for OCO, which maps a certain game history to a decision in the decision set. We formally define the regret of $\mathcal{A}$ after T iterations as:

$$\text{regret}_T(\mathcal{A}) = \sup_{\{f_1,\ldots,f_T\}\subseteq\mathcal{F}} \left\{ \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}) \right\} \tag{1.1}$$

Intuitively, an algorithm performs well if its regret is sublinear as a function of T, i.e. $\text{regret}_T(\mathcal{A}) = o(T)$, since this implies that on the average the algorithm performs as well as the best fixed strategy in hindsight.

The running time of an algorithm for OCO is defined to be the worst-case expected time to produce $\mathbf{x}_t$, for an iteration $t \in [T]$[1] in a T-iteration repeated game. Typically, the running time will depend on n (the dimensionality of the decision set $\mathcal{K}$), T (the total number of game

[1] Here and henceforth we denote by $[n]$ the set of integers $\{1, ..., n\}$.

The full text is available at: http://dx.doi.org/10.1561/2400000013

iterations), and the parameters of the cost functions and underlying convex set.

1.2 Examples of problems that can be modeled via OCO

Perhaps the main reason that OCO has become a leading online learning framework in recent years is its powerful modeling capability: problems from diverse domains such as online routing, ad selection for search engines and spam filtering can all be modeled as special cases. In this section, we briefly survey a few special cases and how they fit into the OCO framework.

Prediction from expert advice

Perhaps the most well known problem in prediction theory is the so-called "experts problem". The decision maker has to choose among the advice of n given experts. After making her choice, a loss between zero and one is incurred. This scenario is repeated iteratively, and at each iteration the costs of the various experts are arbitrary (possibly even adversarial, trying to mislead the decision maker). The goal of the decision maker is to do as well as the best expert in hindsight.

The online convex optimization problem captures this problem as a special case: the set of decisions is the set of all distributions over n elements (experts), i.e., the n-dimensional simplex $\mathcal{K} = \Delta_n = \{\mathbf{x} \in \mathbb{R}^n \ , \ \sum_i \mathbf{x}_i = 1 \ , \ \mathbf{x}_i \geq 0\}$. Let the cost of the i'th expert at iteration t be $\mathbf{g}_t(i)$, and let $\mathbf{g}_t$ be the cost vector of all n experts. Then the cost function is the expected cost of choosing an expert according to distribution $\mathbf{x}$, and is given by the linear function $f_t(\mathbf{x}) = \mathbf{g}_t^\top \mathbf{x}$.

Thus, prediction from expert advice is a special case of OCO in which the decision set is the simplex and the cost functions are linear and bounded, in the ℓ_∞ norm, to be at most one. The bound on the cost functions is derived from the bound on the elements of the cost vector $\mathbf{g}_t$.

The fundamental importance of the experts problem in machine learning warrants special attention, and we shall return to it and analyze it in detail at the end of this chapter.

The full text is available at: http://dx.doi.org/10.1561/2400000013

Online spam filtering

Consider an online spam-filtering system. Repeatedly, emails arrive into the system and are classified as spam/valid. Obviously such a system has to cope with adversarially generated data and dynamically change with the varying input—a hallmark of the OCO model.

The linear variant of this model is captured by representing the emails as vectors according to the “bag-of-words” representation. Each email is represented as a vector $\mathbf{x} \in \mathbb{R}^d$, where d is the number of words in the dictionary. The entries of this vector are all zero, except for those coordinates that correspond to words appearing in the email, which are assigned the value one.

To predict whether an email is spam, we learn a filter, for example a vector $\mathbf{x} \in \mathbb{R}^d$. Usually a bound on the Euclidean norm of this vector is decided upon a priori, and is a parameter of great importance in practice.

Classification of an email $\mathbf{a} \in \mathbb{R}^d$ by a filter $\mathbf{x} \in \mathbb{R}^d$ is given by the sign of the inner product between these two vectors, i.e., $\hat{y} = \text{sign}\langle \mathbf{x}, \mathbf{a} \rangle$ (with, for example, $+1$ meaning valid and -1 meaning spam).

In the OCO model of online spam filtering, the decision set is taken to be the set of all such norm-bounded linear filters, i.e., the Euclidean ball of a certain radius. The cost functions are determined according to a stream of incoming emails arriving into the system, and their labels (which may be known by the system, partially known, or not known at all). Let $(\mathbf{a}, y)$ be an email/label pair. Then the corresponding cost function over filters is given by $f(\mathbf{x}) = \ell(\hat{y}, y)$. Here $\hat{y}$ is the classification given by the filter $\mathbf{x}$, y is the true label, and ℓ is a convex loss function, for example, the square loss $\ell(\hat{y}, y) = (\hat{y} - y)^2$.

Online shortest paths

In the online shortest path problem, the decision maker is given a directed graph $G = (V, E)$ and a source-sink pair $u, v \in V$. At each iteration $t \in [T]$, the decision maker chooses a path $p_t \in \mathcal{P}_{u,v}$, where $\mathcal{P}_{u,v} \subseteq E^{|V|}$ is the set of all u-v-paths in the graph. The adversary independently chooses weights (lengths) on the edges of the graph, given by a function from the edges to the real numbers $\mathbf{w}_t : E \mapsto \mathbb{R}$, which can be represented as a vector $\mathbf{w}_t \in \mathbb{R}^m$, where $m = |E|$. The decision maker suffers and observes a loss, which is the weighted length of the chosen path $\sum_{e \in p_t} \mathbf{w}_t(e)$.

The discrete description of this problem as an experts problem, where we have an expert for each path, presents an efficiency challenge. There are potentially exponentially many paths in terms of the graph representation size.

Alternatively, the online shortest path problem can be cast in the online convex optimization framework as follows. Recall the standard description of the set of all distributions over paths (flows) in a graph as a convex set in $\mathbb{R}^m$, with $O(m + |V|)$ constraints (Figure 1.1). Denote this flow polytope by $\mathcal{K}$. The expected cost of a given flow $\mathbf{x} \in \mathcal{K}$ (distribution over paths) is then a linear function, given by $f_t(\mathbf{x}) = \mathbf{w}_t^\top \mathbf{x}$, where, as defined above, $\mathbf{w}_t(e)$ is the length of the edge $e \in E$. This inherently succinct formulation leads to computationally efficient algorithms.

$$\begin{array}{lr}
\displaystyle\sum_{e=(u,w),w \in V} \mathbf{x}_e = 1 = \sum_{e=(w,v),w \in V} \mathbf{x}_e & \text{flow value is one} \\
\displaystyle\forall w \in V \setminus \{u, v\} \quad \sum_{e=(v,x) \in E} \mathbf{x}_e = \sum_{e=(x,v) \in E} \mathbf{x}_e & \text{flow conservation} \\
\forall e \in E \quad 0 \leq \mathbf{x}_e \leq 1 & \text{capacity constraints}
\end{array}$$

Figure 1.1: Linear equalities and inequalities that define the flow polytope, which is the convex hull of all u-v paths.

Portfolio selection

In this section we consider a portfolio selection model that does not make any statistical assumptions about the stock market (as opposed to the standard geometric Brownian motion model for stock prices), and is called the "universal portfolio selection" model.

At each iteration $t \in [T]$, the decision maker chooses a distribution of her wealth over n assets $\mathbf{x}_t \in \Delta_n$. The adversary independently chooses market returns for the assets, i.e., a vector $\mathbf{r}_t \in \mathbb{R}^n$ with strictly positive entries such that each coordinate $\mathbf{r}_t(i)$ is the price ratio for the i'th asset between the iterations t and $t+1$. The ratio between the wealth of the investor at iterations $t+1$ and t is $\mathbf{r}_t^\top \mathbf{x}_t$, and hence the gain in this setting is defined to be the logarithm of this change ratio in wealth $\log(\mathbf{r}_t^\top \mathbf{x}_t)$. Notice that since $\mathbf{x}_t$ is the distribution of the investor's wealth, even if $\mathbf{x}_{t+1} = \mathbf{x}_t$, the investor may still need to trade to adjust for price changes.

The goal of regret minimization, which in this case corresponds to minimizing the difference $\max_{\mathbf{x}^\star \in \Delta_n} \sum_{t=1}^T \log(\mathbf{r}_t^\top \mathbf{x}^\star) - \sum_{t=1}^T \log(\mathbf{r}_t^\top \mathbf{x}_t)$, has an intuitive interpretation. The first term is the logarithm of the wealth accumulated by the best possible in-hindsight distribution $\mathbf{x}^\star$. Since this distribution is fixed, it corresponds to a strategy of rebalancing the position after every trading period, and hence, is called a *constant rebalanced portfolio*. The second term is the logarithm of the wealth accumulated by the online decision maker. Hence regret minimization corresponds to maximizing the ratio of the investor's wealth to the wealth of the best benchmark from a pool of investing strategies.

A *universal* portfolio selection algorithm is defined to be one that, in this setting, attains regret converging to zero. Such an algorithm, albeit requiring exponential time, was first described by Cover (see bibliographic notes at the end of this chapter). The online convex optimization framework has given rise to much more efficient algorithms based on Newton's method. We shall return to study these in detail in Chapter 4.

Matrix completion and recommendation systems

The prevalence of large-scale media delivery systems such as the Netflix online video library, Spotify music service and many others, give rise to very large scale recommendation systems. One of the most popular and successful models for automated recommendation is the matrix completion model.

In this mathematical model, recommendations are thought of as composing a matrix. The customers are represented by the rows, the different media are the columns, and at the entry corresponding to a particular user/media pair we have a value scoring the preference of the user for that particular media.

For example, for the case of binary recommendations for music, we have a matrix $X \in \{0,1\}^{n \times m}$ where n is the number of persons considered, m is the number of songs in our library, and 0/1 signifies dislike/like respectively:

$$X_{ij} = \begin{cases} 0, & \text{person } i \text{ dislikes song } j \\ \\ 1, & \text{person } i \text{ likes song } j \end{cases} .$$

In the online setting, for each iteration the decision maker outputs a preference matrix $X_t \in \mathcal{K}$, where $\mathcal{K} \subseteq \{0,1\}^{n \times m}$ is a subset of all possible zero/one matrices. An adversary then chooses a user/song pair (i_t, j_t) along with a "real" preference for this pair $y_t \in \{0,1\}$. Thus, the loss experienced by the decision maker can be described by the convex loss function,

$$f_t(X) = (X_{i_t,j_t} - y_t)^2.$$

The natural comparator in this scenario is a low-rank matrix, which corresponds to the intuitive assumption that preference is determined by few unknown factors. Regret with respect to this comparator means performing, on the average, as few preference-prediction errors as the best low-rank matrix.

We return to this problem and explore efficient algorithms for it in Chapter 7.

1.3 A gentle start: learning from expert advice

Consider the following fundamental iterative decision making problem:

At each time step $t = 1, 2, \ldots, T$, the decision maker faces a choice between two actions A or B (i.e., buy or sell a certain stock). The decision maker has assistance in the form of N "experts" that offer their advice. After a choice between the two actions has been made, the decision maker receives feedback in the form of a loss associated with each decision. For simplicity one of the actions receives a loss of zero (i.e., the "correct" decision) and the other a loss of one.

We make the following elementary observations:

1. A decision maker that chooses an action uniformly at random each iteration, trivially attains a loss of $\frac{T}{2}$ and is "correct" 50% of the time.

2. In terms of the number of mistakes, no algorithm can do better in the worst case! In a later exercise, we will devise a randomized setting in which the expected number of mistakes of any algorithm is at least $\frac{T}{2}$.

We are thus motivated to consider a *relative performance metric*: can the decision maker make as few mistakes as the best expert in hindsight? The next theorem shows that the answer in the worst case is negative for a deterministic decision maker.

Theorem 1.1. Let $L \leq \frac{T}{2}$ denote the number of mistakes made by the best expert in hindsight. Then there does not exist a deterministic algorithm that can guarantee less than $2L$ mistakes.

Proof. Assume that there are only two experts and one always chooses option A while the other always chooses option B. Consider the setting in which an adversary always chooses the opposite of our prediction (she can do so, since our algorithm is deterministic). Then, the total number of mistakes the algorithm makes is T. However, the best expert makes no more than $\frac{T}{2}$ mistakes (at every iteration exactly one of the two experts is mistaken). Therefore, there is no algorithm that can always guarantee less than $2L$ mistakes.

□

This observation motivates the design of random decision making algorithms, and indeed, the OCO framework gracefully models decisions on a continuous probability space. Henceforth we prove Lemmas 1.3 and 1.4 that show the following:

Theorem 1.2. Let $\varepsilon \in (0, \frac{1}{2})$. Suppose the best expert makes L mistakes. Then:

1. There is an efficient deterministic algorithm that can guarantee less than $2(1+\varepsilon)L + \frac{2 \log N}{\varepsilon}$ mistakes;

2. There is an efficient randomized algorithm for which the expected number of mistakes is at most $(1+\varepsilon)L + \frac{\log N}{\varepsilon}$.

1.3.1 The weighted majority algorithm

Simple observations: The weighted majority (WM) algorithm is intuitive to describe: each expert i is assigned a weight $W_t(i)$ at every iteration t. Initially, we set $W_1(i) = 1$ for all experts $i \in [N]$. For all $t \in [T]$ let $S_t(A), S_t(B) \subseteq [N]$ be the set of experts that choose A (and respectively B) at time t. Define,

$$W_t(A) = \sum_{i \in S_t(A)} W_t(i) \qquad\qquad W_t(B) = \sum_{i \in S_t(B)} W_t(i)$$

and predict according to

$$a_t = \begin{cases} A & \text{if } W_t(A) \geq W_t(B) \\ B & \text{otherwise.} \end{cases}$$

Next, update the weights $W_t(i)$ as follows:

$$W_{t+1}(i) = \begin{cases} W_t(i) & \text{if expert } i \text{ was correct} \\ W_t(i)(1-\varepsilon) & \text{if expert } i \text{ was wrong} \end{cases},$$

where ε is a parameter of the algorithm that will affect its performance. This concludes the description of the WM algorithm. We proceed to bound the number of mistakes it makes.

The full text is available at: http://dx.doi.org/10.1561/2400000013

Lemma 1.3. Denote by M_t the number of mistakes the algorithm makes until time t, and by $M_t(i)$ the number of mistakes made by expert i until time t. Then, for any expert $i \in [N]$ we have

$$M_T \le 2(1+\varepsilon)M_T(i) + \frac{2\log N}{\varepsilon}.$$

We can optimize ε to minimize the above bound. The expression on the right hand side is of the form $f(x) = ax + b/x$, that reaches its minimum at $x = \sqrt{b/a}$. Therefore the bound is minimized at $\varepsilon^\star = \sqrt{\log N / M_T(i)}$. Using this optimal value of ε, we get that for the best expert $i^\star$

$$M_T \le 2M_T(i^\star) + O\left(\sqrt{M_T(i^\star)\log N}\right).$$

Of course, this value of $\varepsilon^\star$ cannot be used in advance since we do not know which expert is the best one ahead of time (and therefore we do not know the value of $M_T(i^\star)$). However, we shall see later on that the same asymptotic bound can be obtained even without this prior knowledge.

Let us now prove Lemma 1.3.

Proof. Let $\Phi_t = \sum_{i=1}^N W_t(i)$ for all $t \in [T]$, and note that $\Phi_1 = N$.

Notice that $\Phi_{t+1} \le \Phi_t$. However, on iterations in which the WM algorithm erred, we have

$$\Phi_{t+1} \le \Phi_t(1 - \frac{\varepsilon}{2}),$$

the reason being that experts with at least half of total weight were wrong (else WM would not have erred), and therefore

$$\Phi_{t+1} \le \frac{1}{2}\Phi_t(1-\varepsilon) + \frac{1}{2}\Phi_t = \Phi_t(1 - \frac{\varepsilon}{2}).$$

From both observations,

$$\Phi_t \le \Phi_1(1 - \frac{\varepsilon}{2})^{M_t} = N(1 - \frac{\varepsilon}{2})^{M_t}.$$

On the other hand, by definition we have for any expert i that

$$W_T(i) = (1-\varepsilon)^{M_T(i)}.$$

Since the value of $W_T(i)$ is always less than the sum of all weights Φ_T, we conclude that

$$(1-\varepsilon)^{M_T(i)} = W_T(i) \le \Phi_T \le N(1-\frac{\varepsilon}{2})^{M_T}.$$

Taking the logarithm of both sides we get

$$M_T(i)\log(1-\varepsilon) \le \log N + M_T \log\left(1-\frac{\varepsilon}{2}\right).$$

Next, we use the approximations

$$-x - x^2 \le \log(1-x) \le -x \qquad 0 < x < \frac{1}{2},$$

which follow from the Taylor series of the logarithm function, to obtain that

$$-M_T(i)(\varepsilon+\varepsilon^2) \le \log N - M_T\frac{\varepsilon}{2},$$

and the lemma follows. □

1.3.2 Randomized weighted majority

In the randomized version of the WM algorithm, denoted RWM, we choose expert i w.p. $p_t(i) = W_t(i)/\sum_{j=1}^N W_t(j)$ at time t.

Lemma 1.4. Let M_t denote the number of mistakes made by RWM until iteration t. Then, for any expert $i \in [N]$ we have

$$\mathbf{E}[M_T] \le (1+\varepsilon)M_T(i) + \frac{\log N}{\varepsilon}.$$

The proof of this lemma is very similar to the previous one, where the factor of two is saved by the use of randomness:

Proof. As before, let $\Phi_t = \sum_{i=1}^N W_t(i)$ for all $t \in [T]$, and note that $\Phi_1 = N$. Let $\tilde{m}_t = M_t - M_{t-1}$ be the indicator variable that equals one if the RWM algorithm makes a mistake on iteration t. Let $m_t(i)$ equal one if the i'th expert makes a mistake on iteration t and zero otherwise. Inspecting the sum of the weights:

$$\begin{aligned}
\Phi_{t+1} &= \sum_i W_t(i)(1-\varepsilon m_t(i)) \\
&= \Phi_t(1-\varepsilon \sum_i p_t(i) m_t(i)) && p_t(i) = \frac{W_t(i)}{\sum_j W_t(j)} \\
&= \Phi_t(1-\varepsilon\,\mathbf{E}[\tilde{m}_t]) \\
&\le \Phi_t e^{-\varepsilon\,\mathbf{E}[\tilde{m}_t]}. && 1+x \le e^x
\end{aligned}$$

On the other hand, by definition we have for any expert i that

$$W_T(i) = (1-\varepsilon)^{M_T(i)}$$

Since the value of $W_T(i)$ is always less than the sum of all weights Φ_T, we conclude that

$$(1-\varepsilon)^{M_T(i)} = W_T(i) \le \Phi_T \le N e^{-\varepsilon\,\mathbf{E}[M_T]}.$$

Taking the logarithm of both sides we get

$$M_T(i)\log(1-\varepsilon) \le \log N - \varepsilon\,\mathbf{E}[M_T]$$

Next, we use the approximation

$$-x - x^2 \le \log(1-x) \le -x \qquad , \quad 0 < x < \frac{1}{2}$$

to obtain

$$-M_T(i)(\varepsilon + \varepsilon^2) \le \log N - \varepsilon\,\mathbf{E}[M_T],$$

and the lemma follows. □

1.3.3 Hedge

The RWM algorithm is in fact more general: instead of considering a discrete number of mistakes, we can consider measuring the performance of an expert by a non-negative real number $\ell_t(i)$, which we refer to as the *loss* of the expert i at iteration t. The randomized weighted majority algorithm guarantees that a decision maker following its advice will incur an average expected loss approaching that of the best expert in hindsight.

Algorithm 1 Hedge

```
Initialize: $\forall i \in [N],\ W_1(i) = 1$
for $t = 1$ to $T$ do
    Pick $i_t \sim_R W_t$, i.e., $i_t = i$ with probability $\mathbf{x}_t(i) = \frac{W_t(i)}{\sum_j W_t(j)}$
    Incur loss $\ell_t(i_t)$.
    Update weights $W_{t+1}(i) = W_t(i) e^{-\varepsilon \ell_t(i)}$
end for
```

Historically, this was observed by a different and closely related algorithm called Hedge, whose total loss bound will be of interest to us later on in the book.

Henceforth, denote in vector notation the expected loss of the algorithm by

$$\mathbf{E}[\ell_t(i_t)] = \sum_{i=1}^{N} \mathbf{x}_t(i) \ell_t(i) = \mathbf{x}_t^\top \ell_t$$

Theorem 1.5. Let ℓ_t^2 denote the N-dimensional vector of square losses, i.e., $\ell_t^2(i) = \ell_t(i)^2$, let $\varepsilon > 0$, and assume all losses to be non-negative. The Hedge algorithm satisfies for any expert $i^\star \in [N]$:

$$\sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t \le \sum_{t=1}^{T} \ell_t(i^\star) + \varepsilon \sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t^2 + \frac{\log N}{\varepsilon}$$

Proof. As before, let $\Phi_t = \sum_{i=1}^{N} W_t(i)$ for all $t \in [T]$, and note that $\Phi_1 = N$.

Inspecting the sum of weights:

$$\begin{aligned}
\Phi_{t+1} &= \textstyle\sum_i W_t(i) e^{-\varepsilon \ell_t(i)} \\
&= \Phi_t \textstyle\sum_i \mathbf{x}_t(i) e^{-\varepsilon \ell_t(i)} && \mathbf{x}_t(i) = \frac{W_t(i)}{\sum_j W_t(j)} \\
&\le \Phi_t \textstyle\sum_i \mathbf{x}_t(i)(1 - \varepsilon \ell_t(i) + \varepsilon^2 \ell_t(i)^2)) && \text{for } x \ge 0, \\
& && e^{-x} \le 1 - x + x^2 \\
&= \Phi_t(1 - \varepsilon \mathbf{x}_t^\top \ell_t + \varepsilon^2 \mathbf{x}_t^\top \ell_t^2) \\
&\le \Phi_t e^{-\varepsilon \mathbf{x}_t^\top \ell_t + \varepsilon^2 \mathbf{x}_t^\top \ell_t^2}. && 1 + x \le e^x
\end{aligned}$$

On the other hand, by definition, for expert $i^\star$ we have that

$$W_T(i^\star) = e^{-\varepsilon \sum_{t=1}^{T} \ell_t(i^\star)}$$

Since the value of $W_T(i^\star)$ is always less than the sum of all weights Φ_t, we conclude that

$$W_T(i^\star) \leq \Phi_T \leq Ne^{-\varepsilon \sum_t \mathbf{x}_t^\top \ell_t + \varepsilon^2 \sum_t \mathbf{x}_t^\top \ell_t^2}.$$

Taking the logarithm of both sides we get

$$-\varepsilon \sum_{t=1}^{T} \ell_t(i^\star) \leq \log N - \varepsilon \sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t + \varepsilon^2 \sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t^2$$

and the theorem follows by simplifying. □

1.4 Exercises

1. (Attributed to Claude Shannon)
   Construct market returns over two stocks for which the wealth accumulated over any single stock decreases exponentially, whereas the best constant rebalanced portfolio increases wealth exponentially. More precisely, construct two sequences of numbers in the range $(0, \infty)$, that represent returns, such that:

   (a) Investing in any of the individual stocks results in exponential decrease in wealth. This means that the product of the prefix of numbers in each of these sequences decreases exponentially.

   (b) Investing evenly on the two assets and rebalancing after every iteration increases wealth exponentially.

2. (a) Consider the experts problem in which the payoffs are between zero and a positive real number $G > 0$. Give an algorithm that attains expected payoff lower bounded by:

   $$\sum_{t=1}^{T} \mathbf{E}[\ell_t(i_t)] \geq \max_{i^\star \in [N]} \sum_{t=1}^{T} \ell_t(i^\star) - c\sqrt{T \log N}$$

   for the best constant c you can (the constant c should be independent of the number of game iterations T, and the number of experts n. Assume that T is known in advance).

   (b) Suppose the upper bound G is not known in advance. Give an algorithm whose performance is asymptotically as good as your algorithm in part (a), up to an additive and/or multiplicative constant which is independent of T, n, G. Prove your claim.

3. Consider the experts problem in which the payoffs can be negative and are real numbers in the range $[-1, 1]$. Give an algorithm with regret guarantee of $O(\sqrt{T \log n})$ and prove your claim.

1.5 Bibliographic remarks

The OCO model was first defined by Zinkevich (110) and has since become widely influential in the learning community and significantly extended since (see thesis and surveys (52; 53; 97)).

The problem of prediction from expert advice and the Weighted Majority algorithm were devised in (71; 73). This seminal work was one of the first uses of the multiplicative updates method—a ubiquitous meta-algorithm in computation and learning, see the survey (11) for more details. The Hedge algorithm was introduced in (44).

The Universal Portfolios model was put forth in (32), and is one of the first examples of a worst-case online learning model. Cover gave an optimal-regret algorithm for universal portfolio selection that runs in exponential time. A polynomial time algorithm was given in (62), which was further sped up in (7; 54). Numerous extensions to the model also appeared in the literature, including addition of transaction costs (20) and relation to the Geometric Brownian Motion model for stock prices (56).

In their influential paper, Awerbuch and Kleinberg (14) put forth the application of OCO to online routing. A great deal of work has been devoted since then to improve the initial bounds, and generalize it into a complete framework for decision making with limited feedback. This framework is an extension of OCO, called Bandit Convex Optimization (BCO). We defer further bibliographic remarks to chapter 6 which is devoted to the BCO framework.

References

[1] J. Abernethy, R. M. Frongillo, and A. Wibisono. Minimax option pricing meets black-scholes in the limit. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1029–1040, New York, NY, USA, 2012. ACM.

[2] J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 263–274, 2008.

[3] J. Abernethy, C. Lee, A. Sinha, and A. Tewari. Online linear optimization via smoothing. In *Proceedings of The 27th Conference on Learning Theory*, pages 807–823, 2014.

[4] J. Abernethy, C. Lee, and A. Tewari. Perturbation techniques in online learning and optimization. In T. Hazan, G. Papandreou, and D. Tarlow, editors, *Perturbations, Optimization, and Statistics*, Neural Information Processing Series, chapter 8. MIT Press, 2016. to appear.

[5] J. Abernethy and A. Rakhlin. Beating the adaptive bandit with high probability. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

[6] I. Adler. The equivalence of linear programs and zero-sum games. *International Journal of Game Theory*, 42(1):165–177, 2013.

[7] 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*, ICML '06, pages 9–16, New York, NY, USA, 2006. ACM.

[8] D. J. Albers, C. Reid, and G. B. Dantzig. An interview with george b. dantzig: The father of linear programming. *The College Mathematics Journal*, 17(4):pp. 292–314, 1986.

[9] Z. Allen-Zhu and E. Hazan. Optimal black-box reductions between optimization objectives. *CoRR*, abs/1603.05642, 2016.

[10] N. Alon and J. Spencer. *The Probabilistic Method.* John Wiley, 1992.

[11] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

[12] J. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009.

[13] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.

[14] B. Awerbuch and R. Kleinberg. Online linear optimization and adaptive routing. *J. Comput. Syst. Sci.*, 74(1):97–114, 2008.

[15] K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Mach. Learn.*, 43(3):211–246, June 2001.

[16] F. Bach, S. Lacoste-Julien, and G. Obozinski. On the equivalence between herding and conditional gradient algorithms. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 1359–1366, New York, NY, USA, July 2012. Omnipress.

[17] L. Bachelier. Théorie de la spéculation. *Annales Scientifiques de l'École Normale Supérieure*, 3(17):21–86, 1900.

[18] A. Bellet, Y. Liang, A. B. Garakani, M.-F. Balcan, and F. Sha. Distributed frank-wolfe algorithm: A unified framework for communication-efficient sparse learning. *CoRR*, abs/1404.2644, 2014.

[19] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[20] A. Blum and A. Kalai. Universal portfolios with and without transaction costs. *Mach. Learn.*, 35(3):193–205, June 1999.

[21] J. Borwein and A. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples.* CMS Books in Mathematics. Springer, 2006.

[22] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, 1992.

[23] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, March 2004.

[24] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015.

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

[26] E. Candes and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.

[27] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inf. Theor.*, 50(9):2050–2057, September 2006.

[28] N. Cesa-Bianchi and C. Gentile. Improved risk tail bounds for on-line algorithms. *Information Theory, IEEE Transactions on*, 54(1):386–390, Jan 2008.

[29] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

[30] K. L. Clarkson, E. Hazan, and D. P. Woodruff. Sublinear optimization for machine learning. *J. ACM*, 59(5):23:1–23:49, November 2012.

[31] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[32] T. Cover. Universal portfolios. *Math. Finance*, 1(1):1–19, 1991.

[33] V. Dani, T. Hayes, and S. Kakade. The price of bandit information for online optimization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20.* MIT Press, Cambridge, MA, 2008.

[34] G. B. Dantzig. *Maximization of a Linear Function of Variables Subject to Linear Inequalities, in Activity Analysis of Production and Allocation*, chapter XXI. Wiley, New York, 1951.

[35] O. Dekel, A. Tewari, and R. Arora. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.

[36] P. DeMarzo, I. Kremer, and Y. Mansour. Online trading algorithms and robust option pricing. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 477–486, 2006.

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

[38] J. C. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 257–269, 2010.

[39] M. Dudík, Z. Harchaoui, and J. Malick. Lifted coordinate descent for learning with trace-norm regularization. *Journal of Machine Learning Research - Proceedings Track*, 22:327–336, 2012.

[40] E. Even-Dar, S. Kakade, and Y. Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

[41] E. Even-dar, Y. Mansour, and U. Nadav. On the convergence of regret minimization dynamics in concave games. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 523–532, 2009.

[42] A. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394, 2005.

[43] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:149–154, 1956.

[44] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

[45] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1–2):79 – 103, 1999.

[46] D. Garber and E. Hazan. Approximating semidefinite programs in sublinear time. In *NIPS*, pages 1080–1088, 2011.

[47] D. Garber and E. Hazan. Playing non-linear games with linear oracles. In *FOCS*, pages 420–428, 2013.

[48] A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.

[49] J. Hannan. Approximation to bayes risk in repeated play. *In M. Dresher, A. W. Tucker, and P. Wolfe, editors, Contributions to the Theory of Games, volume 3*, pages 97–139, 1957.

[50] Z. Harchaoui, M. Douze, M. Paulin, M. Dudík, and J. Malick. Large-scale image classification with trace-norm regularization. In *CVPR*, pages 3386–3393, 2012.

[51] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[52] E. Hazan. *Efficient Algorithms for Online Convex Optimization and Their Applications*. PhD thesis, Princeton University, Princeton, NJ, USA, 2006. AAI3223851.

[53] E. Hazan. A survey: The convex optimization approach to regret minimization. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*, pages 287–302. MIT Press, 2011.

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

[55] E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *The 21st Annual Conference on Learning Theory (COLT)*, pages 57–68, 2008.

[56] E. Hazan and S. Kale. On stochastic and worst-case models for investing. In *Advances in Neural Information Processing Systems 22*. MIT Press, 2009.

[57] E. Hazan and S. Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. *Journal of Machine Learning Research - Proceedings Track*, pages 421–436, 2011.

[58] E. Hazan and S. Kale. Projection-free online learning. In *ICML*, 2012.

[59] E. Hazan, T. Koren, and N. Srebro. Beating sgd: Learning svms in sublinear time. In *Advances in Neural Information Processing Systems*, pages 1233–1241, 2011.

[60] M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.

[61] M. Jaggi and M. Sulovský. A simple algorithm for nuclear norm regularized problems. In *ICML*, pages 471–478, 2010.

[62] A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *J. Mach. Learn. Res.*, 3:423–440, March 2003.

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

[64] L. Kantorovich. A new method of solving some classes of extremal problems. *Doklady Akad Sci USSR*, 28:211–214, 1940.

[65] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, Cambridge, MA, USA, 1994.

[66] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inf. Comput.*, 132(1):1–63, 1997.

[67] J. Kivinen and M. K. Warmuth. Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45(3):301–329, 2001.

[68] J. Kivinen and M. Warmuth. Averaging expert predictions. In P. Fischer and H. Simon, editors, *Computational Learning Theory*, volume 1572 of *Lecture Notes in Computer Science*, pages 153–167. Springer Berlin Heidelberg, 1999.

[69] S. Lacoste-Julien, M. Jaggi, M. W. Schmidt, and P. Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 53–61, 2013.

[70] J. Lee, B. Recht, R. Salakhutdinov, N. Srebro, and J. A. Tropp. Practical large-scale optimization for max-norm regularization. In *NIPS*, pages 1297–1305, 2010.

[71] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. In *Proceedings of the 30th Annual Symposium on the Foundations of Computer Science*, pages 256–261, 1989.

[72] N. Littlestone. From on-line to batch learning. In *Proceedings of the Second Annual Workshop on Computational Learning Theory*, COLT '89, pages 269–284, 1989.

[73] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[74] S. Mannor and N. Shimkin. The empirical bayes envelope and regret minimization in competitive markov decision processes. *Mathematics of Operations Research*, 28(2):327–345, 2003.

[75] H. B. McMahan and M. J. Streeter. Adaptive bound optimization for online convex optimization. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 244–256, 2010.

[76] A. S. Nemirovski and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization.* John Wiley UK/USA, 1983.

[77] A. Nemirovskii. Interior point polynomial time methods in convex programming, 2004. Lecture Notes.

[78] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer, 2004.

[79] Y. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

[80] G. Neu, A. György, C. Szepesvári, and A. Antos. Online markov decision processes under bandit feedback. *IEEE Trans. Automat. Contr.*, 59(3):676–691, 2014.

[81] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[82] F. Orabona and K. Crammer. New adaptive algorithms for online classification. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems 2010.*, pages 1840–1848, 2010.

[83] M. F. M. Osborne. Brownian motion in the stock market. *Operations Research*, 2:145–173, 1959.

[84] S. A. Plotkin, D. B. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20(2):257–301, 1995.

[85] A. Rakhlin. Lecture notes on online learning. Lecture Notes, 2009.

[86] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.

[87] A. Rakhlin and K. Sridharan. Theory of statistical learning and sequential prediction. Lecture Notes, 2014.

[88] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 713–719, New York, NY, USA, 2005. ACM.

[89] K. Riedel. A sherman-morrison-woodbury identity for rank augmenting matrices with application to centering. *SIAM J. Mat. Anal.*, 12(1):80–95, January 1991.

[90] H. Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 1952.

[91] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 09 1951.

[92] R. Rockafellar. *Convex Analysis*. Convex Analysis. Princeton University Press, 1997.

[93] T. Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM*, 62(5):32:1–32:42, November 2015.

[94] R. Salakhutdinov and N. Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *NIPS*, pages 2056–2064, 2010.

[95] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[96] S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

[97] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

[98] S. Shalev-Shwartz, A. Gonen, and O. Shamir. Large-scale convex minimization with a low-rank constraint. In *ICML*, pages 329–336, 2011.

[99] S. Shalev-Shwartz and Y. Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.

[100] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: primal estimated sub-gradient solver for svm. *Math. Program.*, 127(1):3–30, 2011.

[101] O. Shamir and S. Shalev-Shwartz. Collaborative filtering with the trace norm: Learning, bounding, and transducing. *JMLR - Proceedings Track*, 19:661–678, 2011.

[102] O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML*, 2013.

[103] N. Srebro. *Learning with Matrix Factorizations*. PhD thesis, Massachusetts Institute of Technology, 2004.

[104] A. Tewari, P. D. Ravikumar, and I. S. Dhillon. Greedy algorithms for structurally constrained high dimensional problems. In *NIPS*, pages 882–890, 2011.

[105] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.

[106] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[107] J. Y. Yu, S. Mannor, and N. Shimkin. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

[108] J. Y. Yu and S. Mannor. Arbitrarily modulated markov decision processes. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 2946–2953, 2009.

[109] T. Zhang. Data dependent concentration bounds for sequential prediction algorithms. In *Proceedings of the 18th Annual Conference on Learning Theory*, COLT'05, pages 173–187, 2005.

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