Online Learning and Online Convex Optimization

Online Learning and Online Convex Optimization

Shai Shalev-Shwartz

Benin School of Computer Science and Engineering
The Hebrew University of Jerusalem
Israel
shais@cs.huji.ac.il

the essence of knowledge

Boston – Delft

Foundations and Trends® in Machine Learning

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
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 S. Shalev-Shwartz, Online Learning and Online Convex Optimization, Foundation and Trends® in Machine Learning, vol 4, no 2, pp 107–194, 2011.

ISBN: 978-1-60198-546-0

Foundations and Trends® in Machine Learning
Volume 4 Issue 2, 2011

Editorial Board

Editorial Scope

Foundations and Trends® in Machine Learning will publish survey and tutorial articles in the following topics:

- Adaptive control and signal processing
- Applications and case studies
- Behavioral, cognitive and neural learning
- Bayesian learning
- Classification and prediction
- Clustering
- Data mining
- Dimensionality reduction
- Evaluation
- Game theoretic learning
- Graphical models
- Independent component analysis
- Inductive logic programming
- Kernel methods
- Markov chain Monte Carlo
- Model choice
- Nonparametric methods
- Online learning
- Optimization
- Reinforcement learning
- Relational learning
- Robustness
- Spectral methods
- Statistical learning theory
- Variational inference
- Visualization

Information for Librarians
Foundations and Trends® in Machine Learning, 2011, Volume 4, 4 issues. ISSN paper version 1935-8237. ISSN online version 1935-8245. Also available as a combined paper and online subscription.

Foundations and Trends® in
Machine Learning
Vol. 4, No. 2 (2011) 107–194

DOI: 10.1561/2200000018

now
the essence of knowledge

Online Learning and Online Convex Optimization

Shai Shalev-Shwartz

Benin School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel, shais@cs.huji.ac.il

Abstract

Online learning is a well established learning paradigm which has both theoretical and practical appeals. The goal of online learning is to make a sequence of accurate predictions given knowledge of the correct answer to previous prediction tasks and possibly additional available information. Online learning has been studied in several research fields including game theory, information theory, and machine learning. It also became of great interest to practitioners due the recent emergence of large scale applications such as online advertisement placement and online web ranking. In this survey we provide a modern overview of online learning. Our goal is to give the reader a sense of some of the interesting ideas and in particular to underscore the centrality of convexity in deriving efficient online learning algorithms. We do not mean to be comprehensive but rather to give a high-level, rigorous yet easy to follow, survey.

Contents

1

Introduction

Online learning is the process of answering a sequence of questions given (maybe partial) knowledge of the correct answers to previous questions and possibly additional available information. The study of online learning algorithms is an important domain in machine learning, and one that has interesting theoretical properties and practical applications.

Online learning is performed in a sequence of consecutive rounds, where at round t the learner is given a question, $\mathbf{x}_t$, taken from an instance domain $\mathcal{X}$, and is required to provide an answer to this question, which we denote by p_t. After predicting an answer, the correct answer, y_t, taken from a target domain $\mathcal{Y}$, is revealed and the learner suffers a loss, $l(p_t, y_t)$, which measures the discrepancy between his answer and the correct one. While in many cases p_t is in $\mathcal{Y}$, it is sometimes convenient to allow the learner to pick a prediction from a larger set, which we denote by D.

Online Learning

for $t = 1, 2, \ldots$
  receive question $\mathbf{x}_t \in \mathcal{X}$
  predict $p_t \in D$
  receive true answer $y_t \in \mathcal{Y}$
  suffer loss $l(p_t, y_t)$

The specific case of yes/no answers and predictions, namely $D = \mathcal{Y} = \{0, 1\}$, is called online classification. In this case it is natural to use the 0–1 loss function: $l(p_t, y_t) = |p_t - y_t|$. That is, $l(p_t, y_t)$ indicates if $p_t = y_t$ (the prediction is correct) or $p_t \neq y_t$ (the prediction is wrong).

For example, consider the problem of predicting whether it is going to rain tomorrow. On day t, the question $\mathbf{x}_t$ can be encoded as a vector of meteorological measurements. Based on these measurements, the learner should predict if it's going to rain tomorrow. In the following day, the learner knows the correct answer.

We can also allow the learner to output a prediction in $[0, 1]$, which can be interpreted as the probability of raining tomorrow. This is an example of an application in which $D \neq \mathcal{Y}$. We can still use the loss function $l(p_t, y_t) = |p_t - y_t|$, which can now be interpreted as the probability to err if predicting that it's going to rain with probability p_t.

The learner's ultimate goal is to minimize the cumulative loss suffered along its run, which translates to making few prediction mistakes in the classification case. The learner tries to deduce information from previous rounds so as to improve its predictions on present and future questions. Clearly, learning is hopeless if there is no correlation between past and present rounds. Classic statistical theory of sequential prediction therefore enforces strong assumptions on the statistical properties of the input sequence (e.g., that it is sampled i.i.d. according to some unknown distribution).

In this review we survey methods which make no statistical assumptions regarding the origin of the sequence of examples. The sequence is allowed to be deterministic, stochastic, or even adversarially adaptive to the learner's own behavior (as in the case of spam email filtering). Naturally, an adversary can make the cumulative loss to our online

learning algorithm arbitrarily large. For example, the adversary can ask the same question on each online round, wait for the learner's answer, and provide the opposite answer as the correct answer. To make non-trivial statements we must further restrict the problem. We consider two natural restrictions.

The first restriction is especially suited to the case of online classification. We assume that all the answers are generated by some target mapping, $h^\star : \mathcal{X} \to \mathcal{Y}$. Furthermore, $h^\star$ is taken from a fixed set, called a hypothesis class and denoted by $\mathcal{H}$, which is known to the learner. With this restriction on the sequence, which we call *the realizable case*, the learner should make as few mistakes as possible, assuming that both $h^\star$ and the sequence of questions can be chosen by an adversary. For an online learning algorithm, A, we denote by $M_A(\mathcal{H})$ the maximal number of mistakes A might make on a sequence of examples which is labeled by some $h^\star \in \mathcal{H}$. We emphasize again that both $h^\star$ and the sequence of questions can be chosen by an adversary. A bound on $M_A(\mathcal{H})$ is called a *mistake-bound* and we will study how to design algorithms for which $M_A(\mathcal{H})$ is minimal.

Alternatively, the second restriction of the online learning model we consider is a relaxation of the realizable assumption. We no longer assume that all answers are generated by some $h^\star \in \mathcal{H}$, but we require the learner to be competitive with the best fixed predictor from $\mathcal{H}$. This is captured by the *regret* of the algorithm, which measures how "sorry" the learner is, in retrospect, not to have followed the predictions of some hypothesis $h^\star \in \mathcal{H}$. Formally, the regret of the algorithm relative to $h^\star$ when running on a sequence of T examples is defined as

$$\text{Regret}_T(h^\star) = \sum_{t=1}^{T} l(p_t, y_t) - \sum_{t=1}^{T} l(h^\star(x_t), y_t), \tag{1.1}$$

and the regret of the algorithm relative to a hypothesis class $\mathcal{H}$ is

$$\text{Regret}_T(\mathcal{H}) = \max_{h^\star \in \mathcal{H}} \text{Regret}_T(h^\star). \tag{1.2}$$

We restate the learner's goal as having the lowest possible regret relative to $\mathcal{H}$. We will sometime be satisfied with "low regret" algorithms, by which we mean that $\text{Regret}_T(\mathcal{H})$ grows sub-linearly with

the number of rounds, T, which implies that the difference between the *average* loss of the learner and the average loss of the best hypothesis in $\mathcal{H}$ tends to zero as T goes to infinity.

1.1 Examples

We already mentioned the problem of online classification. To make the discussion more concrete, we list several additional online prediction problems and possible hypothesis classes.

Online Regression In regression problems, $\mathcal{X} = \mathbb{R}^d$ which corresponds to a set of measurements (often called features), and $\mathcal{Y} = D = \mathbb{R}$. For example, consider the problem of estimating the fetal weight based on ultrasound measurements of abdominal circumference and femur length. Here, each $\mathbf{x} \in \mathcal{X} = \mathbb{R}^2$ is a two-dimensional vector corresponds to the measurements of the abdominal circumference and the femur length. Given these measurements the goal is to predict the fetal weight. Common loss functions for regression problems are the squared loss, $\ell(p, y) = (p - y)^2$, and the absolute loss, $\ell(p, y) = |p - y|$. Maybe the simplest hypothesis class for regression is the class of linear predictors, $\mathcal{H} = \{\mathbf{x} \mapsto \sum_{i=1}^{d} w[i]x[i] : \forall i, w[i] \in \mathbb{R}\}$, where $w[i]$ is the ith element of $\mathbf{w}$. The resulting problem is called *online linear regression*.

Prediction with Expert Advice On each online round the learner has to choose from the advice of d given experts. Therefore, $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^d$, where $x_t[i]$ is the advice of the ith expert, and $D = \{1, \ldots, d\}$. Then, the learner receives the true answer, which is a vector $\mathbf{y}_t \in \mathcal{Y} = [0, 1]^d$, where $y_t[i]$ is the cost of following the advice of the ith expert. The loss of the learner is the cost of the chosen expert, $\ell(p_t, \mathbf{y}_t) = y_t[p_t]$. A common hypothesis class for this problem is the set of constant predictors, $\mathcal{H} = \{h_1, \ldots, h_d\}$, where $h_i(\mathbf{x}) = i$ for all $\mathbf{x}$. This implies that the regret of the algorithm is measured relative to the performance of the strategies which always predict according to the same expert.

Online Ranking On round t, the learner receives a query $\mathbf{x}_t \in \mathcal{X}$ and is required to order k elements (e.g., documents) according to

their relevance to the query. That is, D is the set of all permutations of $\{1,\ldots,k\}$. Then, the learner receives the true answer $y_t \in \mathcal{Y} = \{1,\ldots,k\}$, which corresponds to the document which best matches the query. In web applications, this is the document that the user clicked on. The loss, $\ell(p_t, y_t)$, is the position of y_t in the ranked list p_t.

1.2 A Gentle Start

We start with studying online classification problem, in which $\mathcal{Y} = \mathcal{D} = \{0,1\}$, and $\ell(p,y) = |p-y|$ is the 0–1 loss. That is, on each round, the learner receives $\mathbf{x}_t \in \mathcal{X}$ and is required to predict $p_t \in \{0,1\}$. Then, it receives $y_t \in \{0,1\}$ and pays the loss $|p_t - y_t|$. We make the following simplifying assumption:

- *Finite Hypothesis Class:* We assume that $|\mathcal{H}| < \infty$.

Recall that the goal of the learner is to have a low regret relative to the hypotheses set, $\mathcal{H}$, where each function in $\mathcal{H}$ is a mapping from $\mathcal{X}$ to $\{0,1\}$, and the regret is defined as

$$\text{Regret}_T(\mathcal{H}) = \max_{h\in\mathcal{H}} \left(\sum_{t=1}^{T} |p_t - y_t| - \sum_{t=1}^{T} |h(\mathbf{x}_t) - y_t| \right).$$

We first show that this is an impossible mission — no algorithm can obtain a sublinear regret bound even if $|\mathcal{H}| = 2$. Indeed, consider $\mathcal{H} = \{h_0, h_1\}$, where h_0 is the function that always returns 0 and h_1 is the function that always returns 1. An adversary can make the number of mistakes of any online algorithm to be equal to T, by simply waiting for the learner's prediction and then providing the opposite answer as the true answer. In contrast, for any sequence of true answers, $y_1,\ldots,y_T$, let b be the majority of labels in $y_1,\ldots,y_T$, then the number of mistakes of h_b is at most $T/2$. Therefore, the regret of any online algorithm might be at least $T - T/2 = T/2$, which is not a sublinear with T. This impossibility result is attributed to Cover [13].

To sidestep Cover's impossibility result, we must further restrict the power of the adversarial environment. In the following we present two ways to do this.

1.2.1 Realizability Assumption

The first way to sidestep Cover's impossibility result is by making one additional assumption:

- *Realizability:* We assume that all target labels are generated by some $h^\star \in \mathcal{H}$, namely, $y_t = h^\star(x_t)$ for all t. Our goal is to design an algorithm with an optimal mistake bound. Namely, an algorithm for which $M_A(\mathcal{H})$ is minimal. See definition of $M_A(\mathcal{H})$ in the prequel.

Next, we describe and analyze online learning algorithms assuming both a finite hypothesis class and realizability of the input sequence. The most natural learning rule is to use (at any online round) any hypothesis which is consistent with all past examples.

`Consistent`

input: A finite hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  choose any $h \in V_t$
  predict $p_t = h(\mathbf{x}_t)$
  receive true answer $y_t = h^\star(\mathbf{x}_t)$
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

The `Consistent` algorithm maintains a set, V_t, of all the hypotheses which are consistent with $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{t-1}, y_{t-1})$. This set is often called the version space. It then picks any hypothesis from V_t and predicts according to this hypothesis.

Obviously, whenever `Consistent` makes a prediction mistake, at least one hypothesis is removed from V_t. Therefore, after making M mistakes we have $|V_t| \leq |\mathcal{H}| - M$. Since V_t is always nonempty (by the realizability assumption it contains $h^\star$) we have $1 \leq |V_t| \leq |\mathcal{H}| - M$.

Rearranging, we obtain

Corollary 1.1. Let $\mathcal{H}$ be a finite hypothesis class. The `Consistent` algorithm enjoys the mistake bound $M_{\texttt{Consistent}}(\mathcal{H}) \leq |\mathcal{H}| - 1$.

It is rather easy to construct a hypothesis class and a sequence of examples on which `Consistent` will indeed make $|\mathcal{H}| - 1$ mistakes. Next, we present a better algorithm in which we choose $h \in V_t$ in a smarter way. We shall see that this algorithm is guaranteed to make exponentially fewer mistakes. The idea is to predict according to the majority of hypotheses in V_t rather than according to some arbitrary $h \in V_t$. That way, whenever we err, we are guaranteed to remove at least half of the hypotheses from the version space.

`Halving`

input: A finite hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  predict $p_t = \operatorname{argmax}_{r \in \{0,1\}} |\{h \in V_t : h(\mathbf{x}_t) = r\}|$
    (in case of a tie predict $p_t = 1$)
  receive true answer y_t
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

Theorem 1.2. Let $\mathcal{H}$ be a finite hypothesis class. The `Halving` algorithm enjoys the mistake bound $M_{\texttt{Halving}}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$.

Proof. We simply note that whenever the algorithm errs we have $|V_{t+1}| \leq |V_t|/2$. (Hence the name Halving.) Therefore, if M is the total number of mistakes, we have

$$1 \leq |V_{T+1}| \leq |\mathcal{H}| \, 2^{-M}.$$

Rearranging the above inequality we conclude our proof. □

Of course, `Halving`'s mistake bound is much better than `Consistent`'s mistake bound. Is this the best we can do? What is an

optimal algorithm for a given hypothesis class (not necessarily finite)? We will get back to this question in Section 3.

1.2.2 Randomization

In the previous subsection we sidestepped Cover's impossibility result by relying on the realizability assumption. This is a rather strong assumption on the environment. We now present a milder restriction on the environment and allow the learner to randomize his predictions. Of course, this by itself does not circumvent Cover's impossibility result as in deriving the impossibility result we assumed nothing on the learner's strategy. To make the randomization meaningful, we force the adversarial environment to decide on y_t without knowing the random coins flipped by the learner on round t. The adversary can still know the learner's forecasting strategy and even the random bits of previous rounds, but it doesn't know the actual value of the random bits used by the learner on round t. With this (mild) change of game, we analyze the *expected* 0–1 loss of the algorithm, where expectation is with respect to the learner's own randomization. That is, if the learner outputs $\hat{y}_t$ where $\mathbb{P}[\hat{y}_t = 1] = p_t$, then the expected loss he pays on round t is

$$\mathbb{P}[\hat{y}_t \neq y_t] = |p_t - y_t|.$$

Put another way, instead of having the predictions domain being $D = \{0,1\}$ we allow it to be $D = [0,1]$, and interpret $p_t \in D$ as the probability to predict the label 1 on round t. To summarize, we assume:

- *Randomized Predictions and Expected Regret*: We allow the predictions domain to be $D = [0,1]$ and the loss function is still $l(p_t, y_t) = |p_t - y_t|$.

With this assumption it is possible to derive a low regret algorithm as stated in the following theorem.

Theorem 1.3. Let $\mathcal{H}$ be a finite hypothesis class. There exists an algorithm for online classification, whose predictions come from $D = [0,1]$,

that enjoys the regret bound

$$\sum_{t=1}^{T} |p_t - y_t| - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} |h(\mathbf{x}_t) - y_t| \leq \sqrt{0.5 \ln(|\mathcal{H}|) T}.$$

We will provide a constructive proof of the above theorem in the next section.

To summarize, we have presented two different ways to sidestep Cover's impossibility result: realizability or randomization. At first glance, the two approaches seem to be rather different. However, there is a deep underlying concept that connects them. Indeed, we will show that both methods can be interpreted as *convexification* techniques. Convexity is a central theme in deriving online learning algorithms. We study it in the next section.

1.3 Organization and Scope

How to predict rationally is a key issue in various research areas such as game theory, machine learning, and information theory. The seminal book of Cesa-Bianchi and Lugosi [12] thoroughly investigates the connections between online learning, universal prediction, and repeated games. In particular, results from the different fields are unified using the prediction with expert advice framework.

We feel that convexity plays a central role in the derivation of online learning algorithms, and therefore start the survey with a study of the important sub-family of online learning problems, which is called *online convex optimization*. In this family, the prediction domain is a convex set and the loss function is a convex function with respect to its first argument. As we will show, many previously proposed algorithms for online classification and other problems can be jointly analyzed based on the online convex optimization framework. Furthermore, convexity is important because it leads to *efficient* algorithms.

In Section 3 we get back to the problem of online classification. We characterize a standard optimal algorithm for online classification. In addition, we show how online convex optimization can be used for deriving efficient online classification algorithms.

In Section 4 we study online learning in a limited feedback model, when the learner observes the loss value $l(p_t, y_t)$ but does not observe

the actual correct answer y_t. We focus on the classic multi-armed bandit problem and derive an algorithm for this problem based on the online convex optimization algorithmic framework. We also present a low regret algorithm for the general problem of bandit online convex optimization.

Finally, in Section 5 we discuss several implications of online learning to batch learning problems, in which we assume that the examples are sampled i.i.d. from an unknown probability source.

Part of our presentation shares similarities with other surveys on online prediction problems. In particular, Rakhlin's lecture notes [34] and Hazan's book section [22] are good recent surveys on online convex optimization. While part of our presentation shares similarities with these surveys, we sometimes emphasize different techniques. Furthermore, we connect and relate the new results on online convex optimization to classic results on online classification, thus providing a fresh modern perspective on some classic algorithms. A more classic treatment can be found in Blum's survey [8].

1.4 Notation and Basic Definitions

We denote scalars with lower case letters (e.g., x and λ), and vectors with bold face letters (e.g., $\mathbf{x}$ and $\boldsymbol{\lambda}$). The ith element of a vector $\mathbf{x}$ is denoted by $x[i]$. Since online learning is performed in a sequence of rounds, we denote by $\mathbf{x}_t$ the tth vector in a sequence of vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. The ith element of $\mathbf{x}_t$ is denoted by $x_t[i]$.

The **inner product** between vectors $\mathbf{x}$ and $\mathbf{w}$ is denoted by $\langle \mathbf{x}, \mathbf{w} \rangle$. Whenever we do not specify the vector space we assume that it is the d-dimensional Euclidean space and then $\langle \mathbf{x}, \mathbf{w} \rangle = \sum_{i=1}^{d} x[i]w[i]$. Sets are designated by upper case letters (e.g., S). The set of real numbers is denoted by $\mathbb{R}$ and the set of non-negative real numbers is denoted by $\mathbb{R}_+$. The set of natural numbers is denoted by $\mathbb{N}$. For any $k \geq 1$, the set of integers $\{1, \ldots, k\}$ is denoted by $[k]$. Given a predicate π, we use the notation $\mathbf{1}_{[\pi]}$ to denote the indicator function that outputs 1 if π holds and 0 otherwise. The hinge function is denoted by $[a]_+ = \max\{0, a\}$.

The Euclidean (or ℓ_2) **norm** of a vector $\mathbf{w}$ is $\|\mathbf{w}\|_2 = \sqrt{\langle \mathbf{w}, \mathbf{w} \rangle}$. We omit the subscript when it is clear from the context. We also use other ℓ_p

norms, $\|\mathbf{w}\|_p = (\sum_i |w[i]|^p)^{1/p}$, and in particular $\|\mathbf{w}\|_1 = \sum_i |w[i]|$ and $\|\mathbf{w}\|_\infty = \max_i |w[i]|$. A generic norm of a vector $\mathbf{w}$ is denoted by $\|\mathbf{w}\|$ and its **dual norm** is defined as

$$\|\mathbf{x}\|_\star = \max\{\langle \mathbf{w}, \mathbf{x}\rangle : \|\mathbf{w}\| \leq 1\}.$$

The definition of the dual norm immediately implies the inequality

$$\langle \mathbf{w}, \mathbf{z}\rangle \leq \|\mathbf{w}\| \, \|\mathbf{z}\|_\star. \tag{1.3}$$

For the ℓ_2 norm (which is dual to itself), this is the well known Cauchy–Schwartz inequality. For $p, q \geq 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$ we have that the ℓ_p and ℓ_q norms are dual, and Equation (1.3) becomes Holder's inequality.

A function f is called L-**Lipschitz** over a set S with respect to a norm $\|\cdot\|$ if for all $\mathbf{u}, \mathbf{w} \in S$ we have $|f(\mathbf{u}) - f(\mathbf{w})| \leq L\|\mathbf{u} - \mathbf{w}\|$.

The **gradient** of a differentiable function f is denoted by ∇f and the **Hessian** is denoted by $\nabla^2 f$.

Throughout the review, we make use of basic notions from convex analysis. A set S is **convex** if for all $\mathbf{w}, \mathbf{v} \in S$ and $\alpha \in [0,1]$ we have that $\alpha\mathbf{w} + (1-\alpha)\mathbf{v} \in S$ as well. Similarly, a function $f : S \to \mathbb{R}$ is convex if for all $\mathbf{w}, \mathbf{v}$ and $\alpha \in [0,1]$ we have $f(\alpha\mathbf{w} + (1-\alpha)\mathbf{v}) \leq \alpha f(\mathbf{w}) + (1-\alpha) f(\mathbf{v})$.

It is convenient to allow convex functions to output the value ∞. The **domain** of a function f is the set of points on which f is finite. This is convenient, for example, for constraining the solution of an optimization problem to be within some set A. Indeed, instead of solving $\min_{\mathbf{x}\in A} f(\mathbf{x})$ we can solve $\min_{\mathbf{x}} f(\mathbf{x}) + I_A(\mathbf{x})$, where I_A is the function that outputs 0 if $\mathbf{x} \in A$ and ∞ if $\mathbf{x} \notin A$. In the next section we make use of some additional definitions and tools from convex analysis. For clarity, we define them as per need.

The expected value of a random variable, ψ, is denoted by $\mathbb{E}[\psi]$. In some situations, we have a deterministic function h that receives a random variable as input. We denote by $\mathbb{E}[h(\psi)]$ the expected value of the random variable $h(\psi)$. Occasionally, we omit the dependence of h on ψ. In this case, we may clarify the meaning of the expectation by using the notation $\mathbb{E}_\psi[h]$ or $\mathbb{E}_{\psi\sim P}[h]$ if ψ is distributed according to some distribution P.

References

[1] J. Abernethy, P. L. Bartlett, A. Rakhlin, and A. Tewari, "Optimal strategies and minimax lower bounds for online convex games," in *Proceedings of the Annual Conference on Computational Learning Theory*, 2008.

[2] J. Abernethy, E. Hazan, and A. Rakhlin, "Competing in the dark: An efficient algorithm for bandit linear optimization," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, no. 3, 2008.

[3] S. Agmon, "The relaxation method for linear inequalities," *Canadian Journal of Mathematics*, vol. 6, no. 3, pp. 382–392, 1954.

[4] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SICOMP: SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2003.

[5] P. Auer and M. Warmuth, "Tracking the best disjunction," *Machine Learning*, vol. 32, no. 2, pp. 127–150, 1998.

[6] K. Azoury and M. Warmuth, "Relative loss bounds for on-line density estimation with the exponential family of distributions," *Machine Learning*, vol. 43, no. 3, pp. 211–246, 2001.

[7] S. Ben-David, D. Pal, and S. Shalev-Shwartz, "Agnostic online learning," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, 2009.

[8] A. Blum, "On-line algorithms in machine learning," *Online Algorithms*, pp. 306–325, 1998.

[9] J. Borwein and A. Lewis, *Convex Analysis and Nonlinear Optimization*. Springer, 2006.

[10] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "On the generalization ability of on-line learning algorithms," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2050–2057, 2004.

[11] N. Cesa-Bianchi and C. Gentile, "Improved risk tail bounds for on-line algorithms," *Neural Information Proceesing Systems (NIPS)*, 2006.
[12] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.
[13] T. M. Cover, "Behavior of sequential predictors of binary sequences," in *Transactions on Prague Conference on Information Theory Statistical Decision Functions, Random Processes*, pp. 263–272, 1965.
[14] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
[15] V. Dani, T. Hayes, and S. M. Kakade, "The price of bandit information for online optimization," *Advances in Neural Information Processing Systems*, vol. 20, pp. 345–352, 2008.
[16] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, pp. 385–394, 2005.
[17] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Machine Learning*, vol. 37, no. 3, pp. 277–296, 1999.
[18] C. Gentile, "The robustness of the p-norm algorithms," *Machine Learning*, vol. 53, no. 3, pp. 265–299, 2003.
[19] C. Gentile and N. Littlestone, "The robustness of the p-norm algorithms," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, 1999.
[20] G. Gordon, "Regret bounds for prediction problems," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, 1999.
[21] A. J. Grove, N. Littlestone, and D. Schuurmans, "General convergence results for linear discriminant updates," *Machine Learning*, vol. 43, no. 3, pp. 173–210, 2001.
[22] E. Hazan, *The Convex Optimization Approach to Regret Minimization*. 2009.
[23] D. P. Helmbold and M. Warmuth, "On weak learning," *Journal of Computer and System Sciences*, vol. 50, pp. 551–573, 1995.
[24] S. Kakade, S. Shalev-Shwartz, and A. Tewari, "Regularization techniques for learning with matrices," *Journal of Machine Learning Research*, 2012 (To appear).
[25] A. Kalai and S. Vempala, "Efficient algorithms for online decision problems," *Journal of Computer and System Sciences*, vol. 71, no. 3, pp. 291–307, 2005.
[26] J. Kivinen and M. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Information and Computation*, vol. 132, no. 1, pp. 1–64, 1997.
[27] J. Kivinen and M. K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," *Symposium on Theory of Computing* (*STOC*), See also Technical Report UCSC-CRL-94-16, University of California, Santa Cruz, Computer Research Laboratory, pp. 209–218, 1995.
[28] J. Kivinen and M. Warmuth, "Relative loss bounds for multidimensional regression problems," *Machine Learning*, vol. 45, no. 3, pp. 301–329, 2001.
[29] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Machine Learning*, vol. 2, pp. 285–318, 1988.

[30] N. Littlestone, "From on-line to batch learning," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, pp. 269–284, July 1989.
[31] N. Littlestone, "Mistake bounds and logarithmic linear-threshold learning algorithms," Ph.D. Thesis, University of California at Santa Cruz, 1990.
[32] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, pp. 212–261, 1994.
[33] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.
[34] A. Rakhlin, *Lecture Notes on Online Learning*. draft, 2009.
[35] A. Rakhlin, K. Sridharan, and A. Tewari, "Online learning: Random averages, combinatorial parameters, and learnability," in *Neural Information Processing Systems (NIPS)*, 2010.
[36] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin American Mathematical Society*, vol. 55, pp. 527–535, 1952.
[37] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–407, 1958. (Reprinted in *Neurocomputing*, (MIT Press, 1988)).
[38] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.
[39] S. Shalev-Shwartz, "Online learning: Theory, algorithms, and applications," Ph.D. Thesis, The Hebrew University, 2007.
[40] S. Shalev-Shwartz and Y. Singer, "A primal-dual perspective of online learning algorithms," *Machine Learning Journal*, vol. 69, no. 2, pp. 115–142, 2007.
[41] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, vol. 64. LibreDigital, 2003.
[42] V. N. Vapnik, *Statistical Learning Theory*. Wiley, 1998.
[43] Volodimir G. Vovk, "Aggregating Strategies," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, pp. 371–383, 1990.
[44] C. Zalinescu, *Convex Analysis in General Vector Spaces*. River Edge, NJ: World Scientific Publishing Co. Inc., 2002.
[45] T. Zhang, "Data dependent concentration bounds for sequential prediction algorithms," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, pp. 173–187, 2005.
[46] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *International Conference on Machine Learning*, 2003.