VI. Guide to lectures for sixth two weeks[1]

Updated April 13, 2017

The main topic for the next three lectures is online algorithms for convex optimization. In the models for statistical learning problems discussed earlier in the course it is assumed the data Z^n is generated by independent draws from a probability distribution P on Z. The probability distribution P is unknown, and for a problem to be PAC learnable, there should be an algorithm that is probably almost correct, where the probability of almost correctness, $1-\delta$, should converge to one uniformly over all P ins some class $\mathcal{P}$. Thus, the definition of PAC has some min-max aspect.

We can buy in to the minimax or modeling philosophy more fully by dropping the assumption the samples are drawn at random from some distribution P. Rather, we could consider the samples $z_1, z_2, \ldots$ to be arbitrary. The learner can be viewed as player, and the variables $z_1, z_2, \ldots$ can be considered to be chosen by an adversary. Usually in this context we won't be modeling the adversary, but just assume the adversary could come up with an arbitrary sequence $z_1, z_2, \ldots$. While the problem formulation is somewhat different than the statistical learning framework we have seen earlier in the course, much of the same tools we have seen can be applied to the learning problems.

The performance analysis of stochastic gradient descent we looked at in the last section is about the performance of the stochastic gradient descent algorithm, such that the data points $z_1, \ldots, z_n$ were fixed, The randomness dealt with in the algorithms was due to randomization in the algorithm. The guarantees were uniform in $z_1, \ldots, z_n$.

We shall focus on the online, or adversarial, function minimization problem.[2] An influential paper in this line of work is the one of [5], although there is a long history of literature on learning in an adversarial context.

Lecture Thursday, April 11, completed discussion of SGD with no convexity (end of supplementary notes part 5) and begin disucssion of online convex programming in Section 1.

1 Online convex programing and a regret bound

The paper of Zinkevich [5] sparked much interest in the adversarial framework for modeling online function minimization. The paper shows that a projected gradient descent algorithm achieves zero asymptotic average regret rate for minimizing an arbitrary sequence of uniformly Lipschitz convex functions over a closed bounded convex set in $\mathbb{R}^d$. The framework involves objects familiar to us, although the terminology is a bit closer to game theory.

- Let $\mathcal{F}$ be a nonempty, closed, convex subset of a Hilbert space $\mathcal{H}$. It is assumed $\mathcal{F}$ is bounded, so $D \triangleq \max\{\|f - f'\| : f, f' \in \mathcal{F}\|\} < \infty$. The player selects actions from $\mathcal{F}$.
- Let Z be a set, denoting the possible actions of the adversary.
- Let $\ell : \mathcal{F} \times \mathsf{Z} \mapsto \mathbb{R}_+$. The interpretation is that $\ell(f_t, z_t)$ is the loss to the player for step t. We sometimes use the notation $\ell_t : \mathsf{Z} \to \mathbb{R}_+$, defined by $\ell_t(f) = \ell(f, z_t)$.
- Suppose the player has access to an algorithm that can compute $\ell_t(f)$ and $\nabla \ell_t(f)$ for a given f.
- Suppose the player has access to an algorithm that can calculate $\Pi(f)$ for any $f \in \mathbb{R}^d$, where $\Pi : \mathcal{H} \to \mathcal{F}$ is the projection mapping: $\Pi(f) = \arg\min\{\|f - f'\|^2 : f' \in \mathcal{F}\}$, that maps any $f \in \mathcal{H}$ to a nearest point in $\mathcal{F}$.

[1] Actually, the lectures for part V ran through Tuesday, April 3, so this supplement pertains to lectures from April 5-12.

[2] There is work on a related *online learning* problem for which the player can select a label $\widehat{y}_t$ at each time after observing x_t. Then the actual label y_t is revealed and the player incurs the loss $\ell((x_t, y_t), \widehat{y}_t)$. A simple case is the 0-1 loss. An algorithm A for this problem produce a label $\widehat{y}_t = A(x_1, \ldots, x_t, y_1, \ldots, y_{t-1})$ for each t. See [4] for an introduction.

- $T \geq 1$ represents a *time horizon* of interest

The online convex optimization game proceeds as follows.

- At each time step t from 1 to T, the player chooses $f_t \in \mathcal{F}$
- The adversary chooses $z_t \in \mathsf{Z}$
- The player observes z_t and incurs the loss $\ell(f_t, z_t)$.

Roughly speaking, the player would like to select the sequence of actions (f_t) to minimize the total loss for some time-horizon T, or equivalently, minimize the corresponding average loss per time step:

$$J_T((f_t)) \triangleq \sum_{t=1}^{T} \ell(f_t, z_t.) \qquad L_T((f_t)) \triangleq \frac{1}{T} J_T((f_t)).$$

If we wanted to emphasize the dependence on z^T we could have writen $J_T((f_t), z^T)$ and $L_T((f_t), z^T)$ instead. A possible strategy of the player is to use a fixed $f^* \in \mathcal{F}$ for all time, in which case we write the total loss as $J_T(f^*) \triangleq \sum_{t=1}^{T} \ell(f^*, z_t.)$ and the loss per time step as $L_T(f^*) = \frac{1}{T} J_T(f^*)$. Note that $L_T(f^*)$ is the empirical loss for f^* for T samples. If the player is extremely lucky, or if for each t a genie knowing z_t in advance reveals an optimal choice to the player, the player could use $f_t^{\text{genie}} \triangleq \arg\min_{z \in \mathsf{Z}} \ell(f, z_t)$. Typically it is unreasonable to expect a player without knowing z_t before selecting f_t to achieve, or even nearly achieve, the genie-assisted minimum loss.

It turns out that a realistic goals is for the player to make selections that perform nearly as well as *any fixed strategy* f^* that could possibly be selected after the sequence z^T is revealed. Specifically, if the player uses (f_t) then the *regret* (for not using an optimal fixed strategy) is defined by:

$$R_T((f_t)) = \inf_{f^* \in \mathcal{F}} J_T((f_t)) - J_T(f^*),$$

where for a particular f^*, $J_T((f_t)) - J_T(f^*)$ is the regret for using (f_t) instead of f^*. We shall be interested in strategies the player can use to (approximately) minimize the regret. Even this goal seems ambitious, but one important thing the player can exploit is that the player can let f_t depend on t, whereas the performance the player aspires to match is that of the best policy that is constant over all steps t.

Zinkevich [5] showed that the projected gradience descent algorithm, defined by

$$f_{t+1} = \Pi(f_t - \alpha_t \nabla \ell_t(f_t)), \tag{1}$$

meets some performance guarantees for the regret minimization problem. Specifically, under convexity and the assumption that the functions ℓ_t are all L-Lipschtiz continuous, Zinkevich showed that regret $O(LD\sqrt{T})$ is achievable by gradient descent. Under such assumptions the $\sqrt{T}$ scaling is the best possible (see problem set 6). The paper of Hazan, Agarwal, and Kale [3] shows that if, in addition, the functions ℓ_t are all σ-strongly convex for some $\sigma > 0$, then gradient descent can achieve $O\left(\frac{L^2}{\sigma} \log T\right)$ regret. The paper [3] ties together several different previous approaches including follow-the-leader, exponential weighting, Cover's algorithm, and gradient descent. The following theorem combines the analysis of [5] for the case of Lipschitz continuous objective functions and the analysis of [3] for strongly convex functions. The algorithms used for the two cases differ only in the stepsize selections. Recall that D is the diameter of $\mathcal{F}$.

Theorem 1. *Suppose $\ell(\cdot, z)$ is convex, L-Lipschtiz continuous for each z and suppose the gradient projection algorithm (1) is run with stepsize multipliers $(\alpha_t)_{t \geq 1}$.*
(a) If $\alpha_t = \frac{c}{\sqrt{t}}$ for $t \geq 1$, then the regret is bounded as follows:

$$R_T((f_t)) \leq \frac{D^2 \sqrt{T}}{2c} + \left(\sqrt{T} - \frac{1}{2}\right) L^2 c,$$

which for $c = \frac{D}{L\sqrt{2}}$ *gives:*

$$R_T((f_t)) \le DL\sqrt{2T}.$$

(b) If, in addition, $\nabla\ell(\cdot, z)$ *is* σ*-strongly convex for some* $\sigma > 0$ *and* $\alpha_t = \frac{1}{\sigma t}$ *for* $t \ge 1$*, then the regret is bounded as follows:*

$$R_T((f_t)) \le \frac{L^2(1+\log T)}{2\sigma}.$$

Proof. Most of the proof is the same for parts (a) and (b), where for part (a) we simply take $\sigma = 0$. Let $f_t^\flat = f_t - \alpha_t \nabla\ell_t(f_t)$, so that $f_{t+1} = \Pi(f_{t+1}^\flat)$. Let $f^* \in \mathcal{F}$ be any fixed policy. Note that

$$\begin{aligned} f_{t+1}^\flat - f^* &= f_t - f^* - \alpha_t \nabla\ell_t(f_t) \\ \|f_{t+1}^\flat - f^*\|^2 &= \|f_t - f^*\|^2 - 2\alpha_t\langle f^t - f^*, \nabla\ell_t(f_t)\rangle + \alpha_t^2\|\nabla\ell_t(f_t)\|^2. \end{aligned}$$

By the contraction property of Π, $\|f_{t+1} - f^*\| \le \|f_{t+1}^\flat - f^*\|$. Also, by the Lipschitz assumption, $\|\nabla\ell_t(f_t)\| \le L$. Therefore,

$$\|f_{t+1} - f^*\|^2 \le \|f_t - f^*\|^2 - 2\alpha_t\langle f^t - f^*, \nabla\ell_t(f_t)\rangle + \alpha_t^2 L^2$$

or, equivalently,

$$2\big\langle f_t - f^*, \nabla\ell_t(f_t)\big\rangle \le \frac{\|f_t - f^*\|^2 - \|f_{t+1} - f^*\|^2}{\alpha_t} + \alpha_t L^2. \tag{2}$$

(Equation (2) captures well the fact that this proof is based on the use of $\|f_t - f^*\|$ as a potential function. The only property of the gradient vectors $\nabla\ell_t(f_t)$ used so far is $\|\nabla\ell_t(f_t)\| \le L$. The specific choice of using gradient vectors is exploited next, to bound differences in the loss function.) The strong convexity of ℓ_t implies $\ell_t(f^*) - \ell_t(f_t) \ge \langle f^* - f_t, \nabla\ell_t(f_t)\rangle + \frac{\sigma}{2}\|f^* - f_t\|^2$, or equivalently,
$2(\ell_t(f_t) - \ell_t(f^*)) \le 2\langle f_t - f^*, \nabla\ell_t(f_t)\rangle - \sigma\|f_t - f^*\|^2$, so

$$2(\ell_t(f_t) - \ell_t(f^*)) \le \frac{\|f_t - f^*\|^2 - \|f_{t+1} - f^*\|^2}{\alpha_t} + \alpha_t L^2 - \sigma\|f_t - f^*\|^2 \tag{3}$$

We shall use the following for $1 \le t \le T-1$:

$$\frac{\|f_t - f^*\|^2 - \|f_{t+1} - f^*\|^2}{\alpha_t} = \frac{\|f_t - f^*\|^2}{\alpha_t} - \frac{\|f_{t+1} - f^*\|^2}{\alpha_{t+1}} + \left(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t}\right)\|f_{t+1} - f^*\|^2.$$

Summing each side of (3) from $t = 1$ to T yields:

$$\begin{aligned} 2(J_T((f_t)) - J_T(f^*)) &\le \left(\frac{1}{\alpha_1} - \sigma\right)\|f_1 - f^*\|^2 - \frac{1}{\alpha_T}\|f_{T+1} - f^*\|^2 \\ &\quad + \sum_{t=1}^{T-1}\left(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t} - \sigma\right)\|f_{t+1} - f^*\|^2 + L^2\sum_{t=1}^{T}\alpha_t \\ &\le D^2\left(\frac{1}{\alpha_1} - \sigma + \sum_{t=1}^{T-1}\left(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t} - \sigma\right)\right) + L^2\sum_{t=1}^{T}\alpha_t \\ &\le D^2\left(\frac{1}{\alpha_T} - \sigma T\right) + L^2\sum_{t=1}^{T}\alpha_t \end{aligned} \tag{4}$$

(Part (a)) If $\sigma = 0$ the bound (4) becomes

$$2(J_T((f_t)) - J_T(f^*)) \le \frac{D^2}{\alpha_T} + L^2\sum_{t=1}^{T}\alpha_t \tag{5}$$

Now if $\alpha_1 = \frac{c}{\sqrt{t}}$, then

$$\sum_{t=1}^{T} \alpha_t = c + \sum_{t=2}^{T} \frac{c}{\sqrt{t}} \leq c + c \int_{t=1}^{T} \frac{cdt}{\sqrt{t}} = (2\sqrt{T} - 1)c$$

and we get

$$J_T((f_t)) - J_T(f^*) \leq \frac{D^2\sqrt{T}}{2c} + \left(\sqrt{T} - \frac{1}{2}\right) L^2 c$$

If $c = \frac{D}{L\sqrt{2}}$ then $J_T((f_t)) - J_T(f^*) \leq DL\sqrt{2T}$. Since $f^* \in \mathcal{F}$ is arbitrary it follows that $R_T((f_t)) \leq DL\sqrt{2T}$.

(Part (b)) For the case of $\sigma > 0$ and $\alpha_t = \frac{1}{\sigma t}$ for all $t \geq 1$, then the first term on the right-hand side of (4) is zero, and

$$\sum_{t=1}^{T} \alpha_t = \frac{1}{\sigma}\left(1 + \sum_{t=2}^{T-1} \frac{1}{t}\right) \leq \frac{1 + \log T}{\sigma},$$

so part (b) of the theorem follows from (4) and the fact $f^* \in \mathcal{F}$ is arbitrary. □

2 Connection between online game and stochastic analysis

The paper [1] focuses on the connection between the statistical learning framework based with statistical assumptions on the samples, and the adversarial model for online function minimization. A problem in the homework set based on dynamic programming is related.

3 Online Perceptron Algorithm

The perceptron algorithm of Rossenblatt (1958) is an iterative algorithm for training binary classifiers of the form $y = \mathsf{sgn}(\langle f, x\rangle)$. In this section we show that the perceptron algorithm can be viewed as an instance of the gradient descent algorithm for a certain online convex optimization problem, and adapt the proof of Theorem 1 to bound the total number of iterations in the realizable case. Let the original data set be denoted by $\widetilde{z}^n$. At each time step t, the gradient descent algorithm will be applied to a sample $z_t = \widetilde{z}_{I_t}$ for a choice of index $I_t \in [n]$ to be described in what follows. We use z and (x, y) interchangeably, and, similarly, z_t and (x_t, y_t) interchangeably. Consider the surrogate loss function $\widetilde{\ell}(f, z) = (1 - y\langle f, z\rangle)_+$, which is the penalized version of 0-1 loss arising from use of the hinge penalty function $\varphi(u) = (1 + u)_+$. At step t the learner selects f_t, and then the adversary selects a convex loss function. Usually we would think to use $\widetilde{\ell}_t(\cdot) = \widetilde{\ell}(\cdot, z_t)$. However, the rules of online convex function minimization allow the adversary to present a different sequence of convex functions $\ell_t : t \geq 1$, determined as follows:

- If $y_t\langle f, z_t\rangle \geq 0$ (i.e. if f_t correctly classifies z_t), then $\ell_t \equiv 0$.
- If $y_t\langle f, z_t\rangle < 0$ then $\ell_t \equiv \widetilde{\ell}_t$.

Note that ℓ_t is convex for each t and

$$\nabla \ell_t(f_t) = \begin{cases} 0 & \text{if } y_t = \mathsf{sgn}(\langle f_t, x_t\rangle) \\ -y_t x_t & \text{else.} \end{cases}$$

The gradient descent algorithm, with no projection and constant stepsize α, becomes::

$$f_{t+1} = \begin{cases} f_t & \text{if } y_t = \mathsf{sgn}(\langle f_t, x_t\rangle) \\ f_t + \alpha y_t x_t & \text{else,} \end{cases} \tag{6}$$

where we use the initial state $f_1 = 0$. Since f_t is proportional to α for all t, the classifiers f_t are all proportional to α, so the 0-1 loss performance of the algorithm does not depend on α. We included the stepsize $\alpha > 0$ only for the proof. We now specify how the index I_t is chosen. If there is some sample that f_t does not correctly label, then I_t is the index of such a sample. Otherwise, $I_t \in [n]$ is arbitrary. The classical perceptron algorithm corresponds to this choice of I_t, the updatet rule (6), and stopping at the first time t it is found that f_t separates the original data.

Proposition 1. *(Perceptron classification, realizable case) Let $L \geq \max_{1\leq i\leq n} \|x_i\|$ and $B \geq \min\{\|f^*\| : y_i\langle f^*, x_i\rangle \geq 1 \text{ for } i \in [n]\}$. At most B^2L^2 updates are needed for the perceptron algorithm to find a separating classifier.*

Proof. (Variation of the proof of Theorem 1.) By the assumptions, ℓ_t is L-Lipschitz continuous for all t. Let f^* be a vector so that $\|f^*\| \leq B$ and $y_i\langle f^*, x_i\rangle \geq 1$ for $i \in [n]$. By (3) with $\sigma = 0$ and $\alpha_t = \alpha$ for all t,

$$2(\ell_t(f_t) - \ell_t(f^*)) \leq \frac{\|f_t - f^*\|^2 - \|f_{t+1} - f^*\|^2}{\alpha} + \alpha L^2$$

Summing over t from 1 to T yields

$$\begin{aligned} 2(J_T((f_t)) - J_T(f^*)) &\leq \frac{1}{\alpha}\|f_1 - f^*\|^2 - \frac{1}{\alpha}\|f_{T+1} - f^*\|^2 + \alpha L^2 T \\ &\leq \frac{B^2}{\alpha} + \alpha L^2 T \end{aligned}$$

Now $\ell_t(f_t) \geq 1$ if f_t does not separate the data, whereas $\ell_t(f^*) = 0$ for all t. Thus, if none of $f_1, \dots f_T$ separate the data, then $2T \leq 2(J_T((f_t)) - J_T(f^*)) \leq \frac{B^2}{\alpha} + \alpha L^2 T$ for any $\alpha > 0$. Taking $\alpha = \frac{L}{B\sqrt{T}}$ yields $T \leq BL$. Thus, at most $(BL)^2$ updates are needed for the algorithm to find a separating classifier. □

4 On the Generalization Ability of On-Line Learning Algorithms

This section is based on [2]. An instance of the on-line convex function minimization framework of Section 1 is represented by a tuple $(\mathcal{F}, \mathsf{Z}, \ell)$. An online algorithm A prescribes an action f_1 and then, for each $t \geq 1$, an action $f_{t+1} = A(f_1, \dots, f_t, z_1, \dots, z_t)$. Section 1 considered regret for arbitrary sequences of samples. Consider instead, the statistical learning framework, such that the samples $Z_1, Z_2, \dots, Z_n$ are independent and identically distributed random variables with values in Z and some probability distribution P. Consider using an algorithm A for online convex function minimization with $T = n$, so the algorithm makes one pass through the data using the samples $Z_1, \dots, Z_n$ in order.

The sequence $f_1, f_2, \dots, f_n$ produced by A is random, due to the randomness of the Z's. However, for each $t \in [n]$, f_t is determined by A and $Z_1, \dots, Z_{t-1}$, so f_t is independent of Z_t. Therefore, given f_t, $\ell(f_t, Z_t)$ is an unbiased estimator of the generalization performance of f_t. In essence, each subsequent sample Z_t is a test sample for f_t. It is therefore to be expected that online algorithms in the statistical framework have good generalization ability. An application of the Azuma-Hoeffding inequality makes this precise:

Theorem 2. *(Generalization ability of on-line algorithms) Suppose ℓ is bounded, with values in $[0, 1]$. Suppose $Z_1, Z_2, \dots$ are independent and identically distributed, and let (f_t) be produced by an online learning algorithm A. Then for any $T \geq 1$, with probability at least $1 - \delta$,*

$$\frac{1}{T}\sum_{i=1}^{T} L(f_t) \leq \frac{1}{T}\sum_{t=1}^{T} \ell(f_t, Z_t) + \sqrt{\frac{2\log\frac{1}{\delta}}{T}}, \tag{7}$$

where $L(f)$ denotes the generalization loss of a hypothesis f for the probability distribution P used to generate the samples, and $\frac{1}{T}\sum_{t=1}^{T} \ell(f_t, Z_t) = L_T((f_t))$ is the average loss per sample suffered by the online learner. Furthermore, if $\ell(\cdot, z)$ is convex for each z fixed, and $\overline{f}_T = \frac{1}{T}\sum_{t=1}^{T} f_t$, then with probability at least $1 - \delta$,

$$L(\overline{f}_T) \leq \frac{1}{T}\sum_{t=1}^{T} \ell(f_t, Z_t) + \sqrt{\frac{2\log\frac{1}{\delta}}{T}}. \tag{8}$$

Proof. Let $Y_0 = 0$ and $Y_t = \sum_{s=1}^{t}(L(f_s) - \ell(f_s, Z_s))$. Since $\mathbb{E}\left[L(f_s) - \ell(f_s, Z_s)|Z_1, \ldots, Z_{s-1}\right] = 0$ for all s, (Y_t) is a martingale. Also, $|Y_t - Y_{t-1}| \leq |L(f_t)| + |\ell(f_t, Z_t)| \leq 2$ for all t. Thus, by the Azuma-Hoeffding inequality with $c = 2$, and $\epsilon > 0$,

$$\mathbb{P}\left\{Y_T \geq \epsilon T\right\} \leq \exp\left\{-\frac{2\epsilon^2 T^2}{4T}\right\} = \exp\left\{-\frac{\epsilon^2 T}{2}\right\}.$$

Setting $\epsilon = \sqrt{\frac{2\log\frac{1}{\delta}}{T}}$ and rearranging yields (7). For the second part, the assumed convexity of $\ell(\cdot, z)$ implies convexity of L, so by Jensen's inequality $L(\overline{f}_T) \leq \frac{1}{T}\sum_{i=1}^{T} L(f_t)$. Therefore, (8) follows from (7). □

An interpretation of (7) is that the average (over $t \in [T]$) generalization loss of the hypotheses generated by the online learning algorithm A is not much larger than $L_T((f_t))$, with high probability. Recall that $L_T((f_t))$ is the loss incurred by the learner, which is version of empirical loss, although for a sequence of hypotheses (f_t) rather than a single hypothesis. The bound (8) shows that, in the case $\ell(f, z)$ is convex in f, the hypotheses of the algorithm can be combined to yield a hypothesis $\overline{f}_T$ that has generalization loss, in the standard sense, not much larger than $L_T((f_t))$ incurred by the online learning algorithm.

In the statistical learning framework, if $\ell(f, z)$ is convex in f, Theorem 1 implies that the average $\overline{f}_T$ of the hypotheses (f_t) provided by the projected gradient descent algorithm represents an asymptotic ERM (AERM) algorithm. As we've seen earlier, generalization and AERM together provide consistency; the following corollary illustrates that point.[3]

Corollary 1. *Suppose $\mathcal{F}$ is a closed convex subset of a Hilbert space with finite diameter D. Suppose $\ell : \mathcal{F} \times \mathsf{Z} \to [0, 1]$ is such that $\ell(\cdot, z)$ is L-Lipschitz continuous for each fixed $z \in \mathsf{Z}$. Suppose $Z_1, Z_2, \ldots, Z_n$ are independent and identically distributed. Let $\overline{f}_n = \frac{1}{n}\sum_{t=1}^{n} f_t$, where $f_1, \ldots, f_n$ is produced by the projected gradient descent algorithm with step size $\alpha_t = \frac{D}{L\sqrt{2t}}$ for $t \geq 1$ (as in Theorem 1(a)). Then with probability at least $1 - 2\delta$,*

$$L(\overline{f}_n) \leq L^* + DL\sqrt{\frac{2}{n}} + \sqrt{\frac{8\log\frac{1}{\delta}}{n}},$$

Proof. Let f^* minimizes the generalization loss: $L^* = L(f^*)$. The last statement of Theorem 2 with $T = n$ implies that, with probability at least $1 - \delta$,

$$L(\overline{f}_n) \leq \frac{1}{n}\sum_{t=1}^{n} \ell(f_t, Z_t) + \sqrt{\frac{2\log\frac{1}{\delta}}{n}}.$$

Theorem 1(a) implies that, with probability one,

$$\frac{1}{n}\sum_{t=1}^{n} \ell(f_t, Z_t) \leq \frac{1}{n}\sum_{t=1}^{n} \ell(f^*, Z_t) + DL\sqrt{\frac{2}{n}}.$$

The Azuma-Hoeffding inequality, as used in the proof of Theorem 2, and the choice $L(f^*) = L^*$, implies that with probability at least $1 - \delta$,

$$\frac{1}{n}\sum_{t=1}^{n} \ell(f^*, Z_t) \leq L^* + \sqrt{\frac{2\log\frac{1}{\delta}}{n}}.$$

By the union bound, with probability at least $1 - 2\delta$, the previous three centered inequalities all hold, implying the corollary. □

[3]Earlier we also found that, in a certain sense, consistency is equivalent to stability with respect to replace one sample perturbations. In the context of this section the Azuma-Hoeffding inequality is used to show consistency; stability is not used in this section.

If $\ell(f, z)$ is not convex in f, the bound (7) still implies that at least one of the hypotheses (f_t) generated by the algorithm has generalization loss $L(f_t)$ not much larger than the loss of the on-line learning algorithm. Moreover, for each t, the generalization loss, $L(f_t)$, can be estimated by applying f_t not only to Z_t, but to $Z_t, Z_{t+1}, \ldots, Z_T$, to help identify a value t^* so $L(f_{t^*}) \approx \min_t L(f_t)$. See [2] for details. This provides a single output hypothesis $f_{\widehat{t}}$ with good generalization performance, even for non-convex loss. If somehow an AERM property is also true, a consistency result along the lines of Corollary 1 could be achieved for nonconvex loss.

References

[1] J. Abernethy, A. Agarwal, P.L. Bartlett, and A. Rakhlin. A stochastic view of optimal regret through minimax duality. *arXiv preprint arXiv:0903.5328*, 2009.

[2] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

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

[4] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[5] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. 2003.