Introduction to Statistical Learning Theory

Lecture 10

We now study a different kind of learning - online learning.

At each step the learner gets x_t, predicts p_t, receives true label y_t and finally suffers loss $\ell(p_t, y_t)$. We wish to minimize $\sum_{i=1}^{T} \ell(p_t, y_t)$.

Main differences: No separate training stage. No i.i.d assumption - x_t, y_t can be adversarial.

Examples: Spam filtering, financial predictions.

Assume $\ell(p_t, y_t) \in [0, 1]$. Since data might be adversarial, $\sum_{i=1}^{T} \ell(p_t, y_t)$ might be T no matter what we do.

We first thing is to consider regret: There is some (known) hypothesis class $\mathcal{H}$, but instead of $\sum_{i=1}^{T} \ell(p_t, y_t)$, we look at $Regret_T(\mathcal{H}) = \sum_{i=1}^{T} \ell(p_t, y_t) - \min_{h \in \mathcal{H}} \sum_{i=1}^{T} \ell(h(x_t), y_t)$. We compare to the best (fixed) hypothesis.

In classification, an adversary can still make the regret at least $T/2$.

To get sublinear regret we look at two further restrictions:

Realizable case: There is some (known) hypothesis class $\mathcal{H}$ and $h^* \in \mathcal{H}$, such that $\ell(h^*(x_t), y_t) \equiv 0$.

Randomization: The learner gives a distribution, which the adversary knows (but not the random outcome) and suffers the expected loss.

This turns discrete problems to convex problems, and $0-1$ loss to a Lipschitz loss function.

For the realizable case, regret is just the number of mistakes.

we define the mistake bound of algorithm A, $M_A(\mathcal{H})$ as the maximum number of mistakes over any sequence of inputs and labels $h(x_t)$.

For binary finite hypothesis spaces, we have this simple algorithm:

Algorithm `Halving`

Input: Hypothesis space $\mathcal{H}$
Initialize: $V_1 = \mathcal{H}$
for t=1,...,T **do**
  recieve x_t
  predict $p_t = \arg\max_{b \in \{\pm 1\}} |\{h \in V_t : h(x_t) = b\}|$
  recieve y_t
  Update: $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$.
end for

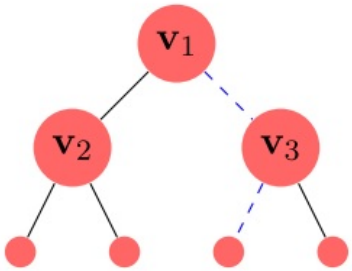
Theorem 1.1

The halving algorithm has mistake bound $M_{halving}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$.

Proof- The halving algorithm is a majority vote. So for each wrong turn t we have $|V_{t+1}| \leq |V_t|/2$,
so if M mistakes where made in T steps then $1 \leq |V_{T+1}| \leq 2^{-M}|\mathcal{H}|$

Consider an adversary that wishes to make you err the first T rounds.

He need to be able to produce $x_t = f(x_1, p_1, ..., x_{t-1}, p_{t-1})$ and $h^* \in \mathcal{H}$ such that $\forall t : h^*(x_t) \neq p_t$. We can think of it as a binary tree of depth T, each internal node is an input and each leaf a hypothesis.



	h_1	h_2	h_3	h_4
$\mathbf{v}_1$	0	0	1	1
$\mathbf{v}_2$	0	1	*	*
$\mathbf{v}_3$	*	*	0	1

Fig. 3.1 An illustration of a shattered tree of depth 2. The *dashed blue* path corresponds to the sequence of examples $((\mathbf{v}_1, 1), (\mathbf{v}_3, 0))$. The tree is shattered by $\mathcal{H} = \{h_1, h_2, h_3, h_4\}$, where the predictions of each hypothesis in $\mathcal{H}$ on the instances $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ is given in the table (the "*" mark means that $h_j(\mathbf{v}_i)$ can be either 1 or 0).

Figure: From "Online Learning and Online Convex Optimization". Shai Shalev-Shwartz

Definition 1.2 (Shattered tree)

A shattered tree of depth d is a complete binary tree of depth d. To each internal nodes i there is $v_i \in \mathcal{X}$. To each leaf j corresponds a hypothesis $h_j \in \mathcal{H}$. Each sequence of predictions $p_1, ..., p_d$ defines a path in the tree $v_{t_1}, v_{t_2}, .., v_{t_d}$ from root $v_{t_1} = v_1$ to leaf $v_{f_d} = h$, 0 moves right 1 moves left, so that $h(v_{t_i}) \neq p_i$.

Definition 1.3 (Littlestones dimension)

Littlestones dimension Ldim(H) is the maximal integer T such that there exist a shattered tree of depth T.

By definition, if $LDim(\mathcal{H}) = T$ then for any algorithm A, $M_A(\mathcal{H}) \geq T$.

Example 1: For finite $\mathcal{H}$, $LDim(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$.

Example 2: $\mathcal{X} = [d]$, $\mathcal{H} = \{h_1, ..., h_d\}$ where $h_d(x) = 1$ iff $x = d$. $LDim(\mathcal{H}) = 1$.

Example 3: If $VC(\mathcal{H}) = d$ then $LDim(\mathcal{H}) \geq d$. Proof - Consider tree with same nodes at each level.

Example 4: $\mathcal{X} = [0, 1]$, $\mathcal{H}$ threshold functions. $VC(\mathcal{H}) = 1$ but $LDim(\mathcal{H}) = \infty$.

We now present an optimal learning algorithm for the realizable setting:

Algorithm `Standard Optimal Algorithm`

Input: Hypothesis space $\mathcal{H}$
Initialize: $V_1 = \mathcal{H}$
for t=1,...,T **do**
recieve x_t
for $b \in \{\pm 1\}$ set $V_t^b = \{h \in V_t : h(x_t) = b\}$
predict $p_t = \arg\max_{b \in \{\pm 1\}} LDim(V_t^b)$
recieve y_t
Update: $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$.
end for

Theorem 1.4

The SOA algorithm has mistake bound $M_{SOA}(\mathcal{H}) = LDim(\mathcal{H})$.

Proof - It is enough to show that for every error $LDim(V_{t+1}) \leq LDim(V_t) - 1$.

Assume by contrary this is not the case. This means $LDim(V_t^0) = LDim(V_t^1) = LDim(V_t)$. This means we can build a shattering tree for $LDim(V_t)$ of depth $LDim(V_t) + 1$ and a contradiction.

We now turn to convex hypothesis spaces and loss function.

As we said before, classification can be convexified via randomization.

Randomization helps the learner, as the adversary does not know the random outcome (only the distribution).

Another way to convexify - surrogate loss, e.g. hinge loss.

Our basic analysis tool will be online convex optimization.

The scenario is almost the same - We have a convex set $\mathcal{S}$, predict a vector $w_t \in \mathcal{S}$, receive a convex loss function $f_t(w)$, and suffer loss $f_t(w_t)$.

For each vector u we define $Regret_T(u) = \sum_{i=1}^{T} f_t(w_t) - \sum_{i=1}^{T} f_t(u)$.

For a set U we define $Regret_T(U) = \max_{u \in U} Regret_T(u)$.

Usually $U = \mathcal{S}$, but not always, e.g. randomization.

A simple OCO algorithm is the following:

Algorithm `Follow-The-Leader (FTL)`

Input: Convex set $\mathcal{S}$
for t=1,...,T **do**
predict $w_t = \arg\min_{w \in \mathcal{S}} \sum_{i=1}^{t-1} f_t(w)$
end for

The main theorem to bound the regret is

Theorem 2.1

Let $w_1, ..., w_T, ...$ be the outputs of the FTL. For all $u \in \mathcal{S}$ we have

$$Regret_T(u) \leq \sum_{t=1}^{T} (f_t(w_t) - f_t(w_{t+1}))$$

Proof - we need to show that

$$\sum_{t=1}^{T}\left(f_t(w_t)-f_t(u)\right)\leq\sum_{t=1}^{T}\left(f_t(w_t)-f_t(w_{t+1})\right) \tag{1}$$

that means, $\sum_{t=1}^{T} f_t(u)\geq\sum_{t=1}^{T} f_t(w_{t+1})$. Proof is by induction.

The case $T=1$ is by the definition of w_2.

Assume by induction that for all u, $\sum_{i=t}^{T-1} f_t(w_{t+1})\leq\sum_{t=t}^{T-1} f_t(u)$. We now get

$$\sum_{t=t}^{T} f_t(w_{t+1})\leq\sum_{t=t}^{T-1} f_t(u)+f_T(w_{T+1}) \tag{2}$$

This holds for all u so we can chose $u=w_{T+1}$ to conclude

$$\sum_{t=t}^{T} f_t(w_{t+1})\leq\sum_{t=t}^{T} f_t(w_{T+1})=\min_{u\in\mathcal{S}}\sum_{t=t}^{T} f_t(u) \tag{3}$$

Consider quadratic optimization - $f_t(w) = \frac{1}{2}||w - z_t||_2^2$. Assume $\mathcal{S} = \mathbb{R}^d$, and $z_t \in \mathbb{R}^d$.

The FTL prediction is $w_t = \frac{1}{t-1}\sum_{i=1}^{t-1} z_t$.

We can write $w_{t+1} = \left(1 - \frac{1}{t}\right) w_t + \frac{1}{t} z_t$, so $w_{t+1} - z_t = \left(1 - \frac{1}{t}\right)(w_t - z_t)$.

$$f_t(w_t) - f_t(w_{t+1}) = \frac{1}{2}||w_t - z_t||_2^2 - \frac{1}{2}||w_{t+1} - z_t||_2^2$$
$$= \frac{1}{2}\left(1 - \left(1 - \frac{1}{t}\right)^2\right)||w_t - z_t||_2^2 \leq \frac{1}{t}||w_t - z_t||_2^2$$

If we assume $||z_t|| \leq L$, then $||w_t - z_t||_2^2 \leq 4L^2$ and

$$\sum_{t=1}^{T}(f_t(w_t) - f_t(w_{t+1})) \leq 4L^2 \sum_{t=1}^{T}\frac{1}{t} \leq 4L^2(\ln(T) + 1)$$

Consider $1d$ linear loss $f_t(w) = z_t \cdot w$ with $\mathcal{S} = [-1, 1]$.

Consider the following inputs: $z_1 = -0.5$, for even t $z_t = 1$ and for odd $t > 1$ we have $z_t = -1$.

At all even t the FTL will return $w_t = 1$ and suffer loss 1, and at odd steps $w_t = -1$ and the loss is again 1. The total loss is T, while $u = 0$ has zero loss.

The reason the linear loss fails (unlike the quadratic), is that it is not stable.

To stabilize it we will add regularization.

Algorithm `Follow-The-Regularized-Leader (FoReL)`

for t=1,...,T **do**
predict $w_t = \arg\min_{R(w)+w\in\mathcal{S}} \sum_{i=1}^{t-1} f_t(w)$
end for

The FoRel has this similar regret bound

Theorem 2.2

Let $w_1, ..., w_T, ...$ be the outputs of the FoRel. For all $u \in \mathcal{S}$ we have

$$Regret_T(u) \leq R(u) - R(w_1) + \sum_{t=1}^{T} (f_t(w_t) - f_t(w_{t+1}))$$

Proof is the same to FTL, just adding $f_0 = R$.

Lets return to the linear example: Assume $\mathcal{S} = \mathbb{R}^d$, $U = \{u : ||u|| \leq B\}$ and $f_t(w) = \langle w, z_t \rangle$. Further assume $\frac{1}{T}\sum_{t=1}^{T} ||Z_t||_2^2 \leq L^2$.

We pick ℓ_2-regularizer $R(w) = \frac{1}{2\eta}||w||_2^2$. We can see that the FoReL returns $w_{t+1} = -\eta \sum_{t=1}^{T} z_t = w_t - \eta z_t$.

$$Regret_T(u) \leq R(u) - R(0) + \sum_{t=1}^{T} (f_t(w_t) - f_t(w_{t+1}))$$

$$\leq \frac{1}{2\eta}||u||_2^2 + \sum_{t=1}^{T} \langle w_t - w_{t+1}, z_t \rangle = \frac{1}{2\eta}||u||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 \leq \frac{B^2}{2\eta} + \eta L^2$$

Setting $\eta = \frac{B}{L\sqrt{2T}}$, we get that the bound is bounded by $BL\sqrt{2T}$.

We showed how FoRel has a square root regret bound for linear loss.

We will now generalize for any convex $f_t(w)$.

For convex differential functions $f(x)$ we have
$f(u) \geq f(w) + \langle \nabla f(w), u - w \rangle$.

For general convex functions, there exits vectors z such that
$f(u) \geq f(w) + \langle z, u - w \rangle$ called sub-gradients, and the set of all sub-gradients at w is marked $\partial f(w)$.

For our convex loss function $f_t(w)$ we have $f_t(w_t) - f_t(u) \leq \langle z_t, w_t - u \rangle$ when $z_t \in \partial f(w_t)$.

This means that

$$\sum_{t=1}^{T} (f_t(w_t) - f_t(u)) \leq \sum_{t=1}^{T} \langle z_t, w_t - u \rangle$$

Trying to minimize the r.h.s as linear loss functions with FoReL we get sub-gradient descent $w_{t+1} = w_t - \eta z_t$.

The previous analysis tells us that the regret is bounded by

$$Regret_T(u) = \sum_{t=1}^{T} (f_t(w_t) - f_t(u)) \leq \frac{1}{2\eta} ||u||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2$$

We now show how to analyse some classical online learning algorithms.

We consider linear predictors with 0-1 loss. to convexify the problem we will use surrogate loss.

We wish to find convex $f_t(w)$ such that $f_t(w_t) \geq \ell(w_t, (x_t, y_t))$.

If we predict correctly we set $f_t(w) = 0$, otherwise we set $f_t(w) = [1 - y_i \langle x_i, w \rangle]_+$.

If we use the FoReL algorithm with f_t, then when we are correct $w_{t+1} = w_t$, otherwise $w_{t+1} = w_t - \eta z_t = w_t + \eta y_i x_i$

Notice that the predictions do not depend on η. Setting $\eta = 1$ we get the perceptron algorithm.

Using the previous OCO work, we know the regret of the perceptron is

$$\sum_{t=1}^{T} f_t(w_t) \leq \sum_{t=1}^{T} f_t(u) + \frac{1}{2\eta}||u||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 \tag{4}$$

Also if the perceptron makes M mistakes then $M \leq \sum_{t=1}^{T} f_t(w_t)$.

Theorem 3.1

Suppose that the perceptron algorithm runs on a sequance $(x_1, y_1), ..., (x_T, y_T)$ *and let* $R = \max_t ||x_t||$. *Let* $\mathcal{M}$ *be the rounds on which the perceptron errs and let* $f_t(w) = \mathbb{1}[t \in \mathcal{M}][1 - t_t \langle x_t, w \rangle]_+$. *Then for any* u,

$$|\mathcal{M}| \leq \sum_t f_t(u) + R||u|| \sqrt{\sum_t f_t(u)} + R^2 ||u||^2 \tag{5}$$

In particular if there exists a u *such that* $y_t \langle x_t, u \rangle \geq 1$ *then* $|\mathcal{M}| \leq R^2 ||u||^2$.

Proof - we know

$$|\mathcal{M}| \leq \sum_{t=1}^{T} f_t(u) + \frac{1}{2\eta}||u||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 \tag{6}$$

We know $||z_t||^2 \leq R^2$, then (defining $M = |\mathcal{M}|$)

$$M \leq \sum_{t=1}^{T} f_t(u) + \frac{1}{2\eta}||u||_2^2 + \eta R^2 M \tag{7}$$

Setting $\eta = ||u||/R\sqrt{M}$ we get

$$M \leq \sum_{t=1}^{T} f_t(u) + R||u||\sqrt{M} \tag{8}$$