Online Learning

9.520 Class 12, 20 March 2006

Andrea Caponnetto, Sanmay Das

About this class

Goal To introduce the general setting of online learning.

To describe an online version of the RLS algorithm and analyze its performance.

To discuss convergence results of the classical Perceptron algorithm.

To introduce the "experts" framework and prove mistake bounds in that framework.

To show the relationship between online learning and the theory of learning in games.

What is online learning?

Sample data are arranged in a sequence.

Each time we get a new input, the algorithm tries to predict the corresponding output.

As the number of seen samples increases, hopefully the predictions improve.

Assets

1. does not require storing all data samples

2. typically fast algorithms

3. it is possible to give formal guarantees **not** assuming probabilistic hypotheses (*mistakes bounds*)

but...

- performance can be worse than best **batch** algorithms

- **generalization bounds** always require some assumption on the generation of sample data

Online setting

Sequence of sample data $z_1, z_2, \ldots, z_n$.

Each sample is an input-output couple $z_i = (x_i, y_i)$.

$x_i \in X \subset \mathbb{R}^d$, $y_i \in Y \subset \mathbb{R}$.

In the *classification* case $Y = \{+1, -1\}$, in the *regression* case $Y = [-M, M]$.

Loss function $V : \mathbb{R} \times Y \rightarrow \mathbb{R}_+$ (e.g. $\mathcal{E}(w, y) = \Theta(-yw)$ and $V(w, y) = |1 - yw|_+$).

Estimators $f_i : X \rightarrow Y$ constructed using the first i data samples.

Online setting (cont.)

- initialization f_0
- for $i = 1, 2, \ldots, n$
  - receive x_i
  - predict $f_{i-1}(x_i)$
  - receive y_i
  - update $(f_{i-1}, z_i) \to f_i$

Note: storing efficiently f_{i-1} may require much less memory than storing all previous samples $z_1, z_2, \ldots, z_{i-1}$.

Goals

Batch learning: reducing *expected loss*

$$I[f_n] = \mathbb{E}_z V(f_n(x), y)$$

Online learning: reducing *cumulative loss*

$$\sum_{i=1}^{n} V(f_{i-1}(x_i), y_i)$$

Online implementation of RLS *

update rule: For some choice of the sequences of positive parameters γ_i and λ_i,

$$f_i = f_{i-1} - \gamma_i \left((f_{i-1}(x_i) - y_i) K_{x_i} + \lambda_i f_{i-1} \right),$$

where $K : X \times X \to \mathbb{R}$ is a pd kernel and for every $x \in X$, $K_x(x') = K(x, x')$.

Note: this rule has a simple justification assuming that the sample points (x_i, y_i) are i.i.d. from a probability distribution ρ.

*Online learning algorithms. Smale, Yao. 05

Interpretation of online RLS

For sake of simplicity, let us set $\lambda_t = \lambda > 0$.

We would like to estimate the ideal regularized least-squares estimator f_ρ^λ

$$f_\rho^\lambda = \arg \min_{f \in \mathcal{H}_K} \int_{X \times Y} (f(x) - y)^2 d\rho(z) + \|f\|_K^2.$$

From the definition above it can be showed that f_ρ^λ also satisfies

$$\mathbb{E}_{z \sim \rho}[(f_\rho^\lambda(x) - y)K_x + \lambda f_\rho^\lambda] = 0,$$

therefore, f_ρ^λ is also the equilibrium point of the averaged online update equation

$$f_i = f_{i-1} - \gamma_i \mathbb{E}_{z_i \sim \rho}[(f_{i-1}(x_i) - y_i)K_{x_i} + \lambda f_{i-1}].$$

Generalization bound for online algorithm *

Theorem: Let f_ρ be the minimizer of the expected squared loss $I[f]$ (i.e. the regression function). Assume $K(x,x) \leq \kappa$ for some positive constant κ, and $L_K^{-r} f_\rho \in L^2(X, \rho_X)$ for some $r \in [1/2, 1]$. Then letting $\gamma_i = c_1 i^{-\frac{2r}{2r+1}}$ and $\lambda_i = c_2 i^{-\frac{1}{2r+1}}$ for some constants c_1 and c_2, with probability greater than $1-\delta$, for all $i \in \mathbb{N}$ it holds

$$I[f_i] \leq I[f_\rho] + C i^{-\frac{2r}{2r+1}},$$

where C depends on M, κ, r, $\|L_K^{-r} f_\rho\|_\rho$ and δ.

Note: the rates of convergence $O\left(i^{-\frac{2r}{2r+1}}\right)$ are the best theoretically attainable under these assumptions.

*Online learning as stochastic approximations of regularization paths. Tarres, Yao. 05

The Perceptron Algorithm

We consider the classification problem: $Y = \{-1, +1\}$.

We deal with linear estimators $f_i(x) = \omega_i \cdot x$, with $\omega_i \in \mathbb{R}^d$.

The 0-1 loss $\mathcal{E}(f_i(x), y) = \Theta(-y(\omega_i \cdot x))$ is the natural choice in the classification context. We will also consider the more tractable hinge-loss

$$V(f_i(x), y) = |1 - y(\omega_i \cdot x)|_+.$$

Update rule:

If $\mathcal{E}_i = \mathcal{E}(f_{i-1}(x_i), y_i) = 0$ then $\omega_i = \omega_{i-1}$, otherwise

$$\omega_i = \omega_{i-1} + y_i x_i$$

The Perceptron Algorithm (cont.)

Passive-Aggressive strategy of the update rule.

If f_{i-1} classifies **correctly** x_i, don't move.

If f_{i-1} classifies **incorrectly**, try to increase the margin $y_i(\omega \cdot x_i)$. In fact,

$$y_i(\omega_i \cdot x_i) = y_i(\omega_{i-1} \cdot x_i) + y_i^2\|x_i\|^2 > y_i(\omega_{i-1} \cdot x_i)$$

Perceptron Convergence Theorem *

Theorem: If the samples $z_1, \ldots, z_n$ are *linearly separable*, then presenting them cyclically to the Perceptron algorithm, the sequence of weight vectors ω_i will eventually converge.

We will proof a more general result encompassing both the *separable* and the *inseparable* cases

*Pattern Classification. Duda, Hart, Stork, 01

Mistakes' Bound *

Theorem: Assume $\|x_i\| \leq R$ for every $i = 1, 2, \ldots, n$. Then for every $u \in \mathbb{R}^d$

$$\mathcal{M} \leq \left(R\|u\| + \sqrt{\sum_{i=1}^{n} \widehat{V}_i^2} \right)^2,$$

where $\widehat{V}_i = V(u \cdot x_i, y_i)$ and $\mathcal{M}$ is the total number of mistakes: $\mathcal{M} = \sum_{i=1}^{n} \mathcal{E}_i = \sum_{i=1}^{n} \mathcal{E}(f_{i-1}(x_i), y_i)$.

*Online Passive-Aggressive Algorithms. Crammer et al, 03

Mistakes' Bound (cont.)

- the boundedness conditions $\|x_i\| \leq R$ is necessary.

- in the *separable* case, there exists u^* inducing margins $y_i(u^* \cdot x_i) \geq 1$, and therefore null "batch" loss over sample points. The Mistakes' Bound becomes

$$\mathcal{M} \leq R^2 \|u^*\|^2.$$

- in the *inseparable* case, we can let u be the best possible linear separator. The bound compares the online performance with the best batch performance over a given class of competitors.

Proof

The terms $\omega_i \cdot u$ increase as i increases

1. If $\mathcal{E}_i = 0$ then $\omega_i \cdot u = \omega_{i-1} \cdot u$

2. If $\mathcal{E}_i = 1$, since $\widehat{V}_i = |1 - y_i(x_i \cdot u)|_+$,

$$\omega_i \cdot u = \omega_{i-1} \cdot u + y_i(x_i \cdot u) \geq \omega_{i-1} \cdot u + 1 - \widehat{V}_i.$$

3. Hence, in both cases $\omega_i \cdot u \geq \omega_{i-1} \cdot u + (1 - \widehat{V}_i)\mathcal{E}_i$

4. Summing up, $\omega_n \cdot u \geq \mathcal{M} - \sum_{i=1}^n \widehat{V}_i \mathcal{E}_i$.

Proof (cont.)

The terms $\|\omega_i\|$ do not increase too quickly

1. If $\mathcal{E}_i = 0$ then $\|\omega_i\|^2 = \|\omega_{i-1}\|^2$

2. If $\mathcal{E}_i = 1$, since $y_i(\omega_{i-1} \cdot x_i) \leq 0$,

$$\begin{aligned} \|\omega_i\|^2 &= (\omega_{i-1} + y_i x_i) \cdot (\omega_{i-1} + y_i x_i) \\ &= \|\omega_{i-1}\|^2 + \|x_i\|^2 + 2y_i(\omega_{i-1} \cdot x_i) \leq \|\omega_{i-1}\|^2 + R^2. \end{aligned}$$

3. Summing up, $\|\omega_n\|^2 = \mathcal{M}R^2$.

Proof (cont.)

Using the estimates for $\omega_n \cdot u$ and $\|\omega_n\|^2$, and applying Cauchy-Schwartz inequality

1. By C-S, $\omega_n \cdot u \leq \|\omega_n\|\|u\|$, hence

$$\mathcal{M} - \sum_{i=1}^{n} \hat{V}_i \mathcal{E}_i \leq \omega_n \cdot u \leq \|\omega_n\|\|u\| \leq \sqrt{\mathcal{M}} R \|u\|$$

2. Finally, by C-S, $\sum_{i=1}^{n} \hat{V}_i \mathcal{E}_i \leq \sqrt{\sum_{i=1}^{n} \hat{V}_i^2} \sqrt{\sum_{i=1}^{n} \mathcal{E}_i^2}$, hence

$$\sqrt{\mathcal{M}} - \sqrt{\sum_{i=1}^{n} \hat{V}_i^2} \leq R\|u\|.$$

The Experts Framework

We will focus on the classification case.

Suppose we have a pool of prediction strategies, called experts. Denote by $E = \{E_1, \ldots, E_n\}$.

Each expert predicts y_i based on x_i.

We want to combine these experts to produce a single *master algorithm* for classification and prove bounds on how much worse it is than the *best* expert.

The Halving Algorithm[*]

Suppose all the experts are functions (their predictions for a point in the space do not change over time) and at least one of them is *consistent* with the data.

At each step, predict what the majority of experts that have not made a mistake so far would predict.

Note that all inconsistent experts get thrown away!

Maximum of $\log_2(|E|)$ errors.

But what if there is no consistent function in the pool? (Noise in the data, limited pool, etc.)

[*]Barzdin and Freivald, *On the prediction of general recursive functions*, 1972, Littlestone and Warmuth, *The Weighted Majority Algorithm*, 1994

The Weighted Majority Algorithm*

Associate a weight w_i with every expert. Initialize all weights to 1.

At example t:

$$q_{-1} = \sum_{i=1}^{|E|} w_i I[E_i \text{ predicted } y_t = -1]$$

$$q_1 = \sum_{i=1}^{|E|} w_i I[E_i \text{ predicted } y_t = 1]$$

Predict $y_t = 1$ if $q_1 > q_{-1}$, else predict $y_t = -1$

If the prediction is wrong, multiply the weights of each expert that made a wrong prediction by $0 \leq \beta < 1$.

Note that for $\beta = 0$ we get the halving algorithm.

*Littlestone and Warmuth, 1994

Mistake Bound for WM

For some example t let $W_t = \sum_{i=1}^{|E|} w_i = q_{-1} + q_1$

Then when a mistake occurs $W_{t+1} \leq uW_t$ where $u < 1$

Therefore $W_0 u^m \geq W_n$

Or $m \leq \frac{\log(W_0/W_n)}{\log(1/u)}$

Then $m \leq \frac{\log(W_0/W_n)}{\log(2/(1+\beta))}$ (setting $u = \frac{1+\beta}{2}$)

Mistake Bound for WM (contd.)

Why? Because when a mistake is made, the ratio of total weight after the trial to total weight before the trial is at most $(1+\beta)/2$.

W.L.o.G. assume WM predicted -1 and the true outcome was $+1$. Then new weight after trial is:

$$\beta q_{-1} + q_1 \leq \beta q_{-1} + q_1 + \tfrac{1-\beta}{2}(q_{-1} - q_1) = \tfrac{1+\beta}{2}(q_{-1} + q_1.$$

The main theorem (Littlestone & Warmuth):
Assume m_i is the number of mistakes made by the ith expert on a sequence of n instances and that $|E| = k$. Then the WM algorithm makes at most the following number of mistakes:

$$\frac{\log(k) + m_i \log(1/\beta)}{\log(2/(1+\beta))}$$

Big fact: Ignoring leading constants, the number of errors of the pooled predictor is bounded by the sum of the number of errors of the best expert in the pool and the log of the number of experts!

Finishing the Proof

$W_0 = k$ and $W_n \geq \beta^{m_i}$

$\log(W_0/W_n) = \log(W_0) - \log(W_n)$

$\log(W_n) > m_i \log\beta$, so $-\log(W_n) < m_i \log(1/\beta)$

Therefore $\log(W_0) - \log(W_n) < \log k + m_i \log(1/\beta)$

A Whirlwind Tour of Game Theory

Players choose actions, receive rewards based on their own actions and those of the other players.

A **strategy** is a specification for how to play the game for a player. A **pure strategy** defines, for every possible choice a player could make, which action the player picks. A **mixed strategy** is a probability distribution over strategies.

A **Nash equilibrium** is a profile of strategies for all players such that each player's strategy is an optimal response to the other players' strategies. Formally, a mixed-strategy profile σ_*^i is a Nash equilibrium if for all players i:

$$u^i(\sigma_*^i, \sigma_*^{-i}) \geq u^i(s^i, \sigma_*^{-i}) \forall s^i \in S^i$$

Some Games: Prisoners' Dilemma

	Cooperate	Defect
Cooperate	$+3, +3$	$0, +5$
Defect	$+5, 0$	$+1, +1$

Nash equilibrium: Both players defect!

Some Games: Matching Pennies

	H	T
H	$+1, -1$	$-1, +1$
T	$-1, +1$	$+1, -1$

Nash equilibrium: Both players randomize half and half between actions.

Learning in Games*

Suppose I don't know what payoffs my opponent will receive.

I can try to learn her actions when we play repeatedly (consider 2-player games for simplicity).

Fictitious play in two player games. Assumes stationarity of opponent's strategy, and that players do not attempt to influence each others' future play. Learn *weight functions*

$$\kappa_t^i(s^{-i}) = \kappa_{t-1}^i(s^{-i}) + \begin{cases} 1 & \text{if } s_{t-1}^{-i} = s^{-i} \\ 0 & \text{otherwise} \end{cases}$$

*Fudenberg & Levine, *The Theory of Learning in Games*, 1998

Calculate probabilities of the other player playing various moves as:

$$\gamma_t^i(s^{-i}) = \frac{\kappa_t^i(s^{-i})}{\sum_{\tilde{s}^{-i} \in S^{-i}} \kappa_t^i(\tilde{s}^{-i})}$$

Then choose the best response action.

Fictitious Play (contd.)

If fictitious play converges, it converges to a Nash equilibrium.

If the two players ever play a (strict) NE at time t, they will play it thereafter. (Proofs omitted)

If *empirical marginal distributions* converge, they converge to NE. But this doesn't mean that play is similar!

t	Player1 Action	Player2 Action	κ_T^1	κ_T^2
1	T	T	$(1.5, 3)$	$(2, 2.5)$
2	T	H	$(2.5, 3)$	$(2, 3.5)$
3	T	H	$(3.5, 3)$	$(2, 4.5)$
4	H	H	$(4.5, 3)$	$(3, 4.5)$
5	H	H	$(5.5, 3)$	$(4, 4.5)$
6	H	H	$(6.5, 3)$	$(5, 4.5)$
7	H	T	$(6.5, 4)$	$(6, 4.5)$

Cycling of actions in fictitious play in the matching pennies game

Universal Consistency

Persistent miscoordination: Players start with weights of $(1, \sqrt{2})$

	A	B
A	$0, 0$	$1, 1$
B	$1, 1$	$0, 0$

A rule ρ^i is said to be ϵ-**universally consistent** if for any ρ^{-i}

$$\lim_{T \to \infty} \sup \max_{\sigma^i} u^i(\sigma^i, \gamma^i_t) - \frac{1}{T} \sum_t u^i(\rho^i_t(h_{t-1})) \leq \epsilon$$

almost surely under the distribution generated by (ρ^i, ρ^{-i}), where h_{t-1} is the history up to time $t-1$, available for the decision-making algorithm at time t.

Back to Experts

Bayesian learning cannot give good payoff guarantees.

Define *universal expertise* analogously to universal consistency, and bound regret (lost utility) with respect to the best expert, which is a strategy.

The best response function is derived by solving the optimization problem

$$\max_{\mathcal{I}^i} \mathcal{I}^i \vec{u}_t^i + \lambda v^i(\mathcal{I}^i)$$

$\vec{u}_t^i$ is the vector of average payoffs player i would receive by using each of the experts

$\mathcal{I}^i$ is a probability distribution over experts

λ is a small positive number.

Under technical conditions on v, satisfied by the negative entropy:

$$-\sum_{s} \sigma(s) \log \sigma(s)$$

we retrieve the exponential weighting scheme, and for every ϵ there is a λ such that our procedure is ϵ-universally expert.