Online Learning: Expert Setting

CS6780 – Advanced Machine Learning
Spring 2019

Thorsten Joachims
Cornell University

Reading: Shalev-Shwartz/Ben-David, 287-297
(at http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/copy.html)

Online Classification Model

- Setting
  - Classification
  - Hypothesis space H with h: X→Y
  - Measure misclassifications (i.e. zero/one loss)
- Interaction Model
  - Initialize hypothesis $h \in H$
  - FOR t from 1 to T
    - Receive x_{t}
    - Make prediction $\widehat{y_{\mathrm{t}}} = h(x_{\mathrm{t}})$
    - Receive true label y_{t}
    - Record if prediction was correct (e.g., $\widehat{y_{\mathrm{t}}} = y_{\mathrm{t}}$)
    - Update h

(Online) Perceptron Algorithm

- Input: $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n))$, $\vec{x}_i \in \Re^N$, $y_i \in \{-1, 1\}$
- Algorithm:
  - $\vec{w}_0 = \vec{0}$, $k = 0$
  - FOR i=1 TO n
    - * IF $y_i(\vec{w}_k \cdot \vec{x}_i) \leq 0$ ### makes mistake
      - · $\vec{w}_{k+1} = \vec{w}_k + y_i\vec{x}_i$
      - · $k = k + 1$
    - * ENDIF
  - ENDFOR
- Output: $\vec{w}_k$

Perceptron Mistake Bound

Theorem: For any sequence of training examples $S = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$ with

$$R = \max \|\vec{x}_i\| ,$$

if there exists a weight vector $\vec{w}_{opt}$ with $\|\vec{w}_{opt}\| = 1$ and

$$y_i \left(\vec{w}_{opt} \cdot \vec{x}_i\right) \geq \delta$$

for all $1 \leq i \leq n$, then the Perceptron makes at most

$$\frac{R^2}{\delta^2}$$

errors.

Expert Learning Model

- Setting
  - N experts named $\mathrm{H} = \{h_1, \ldots, h_N\}$
  - Each expert h_i takes an action $y = h_i(x_t)$ in each round t and incurs loss $\Delta_{t,i}$
  - Algorithm can select which expert's action to follow in each round
- Interaction Model
  - FOR t from 1 to T
    - Algorithm selects expert h_{i_t} according to strategy A_{w_t} and follows its action y
    - Experts incur losses $\Delta_{t,1}$ … $\Delta_{t,N}$
    - Algorithm incurs loss Δ_{t,i_t}
    - Algorithm updates w_t to w_{t+1} based on $\Delta_{t,1}$ … $\Delta_{t,N}$

Halving Algorithm

- Setting
  - N experts named $\mathrm{H} = \{h_1, \dots, h_N\}$
  - Binary actions $y = \{+1, -1\}$ given input x, zero/one loss
  - Perfect expert exists in H
- Algorithm
  - $VS_1 = H$
  - FOR t = 1 TO T
    - Predict the same y as majority of $h_i \in VS_t$
    - $VS_{t+1} = VS_t$ minus those $h_i \in VS_t$ that were wrong
- Mistake Bound
  - How many mistakes can the Halving algorithm make before predicting perfectly?

Regret

- Idea
  - N experts named $\mathrm{H} = \{h_1, \dots, h_N\}$
  - Compare performance of A to best expert i^* in hindsight.
- Regret
  - Overall loss of best expert i^* in hindsight is

$$\Delta_T^* = \min_{i^* \in [1..N]} \sum_{t=1}^{T} \Delta_{t,i^*}$$

  - Loss of algorithm A at time t is

$$\Delta_{t,i}$$

    for algorithm that picks recommendation of expert $i = A(w_t)$ at time t.
  - Regret is difference between loss of algorithm and best fixed expert in hindsight

$$Regret(T) = \sum_{t=1}^{T} \Delta_{t,A(w_t)} - \min_{i^* \in [1..N]} \sum_{t=1}^{T} \Delta_{t,i^*}$$

Weighted Majority Algorithm (Deterministic)

- Setting
  - N experts named $\mathrm{H} = \{h_1, \dots, h_N\}$
  - Binary actions $y = \{+1, -1\}$ given input x, zero/one loss
  - There may be no expert in H that acts perfectly
- Algorithm
  - Initialize $\mathrm{w}_1 = (1,1, \dots, 1)$
  - FOR t = 1 TO T
    - Predict the same y as majority of $h_i \in H$, each weighted by $w_{t,i}$
    - FOREACH $h_i \in H$
      - IF h_i incorrect THEN $w_{t+1,i} = w_{t,i} * \beta$
        ELSE $w_{t+1,i} = w_{t,i}$
- Regret Bound
  - How close is the number of mistakes the Weighted Majority Algorithm makes to the number of mistakes of the best expert in hindsight?

Exponentiated Gradient Algorithm for Expert Setting (EG)

- Setting
  - N experts named $\mathrm{H} = \{h_1, \dots, h_N\}$
  - Any actions, any positive and bounded loss
  - There may be no expert in H that acts perfectly
- Algorithm
  - Initialize $\widehat{w}_1 \; = \; (1, \dots, 1)$
  - FOR t from 1 to T
    - Algorithm randomly picks i_t from $P(I_t = i_t) = w_{t,i}$ where $w_{t,i} = \widehat{w}_{t,i}/Z_t$ and $Z_t = \sum_i \widehat{w}_{t,i}$
    - Experts incur losses $\Delta_{t,1} \dots \Delta_{t,N}$
    - Algorithm incurs loss Δ_{t,i_t}
    - Algorithm updates w for all experts i as
$$\forall i, \widehat{w}_{t+1,i} = \widehat{w}_{t,i} \exp\left(-\eta \Delta_{t,i}\right)$$

Expected Regret

- Idea
  - Compare performance to best expert in hindsight
- Regret
  - Overall loss of best expert i^* in hindsight is

$$\Delta_T^* = \min_{i^* \in [1..N]} \sum_{t=1}^{T} \Delta_{t,i^*}$$

  - Expected loss of algorithm $A(w_t)$ at time t is

$$E_{A(w_t)}\left[\Delta_{t,i}\right] = w_t \Delta_t$$

    for randomized algorithm that picks recommendation of expert i at time t with probability $w_{t,i}$.

  - Regret is difference between expected loss of algorithm and best fixed expert in hindsight

$$ExpectedRegret(T) = \sum_{t=1}^{T} w_t \Delta_t - \min_{i^* \in [1..N]} \sum_{t=1}^{T} \Delta_{t,i^*}$$

Regret Bound for Exponentiated Gradient Algorithm

- Theorem

  The expected regret of the exponentiated gradient algorithm in the expert setting is bounded by

$$\text{Expected}Regret(T) \leq \sqrt{2\;T\;log(|H|)}$$

  where $\Delta \in [0,1]$ and $\eta = \sqrt{2\log(|H|)\,/T}$ and $T > 2\log(|H|)$.