Statistical Learning Theory: Expert Learning

CS4780/5780 – Machine Learning
Fall 2014
Thorsten Joachims
Cornell University
Reading: Mitchell Chapter 7.5

Generalization Error Bound: Infinite H, Non-Zero Error

- Setting
  - Sample of n labeled instances S
  - Learning Algorithm L using a hypothesis space H with VCDim(H)=d
  - L returns hypothesis ĥ(S) with lowest training error
- Definition: The VC-Dimension of H is equal to the maximum number d of examples that can be split into two sets in all \( 2^d \) ways using functions from H (shattering).
- Given hypothesis space H with VCDim(H) equal to d and an i.i.d. sample S of size n, with probability (1-δ) it holds that:

\[
F_{\text{emp}}(ĥ(S)) \leq F_{\text{true}}(ĥ(\mathbb{Z}^d)) + \sqrt{\frac{\ln \left( \frac{\ln (\frac{1}{δ})}{\ln \left( \frac{n^d}{\delta} \right)} \right) + \ln \left( \frac{1}{δ} \right)}{n}}
\]

Outline

- Online learning
- Review of perceptron and mistake bound
- Expert model
  - Halving Algorithm
  - Weighted Majority Algorithm
  - Exponentiated Gradient Algorithm
- Bandit model
  - EXP3 Algorithm

Online Classification Model

- Setting
  - Classification
  - Hypothesis space H with h: X ↦ Y
  - Measure misclassifications (i.e., zero/one loss)
- Interaction Model
  - Initialize hypothesis h ∈ H
  - FOR t from 1 to T
    - Receive \( x_t \)
    - Make prediction \( y_t = h(x_t) \)
    - Receive true label \( y_t \)
    - Record if prediction was correct (e.g., \( y_t = y_t \))
    - Update h

(Online) Perceptron Algorithm

- Input: \( S = (x_1, y_1), ..., (x_n, y_n) \), \( x_i \in \mathbb{R}^d \), \( y_i \in \{-1, 1\} \)
- Algorithm:
  - \( \bar{w}_0 = 0, \bar{x} = 0 \)
  - FOR \( i = 1 \) to \( n \)
    - IF \( y_i (\bar{w}_k \cdot \bar{x}_i) < 0 \) # makes mistake
      - \( \bar{w}_{k+1} = \bar{w}_k + y_i \bar{x}_i \)
      - \( k = k + 1 \)
  - END FOR
- Output: \( \bar{w}_k \)

Perceptron Mistake Bound

Theorem: For any sequence of training examples \( S = (x_1, y_1), ..., (x_n, y_n) \) with

\[
R = \max \| \bar{x}_i \|,
\]
if there exists a weight vector \( \bar{w}_{\text{opt}} \) with \( \| \bar{w}_{\text{opt}} \| = 1 \) and

\[
y_i \left( \bar{w}_{\text{opt}} \cdot \bar{x}_i \right) \geq \delta
\]
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 $H = \{h_1, ..., h_N\}$
  - Each expert $h_i$ takes an action $y = h_i(x_t)$ in each round $t$ and incurs loss $\Delta_{i,t}$
  - Algorithm can select which expert’s action to follow in each round

- **Interaction Model**
  - FOR $t$ from 1 to $T$
    - Algorithm selects expert $h_i$ according to strategy $A_{w_t}$ and follows its action $y$
    - Experts incur losses $\Delta_{i,1} - \Delta_{i,N}$
    - Algorithm incurs loss $\Delta_{i,t}$
    - Algorithm updates $w_t$ to $w_{t+1}$ based on $\Delta_{i,1} - \Delta_{i,N}$

**Halving Algorithm**

- **Setting**
  - $N$ experts named $H = \{h_1, ..., h_N\}$
  - Binary actions $y = \{+1, -1\}$ given input $x$, zero/one loss
  - Perfect expert exists in $H$

- **Algorithm**
  - $V_{S_1} = H$
  - FOR $t = 1$ TO $T$
    - Predict the same $y$ as majority of $h_i \in V_{S_t}$
    - $V_{S_{t+1}} = V_{S_t}$ minus those $h_i \in V_{S_t}$ that were wrong

- **Mistake Bound**
  - How many mistakes can the Halving algorithm make before predicting perfectly?