CS 412 — Introduction to Machine Learning (UIC) April 29, 2025

Lecture 28

Instructor: Aadirupa Saha *Scribe(s): Harsh Kothari*

Overview

In the last lecture, we covered the following main topics:

1. Gaussian Processes

This lecture focuses on:

1. Online Learning
2. Halving Algorithm
3. Weighted Majority Algorithm

1 Online Learning

1.1 Online Learning Framework

Motivation: Unlike batch learning, where all training data is available upfront, **online learning** handles data sequentially. The model is updated in real-time as new data arrives.
Setting:

- The learner does not have access to the full dataset in advance.
- Learning proceeds over T rounds (or timesteps).

Online Learning Protocol:
For $t = 1, 2, \ldots, T$:

1. **Receive Input:** Receive an instance
$$\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$$
2. **Predict:** Predict a label
$$\hat{y}_t \in \mathcal{Y}$$
using the current predictor
$$f_t : \mathcal{X} \to \mathcal{Y}$$
3. **Receive Feedback:** The true label $y_t \in \mathcal{Y}$ is revealed.
4. **Incur Loss:** Compute the loss using a suitable loss function ℓ:
$$\ell(y_t, \hat{y}_t)$$
5. **Update Predictor:** Update the model based on the feedback:
$$f_{t+1} \leftarrow \text{Update}(f_t, \mathbf{x}_t, y_t)$$

2 Halving Algorithm

2.1 Problem Setup: Halving Algorithm

Task: Classification

We are in a binary classification setting where the goal is to learn a mapping:

$$\mathcal{X} \to \{0, 1\}$$

At each round $t = 1, 2, \ldots, T$:

- The learner receives an input $\mathbf{x}_t \in \mathcal{X}$
- The learner predicts a label $\hat{y}_t \in \{0, 1\}$
- The true label $y_t \in \{0, 1\}$ is revealed

Hypothesis Class

The learner has access to a finite hypothesis class:

$$\mathcal{H} = \{h_1, h_2, \ldots, h_N\}$$

where each hypothesis $h_i \in \mathcal{H}$ is a function:

$$h_i : \mathcal{X} \to \{0, 1\}$$

That is, each hypothesis maps any input instance to a binary label.

Objective

The goal is to minimize the cumulative prediction error over T rounds:

$$\min \sum_{t=1}^{T} \ell(y_t, \hat{y}_t)$$

where:

- y_t is the true label
- $\hat{y}_t$ is the learner's prediction
- $\ell(y_t, \hat{y}_t)$ is the 0-1 loss:

$$\ell(y_t, \hat{y}_t) = \begin{cases} 0 & \text{if } y_t = \hat{y}_t \\ 1 & \text{if } y_t \neq \hat{y}_t \end{cases}$$

Realizability Assumption

We assume the hypothesis class $\mathcal{H}$ is **realizable**, i.e.,

$$\exists h^* \in \mathcal{H} \text{ such that } \forall t \in \{1, \ldots, T\}, \quad h^*(\mathbf{x}_t) = y_t$$

This means there exists a perfect hypothesis in $\mathcal{H}$ that makes zero mistakes over the entire sequence.

2.2 Halving Algorithm

Initialization

Start with the full hypothesis class:

$$H_1 = \mathcal{H}$$

For each round $t = 1, 2, \ldots, T$:

1. **Receive input:** $\mathbf{x}_t \in \mathcal{X}$
2. **Predict label:**
$$\hat{y}_t = \begin{cases} 1 & \text{if } \sum_{h \in H_t} \mathbb{I}[h(\mathbf{x}_t) = 1] \geq \frac{|H_t|}{2} \\ 0 & \text{otherwise} \end{cases}$$
(Majority vote over hypotheses)
3. **Receive true label:** $y_t \in \{0, 1\}$
4. **Update version space:**
$$H_{t+1} \leftarrow \{h \in H_t \mid h(\mathbf{x}_t) = y_t\}$$

Mistake Bound

At each mistake, at least half of the hypotheses are eliminated. Therefore, the total number of mistakes is at most:

$$\log_2 |\mathcal{H}|$$

Assumptions

- The hypothesis class $\mathcal{H}$ is finite: $|\mathcal{H}| = N$
- Realizability holds: there exists $h^* \in \mathcal{H}$ such that for all t,
$$h^*(\mathbf{x}_t) = y_t$$
- At each mistake, the algorithm eliminates all hypotheses that disagree with the true label.

Key Observation

When a mistake is made at time t, the algorithm predicts the majority label, so strictly more than half of the hypotheses in H_t were wrong. Therefore, the size of the hypothesis set is at most halved:

$$|H_{t+1}| \leq \frac{1}{2}|H_t|$$

Derivation

Let M be the total number of mistakes. Then after M mistakes:

$$|H_{M+1}| \leq \frac{N}{2^M}$$

But by the realizability assumption, the correct hypothesis h^* is never eliminated, so:

$$|H_{M+1}| \geq 1$$

Combining the inequalities:

$$\frac{N}{2^M} \geq 1 \quad \Rightarrow \quad 2^M \leq N \quad \Rightarrow \quad M \leq \log_2 N$$

Conclusion

The total number of mistakes made by the Halving Algorithm is at most:

$$\boxed{M \leq \log_2 |\mathcal{H}|}$$

3 Weighted Majority Algorithm

Setting

We consider the same online binary classification setting as the Halving Algorithm:

$$\mathcal{X} \to \{0, 1\}$$

At each round $t = 1, 2, \ldots, T$, the learner receives an input $\mathbf{x}_t \in \mathcal{X}$ and must predict a label $\hat{y}_t \in \{0, 1\}$. The learner has access to a finite hypothesis class:

$$\mathcal{H} = \{h_1, h_2, \ldots, h_N\}, \quad h_i : \mathcal{X} \to \{0, 1\}$$

Note: The realizability assumption does *not* hold here. There may be no perfect hypothesis in $\mathcal{H}$.

Initialization

- Assign initial weights: $w_1(i) = 1 \quad \forall i \in \{1, \ldots, N\}$
- Choose a learning rate $\varepsilon \in [0, 1]$

For each round $t = 1, 2, \ldots, T$:

1. **Receive input:** $\mathbf{x}_t \in \mathcal{X}$
2. **Compute normalized weights:**

$$p_t(i) = \frac{w_t(i)}{\sum_{j=1}^{N} w_t(j)}$$

3. **Make prediction (Weighted Majority):**

$$\hat{y}_t = \text{round}\left(\sum_{i=1}^{N} p_t(i) \cdot h_i(\mathbf{x}_t)\right)$$

   where

$$\text{round}(x) = \begin{cases} 1 & \text{if } x \geq 0.5 \\ 0 & \text{if } x < 0.5 \end{cases}$$

4. **Receive true label:** $y_t \in \{0, 1\}$

5. **Update weights:** For each $i \in \{1, \dots, N\}$,

$$w_{t+1}(i) = \begin{cases} w_t(i) \cdot (1 - \varepsilon) & \text{if } h_i(\mathbf{x}_t) \neq y_t \\ w_t(i) & \text{otherwise} \end{cases}$$

4 Mistake Bound of the Weighted Majority Algorithm

Theorem 2.6

For all experts $i \in \{1, \dots, N\}$, the number of mistakes made by the Weighted Majority Algorithm (WMA) up to round T is bounded by:

$$M_T(\text{WMA}) \leq \frac{2 \log N}{\varepsilon} + 2(1 + \varepsilon) M_T(\text{expert } i)$$

where:

- $M_T(\text{WMA})$: total mistakes made by the algorithm
- $M_T(\text{expert } i)$: total mistakes made by expert i
- N: number of experts
- $\varepsilon \in [0, 1]$: learning rate

Definitions and Setup

Let:

$$\Phi_t = \sum_{i=1}^{N} w_t(i)$$

be the total weight of all experts at time t, where $w_t(i)$ is the weight of expert i.
Initial total weight:

$$\Phi_1 = N$$

Weight of expert i at time T:

$$w_T(i) = (1 - \varepsilon)^{M_T(\text{expert } i)} \Rightarrow \Phi_T \geq w_T(i)$$

Key Lemma (Lemma 2.7)

If the algorithm makes a mistake at time t, then:

$$\Phi_{t+1} \leq \Phi_t \left(1 - \frac{\varepsilon}{2}\right)$$

Proof: Let $S = \sum_{i \in \text{wrong}} w_t(i)$ be the total weight of experts who predicted incorrectly at round t. Since the algorithm made a mistake, the majority (by weight) must have been wrong:

$$S > \frac{1}{2}\Phi_t$$

Then the updated total weight is:

$$\Phi_{t+1} = \sum_{\text{correct}} w_t(i) + \sum_{\text{wrong}} w_t(i)(1-\varepsilon) = \Phi_t - \varepsilon S$$

Thus:

$$\Phi_{t+1} < \Phi_t - \varepsilon \cdot \frac{1}{2}\Phi_t = \Phi_t \left(1 - \frac{\varepsilon}{2}\right)$$

Bounding the Total Weight

If the algorithm makes $M_T(\text{WMA})$ mistakes, repeatedly applying Lemma 2.7:

$$\Phi_T \leq \Phi_1 \left(1 - \frac{\varepsilon}{2}\right)^{M_T(\text{WMA})} = N \left(1 - \frac{\varepsilon}{2}\right)^{M_T(\text{WMA})}$$

From earlier, we also have:

$$\Phi_T \geq w_T(i) = (1-\varepsilon)^{M_T(\text{expert } i)}$$

Combining the Bounds

$$(1-\varepsilon)^{M_T(\text{expert } i)} \leq N \left(1 - \frac{\varepsilon}{2}\right)^{M_T(\text{WMA})}$$

Take negative logarithms on both sides:

$$M_T(\text{expert } i) \cdot \log\left(\frac{1}{1-\varepsilon}\right) \geq \log N + M_T(\text{WMA}) \cdot \log\left(\frac{1}{1-\frac{\varepsilon}{2}}\right)$$

Using Approximations

Using the bounds:

$$\log\left(\frac{1}{1-\varepsilon}\right) \leq \varepsilon + \varepsilon^2$$
$$\log\left(\frac{1}{1-\frac{\varepsilon}{2}}\right) \geq \frac{\varepsilon}{2}$$

Substitute into the inequality:

$$M_T(\text{expert } i)(\varepsilon + \varepsilon^2) \geq \log N + \frac{\varepsilon}{2} M_T(\text{WMA})$$

Rearrange to solve for $M_T(\text{WMA})$:

$$M_T(\text{WMA}) \leq \frac{2\log N}{\varepsilon} + 2(1+\varepsilon) M_T(\text{expert } i)$$

Conclusion

The total number of mistakes made by the Weighted Majority Algorithm is bounded by:

$$\boxed{M_T(\text{WMA}) \leq \frac{2\log N}{\varepsilon} + 2(1+\varepsilon)M_T(\text{expert } i)}$$

Next Lecture

The next lecture will cover the following topics:
(i) Exponential Weighted Algorithm

References

[1] Cesa-Bianchi, Nicolò and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[2] Littlestone, Nick and Manfred K. Warmuth. ”The weighted majority algorithm.” *Information and Computation*, 108(2), 1994, pp. 212–261.

[3] Freund, Yoav and Robert E. Schapire. ”A Decision-Theoretic Generalization of Online Learning and an Application to Boosting.” In: *European Conference on Computational Learning Theory*, Springer, 1997, pp. 23–37.