Statistical Techniques in Robotics (16-831, S20) Lecture #04 (Monday, January 13)

PWEA (RWMA) & OLC (Perceptron)

Lecturer: Kris Kitani *Scribes: Vikas Raunak, Sang Keun Choe*

1 Review

In the last lectures, on the topic of Prediction with Expert Advice (PWEA), we had analyzed three main algorithms: (1) Greedy, (2) Halving, and (3) Randomized Greedy. In this lecture, we will introduce and analyze two more algorithms for the PWEA problem, namely: the Weighted Majority Algorithm (WMA) and the Randomized Weighted Majority Algorithm (RWMA). For a quick high level takeaway, we note that the Weighted Majority Algorithm (WMA) has a bounded regret, while the RWMA is our first no regret algorithm. This lecture, after the review section, will derive these results. In the next two subsections, we first recap the problem of prediction with expert advice and a few definitions to help us along the way in deriving those results.

1.1 Prediction with Expert Advice

We recall the characteristics of the problem of Prediction with Expert Advice [2] (an online learning problem):

1. One-Shot: The data arrives as a stream, but its sequence is not correlated.
2. Instructive: We receive a fully observable loss based on the prediction made.
3. Exhaustive: We will eventually have access to the entire state and action space.

Clearly, the key distinction from supervised learning setting arises from the fact that there is no distinct train and test stage, and input & output are given in sequence. This distinction also leads to a different kind of analysis, in terms of bounds on mistakes and regrets, as we will develop later.

1.2 Realizability

Definition 1. Realizability is the assumption that the learner has access to the perfect hypothesis.

To discuss further, we recall that under the conditions of realizability, the performance is measured in terms of the mistake bound, which represents, in retrospect, the cost for not following the perfect hypothesis. Under this assumption of realizability, previously we had seen the halving (Majority) algorithm, which maintains a set of hypothesis (the version space) consistent with past evidence, i.e. it predicts according to the majority vote of hypothesis from the version space. We had also obtained the mistake bound of the halving algorithm to be logarithmic in the number of hypotheses, since it removes at least half the hypotheses after one mistake.

However, we will now relax this realizability assumption and allow even the best expert to make a few mistakes. Under these new conditions of non-realizability, we will evaluate the performance of the algorithm in terms of the *regret bound* instead of the *mistake bound*. In such a case, we would like to design algorithms demonstrating small regrets. Accordingly, we start with providing the (mathematical) definition of regret below.

Definition 2. Regret of the learner, $R^{(T)}(H)$, is the cumulative loss for not following the best hypothesis in the hypothesis class H. Regret can be mathematically formulated as follows:

$$R^{(T)}(H) = \sum_{t=1}^{T} l(\hat{y}^{(t)}, y^{(t)}) - \min_{h \in H} \sum_{t=1}^{T} l(h(x^{(t)}), y^{(t)}) \tag{1}$$

where, $\sum_{t=1}^{T} l(\hat{y}^{(t)}, y^{(t)})$ represents the cumulative loss of the learner, and $\min \sum_{t=1}^{T} l(h(x^{(t)}), y^{(t)})$ represents the loss of best single hypothesis.

Clearly, the regret bound is a more general performance bound, of which the mistake bound is a special case. In this lecture, we will use the regret bound to characterize online algorithms and conversely, we will be interested in finding good algorithms as characterized by this regret bound. We recall that in previous lectures, we had assumed at least one perfect algorithm, which implies realizability. However, in this lecture, the first algorithm we will consider is under the relaxed assumption of non-realizability, namely, the Weighted Majority Algorithm (WMA), which has a multiplicative weighting on the experts that decays with the number of mistakes.

Definition 3. Potential Function is the sum of the weights, a non-integer scalar equivalent to the size of the version space, used in the context of Weighted Majority algorithms.

2 Summary

2.1 Weighted Majority Algorithm (WMA)

The Weighted Majority [1] is listed as Algorithm 1 below. The update equation on line 7 states that the weights are updated when the learner's prediction doesn't match the actual answer. It is important to note this simple weight update rule, which we will be making use of in our regret bound analysis.

Algorithm 1 Weighted Majority Algorithm (WMA)

1: $\mathbf{w}^{(1)} \leftarrow \{w_n^{(1)} = 1\}_{n=1}^{N}$ ▷ Weight initialization
2: $\eta \leq \frac{1}{2}$ ▷ Penalty rate initialization
3: **for** $t = 1, \cdots, T$ **do**
4: RECEIVE $(\mathbf{x}^{(t)} \in \{-1, 1\}^N)$ ▷ Receive expert predictions
5: $\hat{y}^{(t)} = \text{sign}\Big(\sum_{n=1}^{N} x_n^{(t)} \cdot w_n^{(t)}\Big) \in \{-1, 1\}$ ▷ Make learner prediction
6: RECEIVE $(y^{(t)} \in \{-1, 1\})$ ▷ Receive actual answer
7: $w_n^{(t+1)} \leftarrow w_n^{(t)}\big(1 - \eta \cdot \mathbf{1}[y^{(t)} \neq x_n^{(t)}]\big)$ ▷ Weight update
8: **end for**

Moving ahead, the key question is that if WMA makes a mistake, how much does the potential change? At a high level, we want the upper bound of the potential function. Formally, we derive the results below. But first we state a lemma, which will be useful in deriving the regret bound.

Lemma 4. *For $0 < x < 1$, following two inequalities hold:*

$$-x - x^2 \leq \log(1-x) \tag{2}$$
$$\log(1-x) \leq -x \tag{3}$$

Proof. One can easily prove the above inequalities by applying Taylor series to $\log(1-x)$.

Theorem 5. *(Mistake bound of WMA) Let $M^{(t)}$, $m_i^{(t)}$ respectively be the number of mistakes that have been made by the WMA learner and the i-th hypothesis until the time step t, and N, η be the number of experts and the penalty rate. Then, $M^{(t)}$ is upper-bounded as:*

$$M^{(t)} \leq 2(1+\eta)m_i^{(t)} + \frac{2\log N}{\eta}$$

Proof. Lets define the potential function $\Phi^{(t)}$ for WMA as follows:

$$\Phi^{(t)} = \sum_{n=1}^{N} w_n^{(t)} \tag{4}$$

where, $w_n^{(t)}$ is the weight for the expert n at time step t.

Suppose the learner made a mistake at the time step t. Since the majority rule implies at least half of weighted experts are mistaken, corresponding at least half of weights will accordingly be penalized by $(1-\eta)$, and this leads to the following inequality:

$$\Phi^{(t+1)} \leq \Phi^{(t)}\Big(\frac{1}{2} + \frac{1}{2}(1-\eta)\Big) = \Big(1 - \frac{\eta}{2}\Big)\Phi^{(t)} \tag{5}$$

By assuming all weights are initialized to 1 and recursively applying the above inequality, we can derive the upper bound of the potential function as follows:

$$\Phi^{(t+1)} \leq \Phi^{(1)}\Big(1 - \frac{\eta}{2}\Big)^{M^{(t)}} = N\Big(1 - \frac{\eta}{2}\Big)^{M^{(t)}} \tag{6}$$

From the definition of the potential function, we can also derive the lower-bound of the potential function as follows:

$$\Phi^{(t+1)} = \sum_{n=1}^{N} w_n^{(t+1)} \geq w_n^{(t+1)} \quad \text{for all } 1 \leq n \leq N \tag{7}$$

The above inequality comes from the fact that every weight $w_n^{(t+1)}$ is positive, since they are both initialized and multiplied with positive values.

Moreover, by the weight update rule of WMA, we can analytically calculate $w_n^{(t+1)}$ as follows:

$$w_n^{(t+1)} = w_n^{(1)}(1-\eta)^{m_n^{(t)}} = 1 \cdot (1-\eta)^{m_n^{(t)}} \tag{8}$$

By combining Eq. (6) $\sim$ (8), we can now derive the mathematical relation between $M^{(t)}$ and $m_n^{(t)}$:

$$(1-\eta)^{m_n^{(t)}} \leq \Phi^{(t+1)} \leq N\Big(1-\frac{\eta}{2}\Big)^{M^{(t)}} \tag{9}$$

Finally, we can compute the (upper) bound of mistake $M^{(t)}$ by combining the bounds of $\Phi^{(t+1)}$ (Eq. (9)) and Lemma 4 as follows:

$$\begin{aligned}
m_n^{(t)}\log(1-\eta) &\leq \log N + M^{(t)}\log\Big(1-\frac{\eta}{2}\Big) \quad \text{(by applying log to Eq. (9))} \\
m_n^{(t)}(-\eta-\eta^2) &\leq \log N + M^{(t)}\log\Big(1-\frac{\eta}{2}\Big) \quad (\because \text{ Lemma 4}) \\
m_n^{(t)}(-\eta-\eta^2) &\leq \log N + M^{(t)}\Big(-\frac{\eta}{2}\Big) \quad (\because \text{ Lemma 4}) \\
M^{(t)}\Big(\frac{\eta}{2}\Big) &\leq \log N + m_n^{(t)}(\eta+\eta^2) \\
M^{(t)} &\leq \frac{2\log N}{\eta} + 2m_n^{(t)}(1+\eta) \quad \forall n
\end{aligned}$$

Therefore, we have obtained the upper bound of mistakes for WMA. □

Now, let's compare the mistake bound of WMA with algorithms from the previous lectures: earlier, the upper bound always depended on the number of experts (e.g. logarithmically in the Halving (Majority) Algorithm). However, here one of the terms ($2m_n^{(t)}(1+\eta)$) is based on the number of mistakes an expert makes.

Further, recall from Section 1.2 that we conduct our analysis using the *regret bound* instead of the mistake bound under the condition of non-realizability. Before going from the mistake bound to the regret bound, we state the definition of a no-regret algorithm below.

Definition 6. The algorithm is a **no-regret** algorithm if and only if $\frac{R(h)}{T} \to 0$ as $T \to \infty$.

The above definition means that as $T \to \infty$, $\frac{R(h)}{T} \to 0$ (regret divided by T) or average regret over time goes to zero. We will now prove below that WMA shows only a bounded-regret propert, not a no-regret property.

Theorem 7. *WMA is **not** a no-regret algorithm.*

Proof. By the definition of regret and Theorem 5, we can obtain the upper bound of *regret* of WMA, $R(h_n)$, as follows:

$$R(h_n) = M^{(T)} - m_n^{(T)} \leq m_n^{(T)} + 2\eta m_n^{(T)} + \frac{2\log N}{\eta} \tag{10}$$

Since $m_n^{(T)}$ follows $O(T)$ and $2\eta m_n^{(T)} + \frac{2\log N}{\eta} > 0$, the upper bound of the *average regret* over time, $\frac{R(h_n)}{T}$, cannot be smaller than $O(1)$, which doesn't converge to 0 as $T \to \infty$. Thus, WMA is not a no-regret algorithm. □

To analyze further, the LHS term in Eq. (10) is the regret bound of the weighted majority algorithm. We want to ask what kind of regret algorithm is this? Is it bounded, growing or diverging (i.e. the number of mistakes is unbounded)? Formally, we want to ask: how does the regret bound change over time? What is the order of the number of mistakes? Clearly, we can state that the growth of the terms on the RHS is $O(T)$ (the first two terms are $O(T)$, while the last term is constant, so that the dominant complexity term is $O(T)$.), which is neither good nor bad.

Therefore, to summarize, WMA has a bounded regret, where error just grows linearly over time (on the optimistic side, the error does not explode). Further, as we will see, using any kind of deterministic prediction, we can't really get a no-regret algorithm, but just by adding some randomization, we will cut the number of mistakes in half. Actually, just by changing one single line of the algorithm, we will cut the number of mistakes in half. This brings us to the study of the Randomized Weighted Majority Algorithm (RWMA) in the next section.

2.2 Randomized Weighted Majority Algorithm (RWMA)

Now, we introduce the Randomized Weighted majority Algorithm (RWMA), listed as Algorithm 2 below. It is known by many different names, such as Exponentiated Weighted Majority, Hedge Algorithm etc. The key difference from WMA is that we create a multinomial distribution using the weights and make the learner prediction via sampling from this distribution (Line 5, 6).

Algorithm 2 Randomized Weighted Majority Algorithm (RWMA)

1: $\mathbf{w}^{(1)} \leftarrow \{w_n^{(1)} = 1\}_{n=1}^N$ ▷ Weight initialization
2: $\eta \leq \frac{1}{2}$ ▷ Penalty rate initialization
3: **for** $t = 1, \cdots, T$ **do**
4: RECEIVE $(\mathbf{x}^{(t)} \in \{-1, 1\}^N)$ ▷ Receive experts predictions
5: $I \sim \text{MULTINOMIAL}(\mathbf{w}^{(t)}/\Phi^{(t)})$, where $\Phi^{(t)} = \sum_{n=1}^N w_n^{(t)}$
6: $\hat{y}^{(t)} = h_i(\mathbf{x}^{(t)})$ ▷ Make learner prediction via sampling
7: RECEIVE $(y^{(t)} \in \{-1, 1\})$ ▷ Receive actual answer
8: $w_n^{(t+1)} \leftarrow w_n^{(t)}\big(1 - \eta \cdot \mathbf{1}[y^{(t)} \neq h_n(\mathbf{x})^{(t)}]\big)$ ▷ Weight update
9: **end for**

Now, let's derive the mistake bound for RWMA. We will find that the expected number of mistakes for RWMA is upper bounded and the mistake bound is better than a factor of 2, when compared to the WMA. But first, we state a simple lemma to help derive further results in Theorem 9.

Lemma 8. *$e^x \geq 1 + x$ for all $x \in \mathbb{R}$.*

Proof. One can easily show the above inequality by applying Taylor expansion to e^x.

Theorem 9. *(Mistake bound of RWMA) Let $M^{(t)}, m_i^{(t)}$ respectively be the number of mistakes that have been made by the RWMA learner and the i-th hypothesis until the time step t, and N, η be the number of experts and the penalty rate. Then, expected mistakes of learner $\mathbb{E}[M^{(t)}]$ is upper-bounded as:*

$$\mathbb{E}[M^{(t)}] \leq (1 + \eta) m_i^{(t)} + \frac{\log N}{\eta}$$

Proof. Lets again define the potential function $\Phi^{(t)}$ for RWMA as the sum of weights:

$$\Phi^{(t)} = \sum_{n=1}^{N} w_n^{(t)} \tag{11}$$

By the weight update rule of RWMA, $w_n^{(t+1)}$ can be analytically calculated from $w_n^{(t)}$ as follows:

$$w_n^{(t+1)} = \big(1 - \eta \cdot \mathbf{1}[y^{(t)} \neq y_n^{(t)}]\big) w_n^{(t)} = \big(1 - \eta \alpha_n^{(t)}\big) w_n^{(t)} \quad \text{(where } \alpha_n^{(t)} = \mathbf{1}[y^{(t)}) \neq y_n^{(t)}] \tag{12}$$

By combining Eq. (12) & (11), one can derive the mathematical relation between $\Phi^{(t+1)}$ and $\Phi^{(t)}$ as follows:

$$\begin{aligned}
\Phi^{(t+1)} = \sum_n w_n^{(t+1)} &= \sum_n (1 - \eta \alpha_n^{(t)}) w_n^{(t)} \quad (\because \text{Eq. (12)}) \\
&= \sum_n w_n^{(t)} - \sum_n \eta \alpha_n^{(t)} w_n^{(t)} \\
&= \Phi^{(t)} - \frac{\Phi^{(t)}}{\Phi^{(t)}} \sum_n \eta \alpha_n^{(t)} w_n^{(t)} \\
&= \Phi^{(t)} - \Phi^{(t)} \eta \sum_n \alpha_n^{(t)} \frac{w_n^{(t)}}{\Phi^{(t)}} \\
&= \Phi^{(t)} - \Phi^{(t)} \eta \sum_n \alpha_n^{(t)} p_n^{(t)} \quad (\text{where } p_n^{(t)} = \frac{w_n^{(t)}}{\Phi^{(t)}}) \\
&= \Phi^{(t)} \Big(1 - \eta \sum_n \alpha_n^{(t)} p_n^{(t)}\Big)
\end{aligned} \tag{13}$$

Assuming all weights are initialized to 1, $\Phi^{(T+1)}$ can be calculated by recursively applying Eq. (13) as follows:

$$\Phi^{(T+1)} = \Phi^{(1)} \prod_{t=1}^{T} \Big(1 - \eta \sum_n \alpha_n^{(t)} p_n^{(t)}\Big) = N \prod_{t=1}^{T} \Big(1 - \eta \sum_n \alpha_n^{(t)} p_n^{(t)}\Big) \tag{14}$$

We can now derive the upper bound of $\Phi^{(T+1)}$ by combining Lemma 8 and Eq. (14) as follows:

$$\begin{aligned}
\Phi^{(T+1)} &= N \prod_{t=1}^{T} \Big(1 - \eta \sum_n \alpha_n^{(t)} p_n^{(t)}\Big) \\
&\leq N \prod_{t=1}^{T} \exp\Big(-\eta \sum_n \alpha_n^{(t)} p_n^{(t)}\Big) \quad (\because \text{Lemma 8}) \\
&= N \exp\Big(-\eta \sum_{t=1}^{T} \mathbb{E}\big[\mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]\big]\Big) \quad (\because \sum_n \alpha_n^{(t)} p_n^{(t)} = \mathbb{E}_p\big[\mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]\big]) \\
&= N \exp\big(-\mathbb{E}[M^{(T)}]\big) \quad (\because \text{ definition of } M^{(t)})
\end{aligned} \tag{15}$$

By applying the same argument as in WMA, the lower bound of $\Phi^{(T+1)}$ is $(1-\eta)^{m_n^{(T)}}$.

$$\Phi^{(T+1)} \geq (1 - \eta)^{m_n^{(T)}} \tag{16}$$

By combining Eq. (15) & (16), we can now derive the mathematical relation between $\mathbb{E}[M^{(t)}]$ and $m_n^{(t)}$ as follows:

$$(1-\eta)^{m_n^{(T)}} \leq \Phi^{(T+1)} \leq N \exp\big(-\mathbb{E}[M^{(T)}]\big) \tag{17}$$

Finally, we can compute the (upper) bound of expected mistakes $\mathbb{E}[M^{(t)}]$ by combining the bounds of $\Phi^{(t+1)}$ (Eq. (17)) and Lemma 4 as follows:

$$\begin{aligned}
m_n^{(T)} \log(1-\eta) &\leq \log N - \eta \mathbb{E}[M^{(T)}] \\
m_n^{(T)}(-\eta - \eta^2) &\leq \log N - \eta \mathbb{E}[M^{(T)}] \quad (\because \text{ Lemma 4}) \\
-m_n^{(T)}(1+\eta) &\leq \frac{\log N}{\eta} - \mathbb{E}[M^{(T)}] \\
\mathbb{E}[M^{(T)}] &\leq (1+\eta) m_n^{(T)} + \frac{\log N}{\eta}
\end{aligned}$$

Therefore, we have obtained the upper bound of expected regret for RWMA. □

We are now interested in the performance (regret bound) of RWMA algorithm.

Theorem 10. *RWMA is a no-regret algorithm.*

Proof. By the definition of regret and Theorem 9, we could obtain the upper bound of *expected regret* of RWMA, $\mathbb{E}[R]$, as follows:

$$\mathbb{E}[R] = \mathbb{E}[M^{(T)}] - m_n^{(T)} \leq \eta m_n^{(T)} + \frac{\log N}{\eta} \tag{18}$$

If we set η to $\frac{1}{\sqrt{T}}$, both $\eta m_n^{(T)}$ and $\frac{\log N}{\eta}$ follow $O(\sqrt{T})$.
Thus, $\frac{\mathbb{E}[R]}{T} = \left(\eta m_n^{(T)} + \frac{\log N}{\eta}\right) \propto \frac{1}{\sqrt{T}} \to 0$ as $T \to \infty$. □

2.3 Online Linear Classification

Now we will start with an Online Linear Classification (OLC) algorithm. Here, we are moving from an exhaustive space to a sampled space, unlike the characterization in Section 1.1. Further, we don't want to program the behavior of the algorithm but learn from an expert, and we will start with two algorithms involving linear hyperplanes to learn the decision boundary:

1. Perceptron algorithm, which has additive updates.
2. Winnow Algorithm, which has multiplicative updates.

Algorithm 3 Perceptron algorithm

```
w^(1) ← 0                                              ▷ Weight initialization
for t = 1, ··· , T do
    RECEIVE (x^(t) ∈ R^N)                              ▷ Receive expert predictions
    ŷ^(t) = sign(w^(t) x^(t))                          ▷ Make learner prediction
    RECEIVE (y^(t) ∈ {−1, 1})                          ▷ Receive actual answer
    w_n^(t+1) ← w_n^(t) + y^(t) · x^(t) · 1[y^(t) ≠ x_n^(t)]   ▷ Weight update
end for
```

In the perceptron algorithm (listed as Algorithm 3 above), if the prediction is correct then no updates are made. Further, we can ask the following questions to characterize the perceptron algorithm:

1. Is the perceptron algorithm fast? Yes. No weight updates are done for correct prediction, while the incorrect prediction update just involves computing a simple dot product and a sum. The prediction itself is just a dot product.
2. Is it a Large Margin classifier? No, there is no notion of margin in the algorithm.
3. Does it work on non-separable data? No, but when the data is linearly separable, the perceptron algorithm will make a finite number of mistakes.

Now again we could derive the mistake bound, following the similar 5-step strategy, as we had used with the weighted majority algorithms. We list the Perceptron Mistake Bound:

$$M \leq \frac{R^2}{\gamma^2} \tag{19}$$

Here, $R = \max_t \|x^t\|$ is the norm of the obserbvations and $\gamma = \min_t y_t$ is the margin of separability.

Further, Block and Novikoff [3] give the mistake bound, but not the regret bound because under the conditions of linear separability, we do not really need to consider regret, as described in Section 1.2.

References

[1] N. Littlestone, M. K. Warmuth, et al. *The weighted majority algorithm.*

[2] M. L. Littman. Reinforcement learning improves behaviour from evaluative feedback. *Nature*, 521(7553):445–451, 2015.

[3] A. B. Novikoff. On convergence proofs for perceptrons. Technical report, Stanford Research Inst Menlo Park CA, 1963.