CS 412 — Introduction to Machine Learning (UIC)	May 10, 2025
Lecture 29	
Instructor: Aadirupa Saha	*Scribe(s): Datta Sai VVN*

Overview

In the last lecture, we covered:

1. Halving Algorithm (HA)
2. Weighted Majority Algorithm (WMA)

In this lecture, we focus on:

1. Exponential Weights Algorithm (EXP–Wt)
   FTRL with entropy regularizer; proof of $R_T = O(\sqrt{T \ln N})$
2. Online Mirror Descent / Online Convex Optimization (OMD/OCO)
   Generalization of EXP–Wt via mirror maps; regret bound

1 Exponential Weights Algorithm

1.1 Expert-Advice Full-Information Setting

We begin with a pool of N experts (or hypotheses) $\{h_1, \dots, h_N\}$. At each round t, the learner:

- Queries each expert's prediction $h_i(x_t)$.
- Suffers *all* expert losses $\ell_{i,t} = \ell\big(h_i(x_t), y_t\big)$ once the true label y_t is revealed.

Because we see the *entire* vector $(\ell_{1,t}, \dots, \ell_{N,t})$ each round, this is known as the **full-information** setting.

Why WMA only for classification. The Weighted Majority Algorithm maintains integer mistake counts and uses *zero–one* loss, so it applies only when $\ell \in \{0, 1\}$ (classification).

Why EXP–Wt handles any loss. By exponentiating arbitrary bounded losses $\ell_{i,t} \in [0, 1]$, the Exponential Weights method adapts to any convex (or even general) loss function. Thus EXP–Wt is suitable for classification, regression, ranking, etc.

1.2 Online Learning Framework

Unlike batch learning, online learning proceeds sequentially: at each round the learner sees only the current instance and must immediately predict.

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

Protocol for $t = 1, \ldots, T$

1. **Receive input:** $x_t \in \mathcal{X}$.
2. **Predict:** $\hat{y}_t \in \mathcal{Y}$ using current predictor f_t.
3. **Receive feedback:** true label y_t is revealed.
4. **Incur loss:** $\ell(y_t, \hat{y}_t)$.
5. **Update predictor:** $f_{t+1} \leftarrow \mathrm{Update}(f_t, x_t, y_t)$.

1.3 Problem Setup and Regret

We have N experts whose predictions at round t incur losses

$$\ell_{i,t} = \ell\big(h_i(x_t), y_t\big), \quad i = 1, \ldots, N.$$

Our learner maintains weights $w_{i,t} > 0$ and defines the mixture distribution

$$p_t(i) \;=\; \frac{w_{i,t}}{\sum_{j=1}^{N} w_{j,t}}, \qquad \ell(p_t, y_t) = \sum_{i=1}^{N} p_t(i)\,\ell_{i,t}.$$

We define the *regret* after T rounds as

$$R_T = \sum_{t=1}^{T} \ell(p_t, y_t) - \min_{i \in [N]} \sum_{t=1}^{T} \ell_{i,t}.$$

An algorithm has *sublinear regret* if $R_T = o(T)$, i.e. $\lim_{T\to\infty} R_T/T = 0$. Our goal is to show that EXP–Wt achieves $R_T = O(\sqrt{T \ln N})$, which implies sublinear regret.

Regret Definition

$$R_T \;=\; \sum_{t=1}^{T} \ell(p_t, y_t) \;-\; \min_{i \in [N]} \sum_{t=1}^{T} \ell_{i,t}.$$

Remark A deterministic expert-selection algorithm can be forced into *linear* regret. For example, with two experts (Alice and Bob), an adversary who assigns loss 1 to whichever expert is chosen (and 0 to the other) each round causes the learner's total loss to be T, while the better expert suffers at most $T/2$. Hence $R_T \geq T - T/2 = \Omega(T)$, showing that *randomized* play is essential for achieving sublinear regret.

1.4 Examples of Online Learning

Binary classification: Rain-or-not prediction; F/FA fraud detection; stock-buy/hold decisions; loan approval.

Multiclass classification: Which stock will be highest tomorrow; tournament outcome prediction.

Regression: Predict tomorrow's temperature; house-price estimation; revenue forecasting.

Concrete Example: Stock-Price Prediction

Consider an online regression task for predicting tomorrow's stock price.

- **Instance:** $x_t = (\text{price}_{t-1},\ \text{volume}_{t-1},\ \dots) \in \mathbb{R}^d$, e.g. closing price and trading volume from the prior day.
- **Label:** $y_t = \text{price}_t \in \mathbb{R}$, the actual closing price on day t.
- **Loss:** $\ell(\hat{y}_t, y_t) = (\hat{y}_t - y_t)^2$ (squared-error) or $|\hat{y}_t - y_t|$ (absolute error).

Each expert h_i might be a simple autoregressive model (e.g. $h_i(x_t) = w_i^\top x_t$) or any other predictor; EXP–Wt will maintain weights over these N experts and mix their predictions to minimize cumulative squared (or absolute) error in this full-information setting.

1.5 Exponential Weights Algorithm Description

The Exponential Weights algorithm maintains a distribution p_t over experts that is updated multiplicatively according to observed losses. Concretely, at each round:

1. **Form distribution:**
$$p_t(i) \ \propto \ w_{i,t} = \exp\big(-\eta\, L_{i,t-1}\big), \quad L_{i,t-1} = \sum_{s=1}^{t-1} \ell_{i,s}.$$
2. **Predict:** Use the mixture prediction $\hat{y}_t = \sum_i p_t(i)\, h_i(x_t)$.
3. **Observe losses:** Receive $\ell_{i,t} = \ell\big(h_i(x_t), y_t\big)$ for all i.
4. **Update weights:**
$$w_{i,t+1} \ = \ w_{i,t}\, \exp\big(-\eta\, \ell_{i,t}\big).$$

This ensures that experts incurring high loss are exponentially down-weighted.

1.6 Pseudocode

Algorithm 1.1: Exponential Weights (EXP–Wt)

```
Input: Number of experts N, learning rate η > 0
Initialize w_{i,1} ← 1 for all i ∈ [N]
for t = 1 to T do
    Receive x_t
    Form p_t(i) ← w_{i,t} / Σ_j w_{j,t}
    Predict according to p_t
    Observe ℓ_{i,t} for all i
    Update w_{i,t+1} ← w_{i,t} exp(−η ℓ_{i,t})
end for
```

1.7 Regret Bound (Proof of Theorem)

Theorem 1.1: Regret of EXP–Wt

If we set

$$\eta = \sqrt{\frac{8 \ln N}{T}},$$

then Algorithm 1.6 achieves

$$R_T \;\le\; \sqrt{\frac{T}{2}\,\ln N}\,.$$

Proof. Let $W_t = \sum_{i=1}^{N} w_{i,t}$. Since $w_{i,1} = 1$, $W_1 = N$. Fix the best expert $i^* = \arg\min_i \sum_{t=1}^{T} \ell_{i,t}$. Then

$$w_{i^*,T+1} = \exp\Big(-\eta \sum_{t=1}^{T} \ell_{i^*,t}\Big) \;\le\; W_{T+1},$$

so

$$\ln \frac{W_{T+1}}{W_1} \;\ge\; -\eta \sum_{t=1}^{T} \ell_{i^*,t} - \ln N. \tag{1}$$

Hoeffding's Lemma. For any random variable $X \in [a, b]$ and any $s \in \mathbb{R}$,

$$\mathbb{E}\big[e^{s(X - \mathbb{E}[X])}\big] \;\le\; \exp\Big(\tfrac{s^2(b-a)^2}{8}\Big).$$

In our setting, $\ell_{i,t} \in [0, 1]$. Applying this with $s = -\eta$ under the distribution p_t gives:

$$\sum_{i=1}^{N} p_t(i)\, e^{-\eta(\ell_{i,t} - \ell(p_t, y_t))} \;\le\; \exp\Big(\tfrac{\eta^2}{8}\Big).$$

Rearranging yields

$$\sum_{i=1}^{N} p_t(i)\, e^{-\eta \ell_{i,t}} \;\le\; \exp\big(-\eta\, \ell(p_t, y_t) + \tfrac{\eta^2}{8}\big).$$

Hence for each round

$$\frac{W_{t+1}}{W_t} = \sum_{i=1}^{N} p_t(i)\, e^{-\eta \ell_{i,t}} \;\le\; \exp\big(-\eta\, \ell(p_t, y_t) + \tfrac{\eta^2}{8}\big). \tag{2}$$

Summing (2) over $t = 1, \dots, T$ gives

$$\ln \frac{W_{T+1}}{W_1} \;\le\; -\eta \sum_{t=1}^{T} \ell(p_t, y_t) + \frac{T\,\eta^2}{8}.$$

Combining (1) and (2) and rearranging yields

$$R_T \;\le\; \frac{\ln N}{\eta} + \frac{T\eta}{8}.$$

Choosing $\eta = \sqrt{8 \ln N / T}$ balances the two terms and completes the proof. □

Exercise 1.1: Toy Example: N

Suppose two experts incur alternating losses $\ell_{1,t} = 0, 1, 0, 1, \ldots,\ \ell_{2,t} = 1, 0, 1, 0, \ldots.$ Initialize $w_{1,1} = w_{2,1} = 1$. With $\eta = \sqrt{8 \ln 2/T}$, track $w_{i,t}$ and verify the regret bound numerically for small T.

2 Online Mirror Descent / OCO

2.1 Setup and Notation

Let $\mathcal{K} \subset \mathbb{R}^d$ be a closed convex set. We choose a differentiable regularizer $R : \mathcal{K} \to \mathbb{R}$ that is β-*strongly convex* with respect to a norm $\|\cdot\|$, i.e.

$$R(u) \ \geq \ R(v) + \big\langle \nabla R(v),\, u - v \big\rangle + \tfrac{\beta}{2}\|u - v\|^2, \quad \forall\, u, v \in \mathcal{K}.$$

Define:

- $\|\theta\|_* = \sup_{\|w\| \leq 1} \langle \theta, w \rangle$ the dual norm.
- The convex conjugate (Legendre–Fenchel transform)
$$R^*(\theta) = \sup_{w \in \mathcal{K}} \{\langle \theta, w \rangle - R(w)\}.$$
- The Bregman divergence
$$D_R(u\|v) = R(u) - R(v) - \langle \nabla R(v),\, u - v \rangle.$$

2.2 Algorithm Description

At each round t, given the current point $w_t \in \mathcal{K}$ and subgradient $g_t \in \partial \ell_t(w_t)$, OMD updates as follows:

1. *Dual step:*
$$u_{t+1} = \nabla R^*\big(\nabla R(w_t) - \eta\, g_t\big).$$
2. *Primal projection:*
$$w_{t+1} = \arg\min_{w \in \mathcal{K}} D_R\big(w \| u_{t+1}\big).$$

2.3 Pseudocode

Algorithm 2.1: Online Mirror Descent (OMD)

Input: Convex set $\mathcal{K}$, β-strongly convex R, step size $\eta > 0$
Initialize $w_1 \leftarrow \arg\min_{w \in \mathcal{K}} R(w)$
for $t = 1$ to T **do**
  Receive convex loss ℓ_t and pick w_t
  Compute subgradient $g_t \in \partial \ell_t(w_t)$
  Dual step: $u_{t+1} \leftarrow \nabla R^*(\nabla R(w_t) - \eta\, g_t)$
  Primal projection: $w_{t+1} \leftarrow \arg\min_{w \in \mathcal{K}} D_R(w \| u_{t+1})$
end for

2.4 Regret Bound

Theorem 2.1: Regret of OMD

Assume each loss ℓ_t is convex and all subgradients satisfy $\|g_t\|_* \le G$. Let R be β-strongly convex on $\mathcal{K}$, and define the initial Bregman diameter

$$D = \max_{w \in \mathcal{K}} D_R(w \| w_1).$$

Then OMD with constant step-size $\eta > 0$ ensures

$$R_T = \sum_{t=1}^{T} \ell_t(w_t) - \min_{u \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(u) \;\le\; \frac{D}{\eta} + \frac{\eta\, G^2\, T}{2\beta}.$$

In particular, choosing $\eta = \sqrt{\frac{2\beta\, D}{G^2\, T}}$ gives

$$R_T \;\le\; G\sqrt{\tfrac{2D\,T}{\beta}} = O\big(G\sqrt{D\,T}\big).$$

2.5 Recovering EXP–Wt

When $\mathcal{K} = \Delta_N$ and

$$R(p) = \sum_{i=1}^{N} p_i \ln p_i \quad (\text{so } \beta = 1),$$

OMD coincides with the Exponential Weights algorithm of Section **1.6**, and Theorem **2.1** recovers the same $O(\sqrt{T \ln N})$ bound.

Next Lecture

The next lecture will cover:

1. EXP–Wt under stochastic vs. adversarial feedback
2. Bandit extensions and partial-information regret
3. Tuning and adaptive learning rates

References

1. Cesa-Bianchi, N. and G. Lugosi. *Prediction, Learning, and Games*. Cambridge Univ. Press, 2006.
2. Shalev-Shwartz, S. "Online Learning and Online Convex Optimization." *Foundations and Trends in Machine Learning*, 2012.
3. Rakhlin, S. "Mindreader demo," UCSD, 2008. http://seed.ucsd.edu/mindreader/
4. Piazza handout: "Proof of Theorem 2.1 (EXP–Wt)," May 2025. "https://piazza.com/class/m5vnz1h225x1c0/post/10"