# Online Classification — Part 2

Source:

Online Learning and Online Convex Optimization, by Shai Shalev-Shwartz, Found. and Trends in ML, Vol. 4, No. 2, 2011 (Chapters 1, 3).

---

Setting: feature space $\mathcal{X}$. Classifiers: $\mathcal{H}$
Label space $\mathcal{Y}$

$h \in \mathcal{H}$ : $h : \mathcal{X} \longrightarrow \mathcal{Y}$ i.e., $x \in \mathcal{X}$, $y \in \mathcal{Y}$

$h(x) = y$, a classifier.

Typically, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{0, 1\}$ (or $\{-1, +1\}$)

$\longrightarrow$ binary classification

Online problem — evolves in discrete time steps, $t = 1, 2, \ldots, T$.

---

# Infinite Hypotheses: Unrealizable Case

($|\mathcal{H}| = \infty$, but $\mathcal{L}\dim(\mathcal{H}) < \infty$).

Unrealizable case $\Rightarrow$ Player takes randomized actions.

Setting: $|\mathcal{H}|$ potentially infinite; $p_t \in [0,1]$, $y_t \in \{0,1\}$. Goal is to minimize expected regret w.r.t. best hypothesis in hindsight. $\mathcal{L}\dim(\mathcal{H}) < \infty$

We will show that $R_T \leq O\left(\sqrt{\mathcal{L}\dim(\mathcal{H}) \cdot \ln(T) \cdot T}\right)$

---

Idea: (to deal with $|\mathcal{H}| = \infty$, but $\mathcal{L}\dim(\mathcal{H}) < \infty$)

① Previously with finite $|\mathcal{H}|$, each hypothesis was an expert. Now, construct a finite number of experts such that:

For each $h \in \mathcal{H}$, and any sequence of features $\{x_1, x_2, \dots, x_T\}$, $\exists$ an expert s.t. the experts prediction $\equiv$ prediction from $h(\cdot)$.

assume $T \geq \mathcal{L}\dim(\mathcal{H})$.

Specifically, for each $L \leq \mathcal{L}\dim(\mathcal{H})$, and any sequence of time indices: $1 \leq i_1 < i_2 < i_3 \dots < i_L \leq T$,

we will construct an expert.

② Then use weighted majority (i.e., Exp4 will full information over this group of experts).

---

## Operation of an Expert

$$\text{Expert}(L, i_1, i_2, \ldots, i_L)$$

Input: $\mathcal{H}$, $L$, $i_1, i_2, \ldots, i_L$.

Initial: $V_1 = \mathcal{H}$.

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

- Observe $x_t$
- Define: $V_t^{(0)} = \{h \in V_t : h(x_t) = 0\}$
  $V_t^{(1)} = \{h \in V_t : h(x_t) = 1\}$

$$\tilde{y}_t = \underset{r \in \{0,1\}}{\operatorname{argmax}} \; L\dim\left(V_t^{(r)}\right)$$

(if $V_t^{(0)} = V_t^{(1)}$, then set $\tilde{y}_t = 0$).

- For $t \in \{i_1, i_2, \ldots, i_L\}$, set $\hat{y}_t = |1 - \tilde{y}_t|$ (i.e., the opposite label)

For $t \notin \{i_1, i_2, \ldots, i_L\}$, set $\hat{y}_t = \tilde{y}_t$.

- Update $V_{t+1} = V_t^{(\hat{y}_t)}$

---

Intuition: Recall, that for each $x_t$, $\Pi_{SOA}$ (i.e., the Standard Optimal Algorithm) predicts the label corresponding to the larger Ldim of the remaining set of hypothesis (ie instead of majority, go with larger Ldim).

This expert above ASSUMES SOA makes a mistake at times $\{i_1, i_2, \ldots, i_L\}$. Thus, it flips the label predicted by SOA at those times, but follows SOA at all other times.

---

Note: # of experts is finite. Specifically, the number of experts E is given by

$$E = \sum_{L=0}^{Ldim(\mathcal{H})} \underbrace{\binom{T}{L}}_{\text{choose } L \text{ time-slots to flip SOA's recommendation} \rightarrow \text{each such choice defines an expert.}} \leq \left(\frac{eT}{Ldim(\mathcal{H})}\right)^{Ldim(\mathcal{H})}$$

---

**Lemma (3.7 in monograph):** $\mathcal{H}$ a hypothesis class with $Ldim(\mathcal{H}) < \infty$. $\{x_1, x_2, \ldots, x_T\}$ a sequence of features. For any $h \in \mathcal{H}$, $\exists$ an expert $Expert(L, i_1, i_2, \ldots, i_L)$ s.t. for this sequence of features, this expert predicts $y_t = h(x_t)$ (i.e., the $Expert(L, i_1, \ldots i_L)$ and $h$ agree on labels for all times $t = 1, 2, \ldots, T$.

**Proof:**

- Fix $h \in \mathcal{H}$, $\{x_1, x_2, \ldots, x_T\}$.
- Consider the sequence $\{(x_1, h(x_1)), (x_2, h(x_2)), \ldots, (x_T, h(x_T))\}$. Run SOA with this sequence. Suppose that SOA makes $L$ mistakes, at times $i_1, i_2, \ldots, i_L$. (Recall that $L \leq Ldim(\mathcal{H})$ for the SOA algorithm). Thus, SOA predicts:

$\{(x_1, h(x_1)), \ldots, (x_{i_1}, |1 - h(x_{i_1})|), \ldots, (x_k, h(x_k)), \ldots$

$\ldots, (x_{i_L}, |1-h(x_{i_L})|), \ldots, (x_T, h(x_T))\}$

Here, green times are correct labels by SOA (and thus match $h$), and red times correspond to incorrect labels by SOA (and thus flip $h$).

- Pick the Expert$(L, i_1, i_2, \ldots, i_L)$. By construction, this expert flips SOA at red times, and thus matches $h$ at all times.

∎

Corollary (3.8): Let $\{(x_1, y_1), \ldots, (x_T, y_T)\}$ be a labeled sequence, $\mathcal{H}$ a hypothesis class with $Ldim(\mathcal{H}) < \infty$. Then, $\exists$ Expert$(L, i_1, i_2, \ldots, i_L)$ that makes no more mistakes that the best $h \in \mathcal{H}$ for this labeled sequence, i.e.,

$$\#\text{ of mistakes by Expert}(L, i_1, i_2, \ldots, i_L) \leq \min_{h \in \mathcal{H}} \sum_{t=1}^{T} |h(x_t) - y_t|.$$

Proof: Immediately follows from Lemma above ∎

Theorem (3.6 in monograph): For any hypothesis class $\mathcal{H}$ s.t. $Ldim(\mathcal{H}) < \infty$, $\exists$ algorithm $\pi$ s.t. for any $h \in \mathcal{H}$, and labeled sequence $\{(x_1, y_1), \cdots, (x_T, y_T)\}$,

$$\sum_{t=1}^{T} |p_t - y_t| - \sum_{t=1}^{T} |h(x_t) - y_t|$$

$$\leq O\left(\sqrt{Ldim(\mathcal{H}) \cdot (\ln T) \cdot T}\right).$$

Proof: Use weighted majority algorithm with the finite set of experts constructed above.

Note $$E \leq \left(\frac{eT}{Ldim(\mathcal{H})}\right)^{Ldim(\mathcal{H})}$$

Corollary 3.8 + Weighted majority proof $\Rightarrow$ Result. $\square$

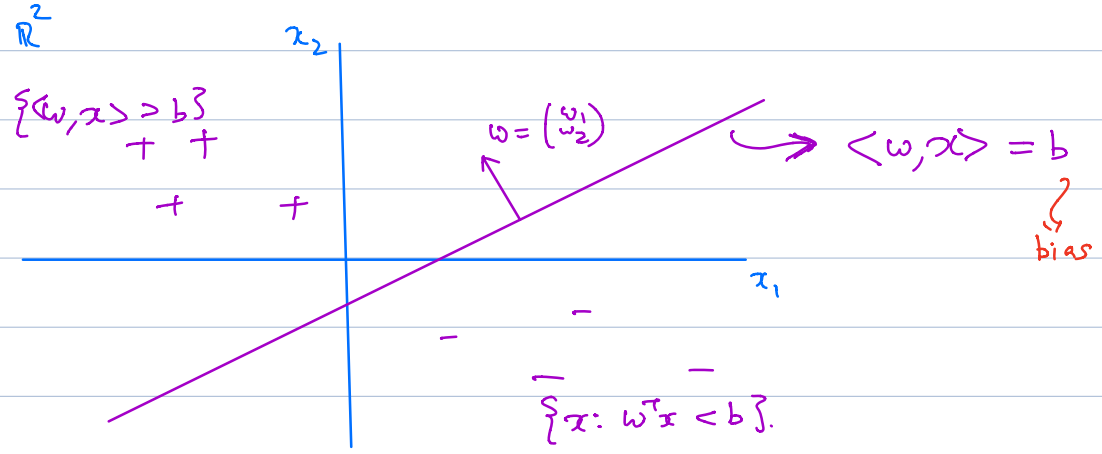
---

Binary Classification over $x_t \in \mathbb{R}^d$ where $Ldim(\mathcal{H}) = \infty$

Perceptron and Winnow Algorithms

Setting: $y \in \{-1, +1\}$ (instead of $\{0, 1\}$ as before).

$\mathcal{X} = \mathbb{R}^d$, $\mathcal{D}$: $p_t \in \{-1, +1\}$.




Note: We will set $b = 0$ in the sequel. Bias can easily be addressed by considering the above in $(d+1)$ dimensions with $x_{d+1} = 1$, and $w_{d+1} = -b$.

---

Setting: In round $t = 1, 2, \ldots, T$, player observes $x_t \in \mathbb{R}^d$.

- Player has $w_t \in \mathbb{R}^d$ (weight vector), and updates it sequentially. Specifically, player predicts

$$p_t = \text{sign}\left(\langle w_t, x_t\rangle\right) \in \{-1, +1\}$$

- Observes label $y_t \in \{-1, +1\}$
- $\ell(p_t, y_t) = \chi_{\{p_t \neq y_t\}} = |p_t - y_t|$

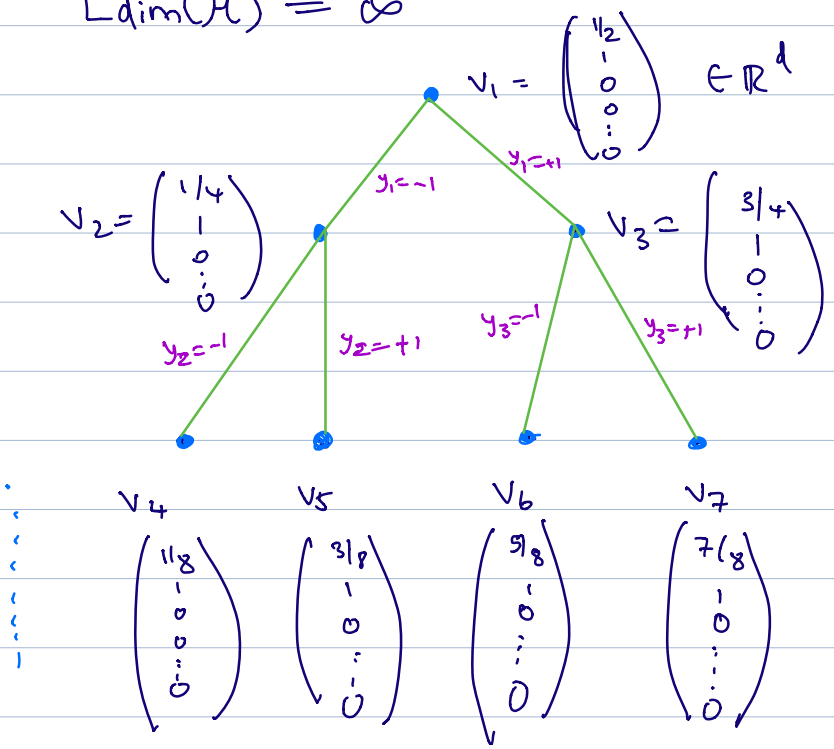
In this case, since $w_t \in \mathbb{R}^d$,

$$\mathcal{H} = \{ w \in \mathbb{R}^d : \; p_t = \text{sign}(\langle w, x \rangle) \}$$

i.e., $h \in \mathcal{H}$, with $h_w(x) = \text{sign}(\langle w, x \rangle)$.
(set of half spaces in $\mathbb{R}^d$).

Claim: $\text{Ldim}(\mathcal{H}) = \infty$

Proof:



Now consider $\tilde{\mathcal{H}} \subseteq \mathcal{H}$, where $h \in \tilde{\mathcal{H}}$ parameterized by $\omega = (-1, a, 0, \cdots 0)$, $a \in [0,1]$. Observe that the infinite tree above can be shattered by the hypotheses in $\tilde{\mathcal{H}}$. Thus, $Ldim(\mathcal{H}) = \infty$ ☒.

---

Note: **Convexification Using Surrogate Loss Functions**

Ref: Sec. 2.1.2, monograph + Sec. 3.3.1 (Perceptron).

Above setting has $Ldim(\mathcal{H}) = \infty$, so it is not immediately clear how to use randomization. This note discusses an alternate approach, namely surrogate convex loss functions (from sec. 2.1.2). The high-level idea:

① Observe that the loss function here is $\chi_{\{p_t \neq y_t\}}$, $= |p_t - y_t|$, where $p_t = \text{sign}(\langle \omega_t, x_t \rangle)$, i.e., the loss function at each time:

$$\ell_t(\omega, (x,y)) = \chi_{\{y\langle \omega, x\rangle \leq 0\}}$$

↑ ↑

iff signs are opposite, the product is $-ve$, and $\ell_t(\cdot) = 1$.

② Upper bound $\ell_t(\cdot)$ by a surrogate convex loss function, i.e., construct a sequence of functions $\{f_t(w), t=1,2,\dots\}$ s.t.

- $f_t(w)$ is convex, and
- $\forall w, t,\ f_t(w) \geq \ell(w, (x_t, y_t))$

③ Using online convex optimization (in full feedback setting), show that online gradient descent with $f_t(\cdot)$ has low regret.

④ Use upper bound property in ② to derive a mistake bound for the $\ell_t(\cdot)$ loss function.

---

## The Perceptron Algorithm

actual loss

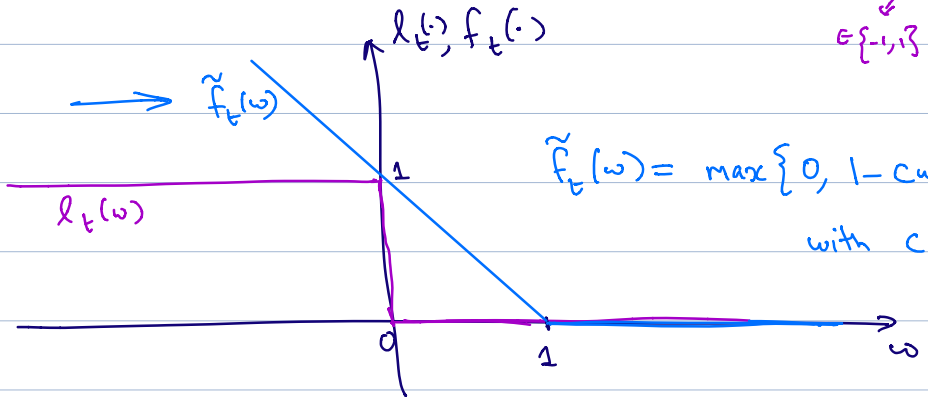
Loss $\ell_t(w, (x,y)) = \chi_{\{y\langle w, x\rangle \leq 0\}}$

Define $\tilde{f}_t(w) = [1 - y_t\langle w, x_t\rangle]_+$,

where $[r]_+ = \max\{r, 0\}$

hinge loss

e.g. if $w \in \mathbb{R}$, (for some fixed $(x,y)$ with $\in \mathbb{R}$, $xy>0$, $y \in \{-1,1\}$)




$\tilde{f}_t(w) = \max\{0, 1-cw\}$ with $c = 1$

Original loss fn: $cw \to \{-1,+1\}$ (replaced by) $\left(cw \to \text{sign}\left(y\langle w, x\rangle\right)\right)$

---

Observe: ① $\tilde{f}_t(w)$ convex in $w$.

② $\tilde{f}_t(w) = \left[1 - y_t\langle w, x_t\rangle\right]_+ \geq \chi_{\{y_t\langle w, x_t\rangle \leq 0\}}$

$= \ell_t(w, (x_t, y_t))$.

Now, define

$$f_t(w) = \begin{cases} \tilde{f}_t(w) & \text{if mistake made at time } t \\ 0 & \text{if no mistake} \end{cases}$$

Note that $f_t(w)$ satisfies ① $f_t(w)$ convex in $w$ for each $t$, and ② $f_t(w) \geq \ell_t(w, (x_t, y_t))$. $\forall\, w, t$.

$\{f_t(w)\}$ is the sequence of convex surrogate loss functions.

**Algorithm** — Online Gradient Descent on $w_t$ using surrogate functions.

$w_1 = 0$ $\qquad (w_i \in \mathbb{R}^d)$.

$w_{t+1} = w_t - \eta z_t$, $\quad z_t \in \partial f_t(w_t)$.
↳ subgradient.

Here: no mistake $\iff y_t \langle w_t, x_t \rangle > 0$
$\iff f_t(\cdot) = 0$ : Set $z_t = 0$

mistake: $z_t = -y_t x_t \in \partial f_t(w_t)$.

$\therefore$ $$w_{t+1} = \begin{cases} w_t + \eta y_t x_t & \text{if } y_t \langle w_t, x_t \rangle \leq 0 \\ w_t & \text{if } y_t \langle w_t, x_t \rangle > 0 \end{cases}$$

Let $\mathcal{W} \subseteq \{1, 2, 3, \ldots, T\}$ be the time slots where mistakes were made, i.e.,

$$s \in \mathcal{W} \iff \underbrace{\text{sign}\left(\langle w_t, x_t \rangle\right)}_{p_t} \neq y_t$$

Observe also that: ($\eta > 0$)

$$p_t = \text{sign}\left(\langle w_t, x_t \rangle\right) = \text{sign}\left(\eta \sum_{\substack{i \in \mathcal{W}: \\ i < t}} y_i \langle x_i, x_t \rangle\right)$$

$$= \text{sign}\left(\sum_{\{i \in \mathcal{W}: i < t\}} y_i \langle x_i, x_t \rangle\right)$$

i.e., invariant to $\eta$.

$\therefore$ **Perceptron Algorithm:** (single neuron with threshold activation)

$$p_t = \text{sign}\left(\langle w_t, x_t \rangle\right)$$

$$w_{t+1} = \begin{cases} w_t + y_t x_t \cdot \eta & \text{if } y_t \langle w_t, x_t \rangle \leq 0 \\ w_t & \text{if } y_t \langle w_t, x_t \rangle > 0 \end{cases}$$

$\eta > 0$ is immaterial. Choose this appropriately for analysis bounds. Algo. does not change.

## Analysis:

Use OGD (online gradient descent) bound for convex functions. This is analogous to the bounds we studied for online linear optimization, see Chap 2 of monograph for details. Specifically, it can be shown that:

$$\underbrace{\sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u)}_{\geq |W| \text{ by property of surrogate loss.}} \leq \frac{1}{2\eta}\|u\|_2^2 + \frac{\eta}{2}|W| R^2,$$

where $R = \max_{1 \leq t \leq T} \|x_t\|_2$.

$\therefore$ Choose $\eta = \|u\|_2 / R\sqrt{|W|}$

$$\Rightarrow |W| - \sum_{t=1}^{T} f_t(u) - R\|u\|\sqrt{|W|} \leq 0$$

$\hookrightarrow$ (*)

**Theorem** (3.9 in monograph): With sequence $\{(x_1, y_1), \dots, (x_T, y_T)\}$, and $R = \max_{1 \le t \le T} \|x_t\|_2$, and $|\mathcal{W}| = \#$ of mistakes until time $T$. Then for any $u \in \mathbb{R}^d$,

$$|\mathcal{W}| \le \sum_{t=1}^{T} f_t(u) + R\|u\|_2 \sqrt{\sum_{t=1}^{T} f_t(u)} + R^2 \|u\|_2^2$$

Further, if $\exists\, u$ s.t. $y_t \langle u, x_t \rangle \ge 1 \ \forall\, t$ (also called seperability with large margin)

$$|\mathcal{W}| \le R^2 \|u\|^2.$$

**Proof:** follows from (*). Also

$$x - b\sqrt{x} - c \le 0$$

$$\Downarrow$$

$$x \le c + b^2 + b\sqrt{c}.$$

Last claim using algebra from (*). ∎

---

**Note:** It can happen that $\exists\, u$ with no mistakes (but does not satisfy large margin condition), but the perceptron algorithm can make many mistakes, because regret w.r.t $\sum_t f_t(u)$ (and not the actual loss function).

Thus, bound is useful only when $\exists u$ s.t. $\sum_t f_t(u)$ is small.

# Binary Classification: From OGD to OMD

Online Gradient Descent — Online Mirror Descent

Previously: By changing regularizer and generalizing gradient descent to mirror descent, substantial improvement in regret in certain settings (mirror descent over simplex with KL divergence as regularizer).

This section: Similar plan: Interpret Perceptron as Online Gradient Descent. Winnow (algo. to be described below) is an Online Mirror Descent algo.

---

Note: These algorithms have a much longer history. Precursor to Exp3 (and boosting in the case of full information) is the Weighted Majority Algorithm (Littlestone and Warmuth, 1994). Winnow predates even that (late 80s). Perceptron goes back to (Rosenblatt, 1958).

---

The unaltered problem: $X = \{0,1\}^d$, ie.,

$x \in \mathcal{X}$ is a binary feature vector. We are in a setting where we know that only $k \leq d$ of these coordinates are useful; rest are spurious. If any of the useful coordinates equals '1', the label is '1'. Thus, we need to find a $w \in \{0,1\}^d$ (a binary weight vector) that is $k$-sparse, and the associated classifier is given by:

$$h(x) = \text{sign}(2\langle w, x\rangle - 1).$$

$\langle w, x\rangle = 0 \Rightarrow h(x) = -1$

$\langle w, x\rangle \geq 1 \Rightarrow h(x) = +1$

$$\in \begin{cases} +1 & \langle w, x\rangle > 0 \\ -1 & \langle w, x\rangle \leq 0 \end{cases}$$

$$\mathcal{H} = \{x \mapsto \text{sign}(2\langle w, x\rangle - 1), \ \|w\|_1 = k, \ w \in \{0,1\}^d\}$$

**Convex Relaxation + Surrogate Hinge Loss**

1. Domain Relaxation: $w \in \mathbb{R}^d_+ = \{x \in \mathbb{R}^d : x_i \geq 0\}$. (positive orthant).

2. Surrogate loss function: Exact loss function is

$$\ell(w,(x,y)) = \chi_{\{y(2\langle w,x\rangle - 1) \le 0\}}.$$

Let $f_t(w) = \begin{cases} 0 & \text{if no mistake by algo. at time } t \text{ (i.e., } y = \text{sign}(2\langle w,x\rangle - 1)\text{)} \\ [1 - y_t(2\langle w, x_t\rangle - 1)]_+ & \text{if mistake at time } t. \end{cases}$

↳ hinge loss.

Observe:

① $f_t(w)$ is convex in $w$ for $\forall t = 1, 2, \ldots, T$

② $f_t(w) \ge \ell(w, (x_t, y_t))$

---

**Goal** Develop regret bound on number of mistakes with algo. over time horizon $T$.

Caveat: Algo. uses weight vector $w \in \mathbb{R}^d_+$. However regret with respect to best weight vector $u$ s.t. $\|u\|_1 = k$, $u \in \{0,1\}^d$. In other words, the algo. has more "power" than the genie's best in hindsight.

If realizable, assume $\exists\, u \in \{0,1\}^d$ s.t. $\|u\|_1 = k$ and $f_t(u) = 0 \quad \forall t$.

**Base line Algo: Perceptron**

$$x \longrightarrow \phi(x) = \begin{pmatrix} 2x \\ -1 \end{pmatrix} \in \mathbb{R}^{d+1}$$

$x \in \{0,1\}^d$ ; $\phi(x)$: modified feature vector

Use perceptron with $p_t = \text{sign}(\langle w, \phi(x) \rangle)$.

Then, with realizability assumption $\left(\exists\, u \in \{0,1\}^d \text{ s.t. } \|u\|_1 = k,\ f_t(u) = 0\right)$,

we have: $R^2 \|u\|_2^2 = \left(\max \|x\|_2\right)^2 \left(\|u\|_2^2\right)$

$= (4d+1)\, k$

$\therefore$ From Thm 3.9 (pp. 16 in these notes), we have

$$|W| \leq R^2 \|u\|_2^2 = (4d+1)\, k.$$

We will see with Winnow (a exponential weighting

algorithm) that $|W| \leq 8k \ln(d)$, an this is an exponential improvement due to OMD instead of OGD.

---

**Winnow Algorithm $\equiv$ Unnormalized Exp Weighting.**

Input: $\eta > 0$ (set $\eta = 1/4$ for bounds).

Initial: $w_1 = \left(\frac{1}{d} \; \frac{1}{d} \cdots \frac{1}{d}\right)^T$, $w_1 \in \mathbb{R}_+^d$.

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

- Observe $x_t \in \{0,1\}^d$
- Predict $p_t = \text{sign}\left(2\langle w_t, x_t\rangle - 1\right)$
- If $y_t\left(2\langle w_t, x_t\rangle - 1\right) > 0$, (correct prediction)

  set $w_{t+1} = w_t$ (no change)

  else if $y_t\left(2\langle w_t, x_t\rangle - 1\right) \leq 0$,

$$w_{t+1}(i) = w_t(i)\, e^{-\eta\, 2 y_t \overbrace{x_t(i)}^{\in\{0,1\}}}, \quad i = 1, 2, \cdots, d.$$

($y_t \in \{-1, 1\}$)

---

**Theorem (3.10 in monograph)** $\{(x_1, y_1), \cdots, (x_T, y_t)\}$ labeled samples, and $\eta \le 1/2$. Let $\mathcal{W} \subseteq \{1, 2, \cdots, T\}$ time steps where mistakes occur with Winnow. Let

$$f_t(w) = \chi_{\{t \in \mathcal{W}\}} \cdot \left[1 - y_t\left(2\langle w_t, x_t\rangle - 1\right)\right]_+ .$$

Then, for any $u \in \{0,1\}^d$, we have:

$$|\mathcal{W}| \le \sum_{t=1}^{T} f_t(w_t) \le \frac{1}{(1-2\eta)}\left(\sum_{t=1}^{T} f_t(u) + \frac{k \ln(d)}{\eta}\right)$$

Further, if $\exists\, u$ s.t. $y_t\left(2\langle u, x_t\rangle - 1\right) \ge 1 \;\forall\, t$, then with $\eta = 1/4$:

$$|\mathcal{W}| \le 8 k \ln(d).$$

Proof: Uses the OMD bound for online convex optimization (mirror descent with KL regularizer on positive orthant, read Thm 2.23 in monograph).

Define $z_t = \begin{cases} 2y_t x_t & \text{if } t \in \mathcal{W} \\ 0 & \text{if } t \notin \mathcal{W} \end{cases}$

Then from Thm 2.23 (please read monograph; this is similar to several proofs we have done with KL regularizer in OCO), we have: $u \in \{0,1\}^d$, $\|u\|_1 = k$,

$$\sum_{t=1}^{T} \langle w_t - u, z_t \rangle \leq \frac{1 + \sum_{i=1}^{d} u_i \ln(d u_i / e)}{\eta} + \eta \sum_{t=1}^{T} w_t(i) \cdot z_t(i)^2$$

Further, $\|u\|_1 = k$, $u \in \{0,1\}^d \Rightarrow$

$$1 + \sum_{i=1}^{d} u_i \ln\left(\frac{d u_i}{e}\right) \leq 1 + \ln d.$$

Also, $z_t \in \partial f_t \big|_{w_t}$, and $f_t$ convex. Thus,

$$\sum_{t=1}^{T} \left(f_t(w_t) - f_t(u)\right) \leq \sum_{t=1}^{T} \langle w_t - u, z_t \rangle \leq \frac{k \ln d}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{d} w_t(i) z_t(i)^2$$

Next, $\forall\ t = 1, 2, \cdots, T$, $\sum_{i=1}^{d} w_t(i)\, z_t(i)^2 \le 2 f_t(w_t)$

$\hookrightarrow$ (*)

Case(i): $t \notin W \Rightarrow f_t(w_t) = 0$, $z_t(i) = 0$. $\forall\ i = 1, 2, \cdots d$.

$\Rightarrow$ (*) holds.

Case(ii): $t \in W$, and $y_t = 1$. Then LHS in (*) is

$$\sum_{i=1}^{d} w_t(i)\, z_t(i)^2 = 4 \langle w_t, x_t \rangle$$

$z_t(i)^2 = 2\, y_t\, x_t(i)$, $y_t = 1$, $\to \in \{0, 1\}$

$\Rightarrow 4 \langle w_t, x_t \rangle \le 2$.

$t \in W \Rightarrow 2 \langle w_t, x_t \rangle - 1 \le 0$

Also RHS in (*) $= 2 f_t(w_t) \ge 2 \ge$ LHS.

($f_t(w_t) \ge 1$ when $t \in W$)

Case(iii): $t \in W$, $y_t = -1$. LHS in (*) $= 4 \langle w_t, x_t \rangle$.

RHS in (*) $= 2 f_t(w_t) = 2 \left( 2 \langle w_t, x_t \rangle \right) =$ RHS.

mistake occurs

$= \left[ 1 - (-1)\, 2 \langle w_t, x_t \rangle - 1 \right] = 2 \langle w_t, x_t \rangle$

$$\therefore \quad \sum_{t=1}^{T} f_t(w_t) \le \frac{1}{1-2\eta}\left(\sum_{t=1}^{T} f_t(u) + \frac{k \ln(d)}{\eta}\right)$$

$\square$

---

Please read monograph chap 2 for the OGD, OMD discussion with convex functions. This is analogous to the discussions for the linear case we studied. However, ideas immediately apply because of linear approximation for a convex fn.

$f_t$ convex $\Rightarrow$ for any $u$,

$$f_t(w_t) - f_t(u) \le \langle w_t - u, z_t \rangle$$

↳ loss vector

linear optimization with loss vector defined by gradient (or sub-gradient) of $f_t$ at $w_t$