Foundations of Machine Learning
On-Line Learning

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Motivation

- PAC learning:
  - distribution fixed over time (training and test).
  - IID assumption.
- On-line learning:
  - no distributional assumption.
  - worst-case analysis (adversarial).
  - mixed training and test.
  - Performance measure: mistake model, regret.

This Lecture

- Prediction with expert advice
- Linear classification

General On-Line Setting

- For $t = 1$ to T do
  - receive instance $x_t \in X$.
  - predict $\widehat{y}_t \in Y$.
  - receive label $y_t \in Y$.
  - incur loss $L(\widehat{y}_t, y_t)$.
- Classification: $Y = \{0, 1\}, L(y, y') = |y' - y|$.
- Regression: $Y \subseteq \mathbb{R}, L(y, y') = (y' - y)^2$.
- Objective: minimize total loss $\sum_{t=1}^{T} L(\widehat{y}_t, y_t)$.

Prediction with Expert Advice

- For $t = 1$ to T do
  - receive instance $x_t \in X$ and advice $y_{t,i} \in Y, i \in [1, N]$.
  - predict $\widehat{y}_t \in Y$.
  - receive label $y_t \in Y$.
  - incur loss $L(\widehat{y}_t, y_t)$.
- Objective: minimize regret, i.e., difference of total loss incurred and that of best expert.

$$\text{Regret}(T) = \sum_{t=1}^{T} L(\widehat{y}_t, y_t) - \min_{i=1}^{N} \sum_{t=1}^{T} L(y_{t,i}, y_t).$$

Mistake Bound Model

- Definition: the maximum number of mistakes a learning algorithm L makes to learn c is defined by

$$M_L(c) = \max_{x_1, \ldots, x_T} |\text{mistakes}(L, c)|.$$

- Definition: for any concept class C the maximum number of mistakes a learning algorithm L makes is

$$M_L(C) = \max_{c \in C} M_L(c).$$

A mistake bound is a bound M on $M_L(C)$.

Halving Algorithm

see (Mitchell, 1997)

$\textsc{Halving}(H)$

$H_1 \leftarrow H$
for $t \leftarrow 1$ **to** T **do**
    $\textsc{Receive}(x_t)$
    $\widehat{y}_t \leftarrow \textsc{MajorityVote}(H_t, x_t)$
    $\textsc{Receive}(y_t)$
    if $\widehat{y}_t \neq y_t$ **then**
        $H_{t+1} \leftarrow \{c \in H_t \colon c(x_t) = y_t\}$
return H_{T+1}

Halving Algorithm - Bound

(Littlestone, 1988)

- Theorem: Let H be a finite hypothesis set, then

$$M_{Halving(H)} \leq \log_2 |H|.$$

- Proof: At each mistake, the hypothesis set is reduced at least by half.

VC Dimension Lower Bound

(Littlestone, 1988)

- Theorem: Let $\text{opt}(H)$ be the optimal mistake bound for H. Then,

$$\text{VCdim}(H) \leq \text{opt}(H) \leq M_{Halving(H)} \leq \log_2 |H|.$$

- Proof: for a fully shattered set, form a complete binary tree of the mistakes with height $\text{VCdim}(H)$.

Weighted Majority Algorithm

(Littlestone and Warmuth, 1988)

WEIGHTED-MAJORITY(N experts) $\triangleright$ $y_t, y_{t,i} \in \{0, 1\}$.
$\beta \in [0, 1)$.

for $i \leftarrow 1$ **to** N **do**
    $w_{1,i} \leftarrow 1$
for $t \leftarrow 1$ **to** T **do**
    RECEIVE(x_t)
    $\widehat{y}_t \leftarrow 1_{\sum_{y_{t,i}=1}^{N} w_t \geq \sum_{y_{t,i}=0}^{N} w_t}$ $\triangleright$ weighted majority vot
    RECEIVE(y_t)
    if $\widehat{y}_t \neq y_t$ **then**
        for $i \leftarrow 1$ **to** N **do**
            if $(y_{t,i} \neq y_t)$ **then**
                $w_{t+1,i} \leftarrow \beta w_{t,i}$
            else $w_{t+1,i} \leftarrow w_{t,i}$
return $\mathbf{w}_{T+1}$

Weighted Majority - Bound

- Theorem: Let m_t be the number of mistakes made by the WM algorithm till time t and m_t^* that of the best expert. Then, for all t,

$$\boxed{m_t \leq \frac{\log N + m_t^* \log \frac{1}{\beta}}{\log \frac{2}{1+\beta}}.}$$

- Thus, $m_t \leq O(\log N) + \text{constant} \times \text{best expert}.$
- Realizable case: $m_t \leq O(\log N).$
- Halving algorithm: $\beta = 0$.

Weighted Majority - Proof

- Potential: $\Phi_t = \sum_{i=1}^N w_{t,i}$.
- Upper bound: after each error,
$$\Phi_{t+1} \leq \left[\tfrac{1}{2} + \tfrac{1}{2} \times \beta\right] \Phi_t = \left[\frac{1+\beta}{2}\right] \Phi_t.$$
Thus, $\Phi_t \leq \left[\frac{1+\beta}{2}\right]^{m_t} N.$
- Lower bound: for any expert i, $\Phi_t \geq w_{t,i} = \beta^{m_{t,i}}$.
- Comparison: $\beta^{m_t^*} \leq \left[\frac{1+\beta}{2}\right]^{m_t} N$
$$\Rightarrow m_t^* \log \beta \leq \log N + m_t \log \left[\frac{1+\beta}{2}\right]$$
$$\Rightarrow m_t \log \left[\frac{2}{1+\beta}\right] \leq \log N + m_t^* \log \frac{1}{\beta}.$$

Weighted Majority - Notes

- Advantage: remarkable bound requiring no assumption.
- Disadvantage: no deterministic algorithm can achieve a regret $R_T = o(T)$ with the binary loss.
  - better guarantee with randomized WM.
  - better guarantee for WM with convex losses.

Exponential Weighted Average

- **Algorithm:**
  - weight update: $w_{t+1,i} \leftarrow w_{t,i}\, e^{-\eta L(y_{t,i}, y_t)} = e^{-\eta L_{t,i}}$.
  - prediction: $\widehat{y}_t = \frac{\sum_{i=1}^N w_{t,i} y_{t,i}}{\sum_{i=1}^N w_{t,i}}$.

total loss incurred by expert i up to time t

- **Theorem:** assume that L is convex in its first argument and takes values in $[0, 1]$. Then, for any $\eta > 0$ and any sequence $y_1, \ldots, y_T \in Y$, the regret at T satisfies

$$\text{Regret}(T) \leq \frac{\log N}{\eta} + \frac{\eta T}{8}.$$

For $\eta = \sqrt{8 \log N / T}$,

$$\boxed{\text{Regret}(T) \leq \sqrt{(T/2) \log N}}.$$

Exponential Weighted Avg - Proof

- Potential: $\Phi_t = \log \sum_{i=1}^N w_{t,i}$.
- Upper bound:

$$
\begin{aligned}
\Phi_t - \Phi_{t-1} &= \log \frac{\sum_{i=1}^N w_{t-1,i}\, e^{-\eta L(y_{t,i}, y_t)}}{\sum_{i=1}^N w_{t-1,i}} \\
&= \log \Big(\mathop{\mathrm{E}}_{w_{t-1}} [e^{-\eta L(y_{t,i}, y_t)}] \Big) \\
&= \log \left(\mathop{\mathrm{E}}_{w_{t-1}} \left[\exp\left(-\eta \Big(L(y_{t,i}, y_t) - \mathop{\mathrm{E}}_{w_{t-1}}[L(y_{t,i}, y_t)] \Big) - \eta \mathop{\mathrm{E}}_{w_{t-1}}[L(y_{t,i}, y_t)] \right) \right] \right) \\
&\leq -\eta \mathop{\mathrm{E}}_{w_{t-1}}[L(y_{t,i}, y_t)] + \frac{\eta^2}{8} \quad \text{(Hoeffding's ineq.)} \\
&\leq -\eta L(\mathop{\mathrm{E}}_{w_{t-1}}[y_{t,i}], y_t) + \frac{\eta^2}{8} \quad \text{(convexity of first arg. of } L) \\
&= -\eta L(\widehat{y}_t, y_t) + \frac{\eta^2}{8}.
\end{aligned}
$$

Exponential Weighted Avg - Proof

- **Upper bound:** summing up the inequalities yields

$$\Phi_T - \Phi_0 \leq -\eta \sum_{t=1}^{T} L(\widehat{y}_t, y_t) + \frac{\eta^2 T}{8}.$$

- **Lower bound:**

$$\Phi_T - \Phi_0 = \log \sum_{i=1}^{N} e^{-\eta L_{T,i}} - \log N \geq \log \max_{i=1}^{N} e^{-\eta L_{T,i}} - \log N$$

$$= -\eta \min_{i=1}^{N} L_{T,i} - \log N.$$

- **Comparison:**

$$-\eta \min_{i=1}^{N} L_{T,i} - \log N \leq -\eta \sum_{t=1}^{T} L(\widehat{y}_t, y_t) + \frac{\eta^2 T}{8}$$

$$\Rightarrow \sum_{t=1}^{T} L(\widehat{y}_t, y_t) - \min_{i=1}^{N} L_{T,i} \leq \frac{\log N}{\eta} + \frac{\eta T}{8}.$$

Exponential Weighted Avg - Notes

- Advantage: bound on regret per bound is of the form $\frac{R_T}{T} = O\Big(\sqrt{\frac{\log(N)}{T}}\Big)$.
- Disadvantage: choice of η requires knowledge of horizon T.

Doubling Trick

- Idea: divide time into periods $[2^k, 2^{k+1}-1]$ of length 2^k with $k=0, \ldots, n$, $T \geq 2^n - 1$, and choose $\eta_k = \sqrt{\frac{8 \log N}{2^k}}$ in each period.
- Theorem: with the same assumptions as before, for any T, the following holds:

$$\text{Regret}(T) \leq \frac{\sqrt{2}}{\sqrt{2}-1} \sqrt{(T/2) \log N} + \sqrt{\log N/2}.$$

Doubling Trick - Proof

- By the previous theorem, for any $I_k = [2^k, 2^{k+1} - 1]$,

$$L_{I_k} - \min_{i=1}^{N} L_{I_k,i} \leq \sqrt{2^k/2 \, \log N}.$$

Thus, $$L_T = \sum_{k=0}^{n} L_{I_k} \leq \sum_{k=0}^{n} \min_{i=1}^{N} L_{I_k,i} + \sum_{k=0}^{n} \sqrt{2^k \, (\log N)/2}$$

$$\leq \min_{i=1}^{N} L_{T,i} + \sum_{k=0}^{n} 2^{\frac{k}{2}} \sqrt{(\log N)/2}.$$

with

$$\sum_{i=0}^{n} 2^{\frac{k}{2}} = \frac{\sqrt{2}^{n+1} - 1}{\sqrt{2} - 1} = \frac{2^{(n+1)/2} - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2}\sqrt{T+1} - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2}(\sqrt{T} + 1) - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2}\sqrt{T}}{\sqrt{2} - 1} + 1.$$

Notes

- Doubling trick used in a variety of other contexts and proofs.
- More general method, learning parameter function of time: $\eta_t = \sqrt{(8 \log N)/t}$. Constant factor improvement:

$$\text{Regret}(T) \leq 2\sqrt{(T/2) \log N} + \sqrt{(1/8) \log N}.$$

This Lecture

- Prediction with expert advice
- Linear classification

Perceptron Algorithm

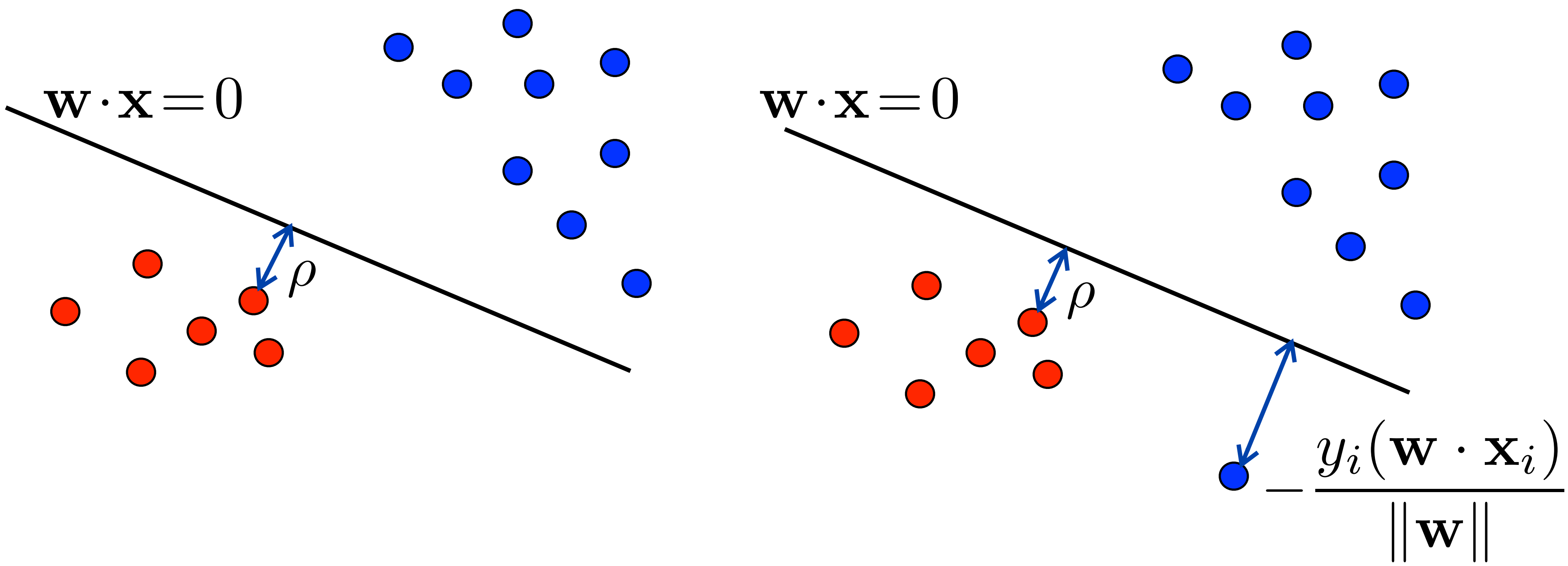
(Rosenblatt, 1958)

$\textsc{Perceptron}(\mathbf{w}_0)$

$\mathbf{w}_1 \leftarrow \mathbf{w}_0$ $\quad \triangleright$ typically $\mathbf{w}_0 = \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
$\quad \textsc{Receive}(\mathbf{x}_t)$
$\quad \widehat{y}_t \leftarrow \mathrm{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$
$\quad \textsc{Receive}(y_t)$
$\quad$ **if** $(\widehat{y}_t \neq y_t)$ **then**
$\quad\quad \mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_t \mathbf{x}_t$ $\quad \triangleright$ more generally $\eta y_t \mathbf{x}_t, \eta > 0$
$\quad$ **else** $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$
return $\mathbf{w}_{T+1}$

Separating Hyperplane

- Margin and errors



Perceptron = Stochastic Gradient Descent

- Objective function: convex but not differentiable.

$$F(\mathbf{w}) = \frac{1}{T} \sum_{t=1}^{T} \max\Big(0, -y_t(\mathbf{w} \cdot \mathbf{x}_t)\Big) = \mathop{\mathrm{E}}_{\mathbf{x} \sim \widehat{D}} [f(\mathbf{w}, \mathbf{x})]$$

with $f(\mathbf{w}, \mathbf{x}) = \max\big(0, -y(\mathbf{w} \cdot \mathbf{x})\big)$.

- Stochastic gradient: for each $\mathbf{x}_t$, the update is

$$\mathbf{w}_{t+1} \leftarrow \begin{cases} \mathbf{w}_t - \eta \nabla_{\mathbf{w}} f(\mathbf{w}_t, \mathbf{x}_t) & \text{if differentiable} \\ \mathbf{w}_t & \text{otherwise,} \end{cases}$$

where $\eta > 0$ is a learning rate parameter.

- Here: $$\mathbf{w}_{t+1} \leftarrow \begin{cases} \mathbf{w}_t + \eta y_t \mathbf{x}_t & \text{if } y_t(\mathbf{w}_t \cdot \mathbf{x}_t) < 0 \\ \mathbf{w}_t & \text{otherwise.} \end{cases}$$

Perceptron Algorithm - Bound

(Novikoff, 1962)

- Theorem: Assume that $\|x_t\| \leq R$ for all $t \in [1, T]$ and that for some $\rho > 0$ and $\mathbf{v} \in \mathbb{R}^N$, for all $t \in [1, T]$,

$$\rho \leq \frac{y_t(\mathbf{v} \cdot \mathbf{x}_t)}{\|\mathbf{v}\|}.$$

  Then, the number of mistakes made by the perceptron algorithm is bounded by $\boxed{R^2/\rho^2}$.

- Proof: Let I be the set of ts at which there is an update and let M be the total number of updates.

- Summing up the assumption inequalities gives:

$$
\begin{aligned}
M\rho &\leq \frac{\mathbf{v} \cdot \sum_{t \in I} y_t \mathbf{x}_t}{\|\mathbf{v}\|} \\
&= \frac{\mathbf{v} \cdot \sum_{t \in I} (\mathbf{w}_{t+1} - \mathbf{w}_t)}{\|\mathbf{v}\|} && \text{(definition of updates)} \\
&= \frac{\mathbf{v} \cdot \mathbf{w}_{T+1}}{\|\mathbf{v}\|} \\
&\leq \|\mathbf{w}_{T+1}\| && \text{(Cauchy-Schwarz ineq.)} \\
&= \|\mathbf{w}_{t_m} + y_{t_m} \mathbf{x}_{t_m}\| && (t_m \text{ largest } t \text{ in } I) \\
&= \Big[\|\mathbf{w}_{t_m}\|^2 + \|\mathbf{x}_{t_m}\|^2 + 2 \underbrace{y_{t_m} \mathbf{w}_{t_m} \cdot \mathbf{x}_{t_m}}_{\leq 0} \Big]^{1/2} \\
&\leq \Big[\|\mathbf{w}_{t_m}\|^2 + R^2 \Big]^{1/2} \\
&\leq \Big[M R^2 \Big]^{1/2} = \sqrt{M} R. && \text{(applying the same to previous } t\text{s in } I)
\end{aligned}
$$

- Notes:
  - bound independent of dimension and tight.
  - convergence can be slow for small margin, it can be in $\Omega(2^N)$.
  - among the many variants: voted perceptron algorithm. Predict according to

$$\operatorname{sign}\Big(\big(\sum_{t \in I} c_t \mathbf{w}_t\big) \cdot \mathbf{x} \Big),$$

    where c_t is the number of iterations $\mathbf{w}_t$ survives.
  - $\{x_t : t \in I\}$ are the support vectors for the perceptron algorithm.
  - non-separable case: does not converge.

Perceptron - Leave-One-Out Analysis

- Theorem: Let h_S be the hypothesis returned by the perceptron algorithm for sample $S=(x_1, \ldots, x_T) \sim D$ and let $M(S)$ be the number of updates defining h_S. Then,

$$\underset{S \sim D^m}{\mathrm{E}}[R(h_S)] \leq \underset{S \sim D^{m+1}}{\mathrm{E}}\left[\frac{\min(M(S), R_{m+1}^2/\rho_{m+1}^2)}{m+1}\right].$$

- Proof: Let $S \sim D^{m+1}$ be a sample linearly separable and let $\mathbf{x} \in S$. If $h_{S-\{\mathbf{x}\}}$ misclassifies $\mathbf{x}$, then $\mathbf{x}$ must be a 'support vector' for h_S (update at $\mathbf{x}$). Thus,

$$\widehat{R}_{\text{loo}}(\text{perceptron}) \leq \frac{M(S)}{m+1}.$$

Perceptron - Non-Separable Bound

(MM and Rostamizadeh, 2013)

- Theorem: let I denote the set of rounds at which the Perceptron algorithm makes an update when processing $\mathbf{x}_1, \ldots, \mathbf{x}_T$ and let $M_T = |I|$. Then,

$$M_T \leq \inf_{\rho > 0, \|\mathbf{u}\|_2 \leq 1} \left[\sqrt{L_\rho(\mathbf{u})} + \frac{R}{\rho} \right]^2,$$

where $R = \max_{t \in I} \|\mathbf{x}_t\|$

$$L_\rho(\mathbf{u}) = \sum_{t \in I} \left(1 - \frac{y_t(\mathbf{u} \cdot \mathbf{x}_t)}{\rho}\right)_+.$$

- Proof: for any t, $1 - \frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho} \leq \left(1 - \frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho}\right)_+$, summing up these inequalities for $t \in I$ yields:

$$
\begin{aligned}
M_T &\leq \sum_{t \in I} \left(1 - \frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho}\right)_+ + \sum_{t \in I} \frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho} \\
&\leq L_\rho(\mathbf{u}) + \frac{\sqrt{M_T}R}{\rho},
\end{aligned}
$$

by upper-bounding $\sum_{t \in I}(y_t \mathbf{u}\cdot\mathbf{x}_t)$ as in the proof for the separable case.

- solving the second-degree inequality

$$
M_T \leq L_\rho(\mathbf{u}) + \frac{\sqrt{M_T}R}{\rho},
$$

gives $\sqrt{M_T} \leq \frac{\frac{R}{\rho} + \sqrt{\frac{R^2}{\rho^2} + 4L_\rho(\mathbf{u})}}{2} \leq \frac{R}{\rho} + \sqrt{L_\rho(\mathbf{u})}.$

Non-Separable Case - L2 Bound

(Freund and Schapire, 1998; MM and Rostamizadeh, 2013)

- Theorem: let I denote the set of rounds at which the Perceptron algorithm makes an update when processing $\mathbf{x}_1, \ldots, \mathbf{x}_T$ and let $M_T = |I|$. Then,

$$M_T \leq \inf_{\rho>0, \|u\|_2 \leq 1} \left[\frac{\|\mathbf{L}_\rho(\mathbf{u})\|_2}{2} + \sqrt{\frac{\|\mathbf{L}_\rho(\mathbf{u})\|_2^2}{4} + \frac{\sqrt{\sum_{t \in I} \|\mathbf{x}_t\|^2}}{\rho}} \right]^2.$$

  - when $\|\mathbf{x}_t\| \leq R$ for all $t \in I$, this implies

$$M_T \leq \inf_{\rho>0, \|u\|_2 \leq 1} \left(\frac{R}{\rho} + \|\mathbf{L}_\rho(\mathbf{u})\|_2 \right)^2,$$

where $\mathbf{L}_\rho(\mathbf{u}) = \left[\left(1 - \frac{y_t(\mathbf{u} \cdot \mathbf{x}_t)}{\rho}\right)_+ \right]_{t \in I}$.

- Proof: Reduce problem to separable case in higher dimension. Let $l_t = \left(1 - \frac{y_t \mathbf{u} \cdot \mathbf{x}_t}{\rho}\right)_+ 1_{t \in I}$, for $t \in [1, T]$.
  - Mapping (similar to trivial mapping):

$(N+t)$th component

$$\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \end{bmatrix} \rightarrow \mathbf{x}'_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \\ 0 \\ \vdots \\ 0 \\ \Delta \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \mathbf{u} \rightarrow \mathbf{u}' = \begin{bmatrix} \frac{u_1}{Z} \\ \vdots \\ \frac{u_N}{Z} \\ \frac{y_1 \rho l_1}{\Delta Z} \\ \vdots \\ \frac{y_T \rho l_T}{\Delta Z} \end{bmatrix}$$

$$\|\mathbf{u}'\| = 1 \implies Z = \sqrt{1 + \frac{\rho^2 \|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2}}.$$

- Observe that the Perceptron algorithm makes the same predictions and makes updates at the same rounds when processing $\mathbf{x}'_1, \ldots, \mathbf{x}'_T$.
- For any $t \in I$,

$$
\begin{aligned}
y_t(\mathbf{u}' \cdot \mathbf{x}'_t) &= y_t\Big(\frac{\mathbf{u} \cdot \mathbf{x}_t}{Z} + \Delta \frac{y_t \rho l_t}{Z\Delta}\Big) \\
&= \frac{y_t \mathbf{u} \cdot \mathbf{x}_t}{Z} + \frac{\rho l_t}{Z} \\
&= \frac{1}{Z}\big(y_t \mathbf{u} \cdot \mathbf{x}_t + [\rho - y_t(\mathbf{u} \cdot \mathbf{x}_t)]_+\big) \geq \frac{\rho}{Z}.
\end{aligned}
$$

- Summing up and using the proof in the separable case yields:

$$
M_T \frac{\rho}{Z} \leq \sum_{t \in I} y_t(\mathbf{u}' \cdot \mathbf{x}'_t) \leq \sqrt{\sum_{t \in I} \|\mathbf{x}'_t\|^2}.
$$

- The inequality can be rewritten as

$$M_T^2 \leq \Big(\frac{1}{\rho^2}+\frac{\|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2}\Big)\Big(r^2+M_T\Delta^2\Big)=\frac{r^2}{\rho^2}+\frac{r^2\|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2}+\frac{M_T\Delta^2}{\rho^2}+M_T\|\,\mathbf{L}_\rho(\mathbf{u})\|^2;$$

where $r = \sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}$.

- Selecting Δ to minimize the bound gives $\Delta^2 = \frac{\rho\|\mathbf{L}_\rho(\mathbf{u})\|_2 r}{\sqrt{M_T}}$ and leads to

$$M_T^2 \leq \frac{r^2}{\rho^2} + 2\frac{\sqrt{M_T}\|\mathbf{L}_\rho(\mathbf{u})\|r}{\rho} + M_T\|\mathbf{L}_\rho(\mathbf{u})\|^2 = (\tfrac{r}{\rho} + \sqrt{M_T}\|\mathbf{L}_\rho(\mathbf{u})\|_2)^2.$$

- Solving the second-degree inequality

$$M_T - \sqrt{M_T}\|\,\mathbf{L}_\rho(\mathbf{u})\|_2 - \frac{r}{\rho} \leq 0$$

yields directly the first statement. The second one results from replacing r with $\sqrt{M_T}R$.

Dual Perceptron Algorithm

$\textsc{Dual-Perceptron}(\boldsymbol{\alpha}^0)$

$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}^0 \qquad \triangleright$ typically $\boldsymbol{\alpha}^0 = \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
  $\textsc{Receive}(\mathbf{x}_t)$
  $\widehat{y}_t \leftarrow \operatorname{sgn}(\sum_{s=1}^{T} \alpha_s y_s (\mathbf{x}_s \cdot \mathbf{x}_t))$
  $\textsc{Receive}(y_t)$
  if $(\widehat{y}_t \neq y_t)$ **then**
    $\alpha_t \leftarrow \alpha_t + 1$
return $\boldsymbol{\alpha}$

Kernel Perceptron Algorithm

(Aizerman et al., 1964)

K PDS kernel.

$\textsc{Kernel-Perceptron}(\boldsymbol{\alpha}^0)$

$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}^0 \qquad \triangleright$ typically $\boldsymbol{\alpha}^0 = \mathbf{0}$

for $t \leftarrow 1$ **to** T **do**

$\quad \textsc{Receive}(x_t)$

$\quad \widehat{y}_t \leftarrow \operatorname{sgn}(\sum_{s=1}^{T} \alpha_s y_s K(x_s, x_t))$

$\quad \textsc{Receive}(y_t)$

$\quad$ **if** $(\widehat{y}_t \neq y_t)$ **then**

$\qquad \alpha_t \leftarrow \alpha_t + 1$

return $\boldsymbol{\alpha}$

Winnow Algorithm

(Littlestone, 1988)

$\textsc{Winnow}(\eta)$

$w_1 \leftarrow \mathbf{1}/N$

for $t \leftarrow 1$ **to** T **do**

  $\textsc{Receive}(\mathbf{x}_t)$

  $\widehat{y}_t \leftarrow \text{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$ $\triangleright \quad y_t \in \{-1, +1\}$

  $\textsc{Receive}(y_t)$

  if $(\widehat{y}_t \neq y_t)$ **then**

    $Z_t \leftarrow \sum_{i=1}^{N} w_{t,i} \exp(\eta y_t x_{t,i})$

    for $i \leftarrow 1$ **to** N **do**

      $w_{t+1,i} \leftarrow \frac{w_{t,i} \exp(\eta y_t x_{t,i})}{Z_t}$

  else $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$

return $\mathbf{w}_{T+1}$

Notes

- Winnow $=$ weighted majority:
  - for $y_{t,i} = x_{t,i} \in \{-1, +1\}$, $\text{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$ coincides with the majority vote.
  - multiplying by e^{η} or $e^{-\eta}$ the weight of correct or incorrect experts, is equivalent to multiplying by $\beta = e^{-2\eta}$ the weight of incorrect ones.
- Relationships with other algorithms: e.g., boosting and Perceptron (Winnow and Perceptron can be viewed as special instances of a general family).

Winnow Algorithm - Bound

- **Theorem**: Assume that $\|x_t\|_\infty \leq R_\infty$ for all $t \in [1, T]$ and that for some $\rho_\infty > 0$ and $\mathbf{v} \in \mathbb{R}^N$, $\mathbf{v} \geq 0$ for all $t \in [1, T]$,

$$\rho_\infty \leq \frac{y_t(\mathbf{v} \cdot \mathbf{x}_t)}{\|\mathbf{v}\|_1}.$$

  Then, the number of mistakes made by the Winnow algorithm is bounded by $\boxed{2\,(R_\infty^2/\rho_\infty^2)\log N}$.

- **Proof**: Let I be the set of ts at which there is an update and let M be the total number of updates.

Notes

- Comparison with perceptron bound:
  - dual norms: norms for $\mathbf{x}_t$ and $\mathbf{v}$.
  - similar bounds with different norms.
  - each advantageous in different cases:
    - Winnow bound favorable when a sparse set of experts can predict well. For example, if $\mathbf{v} = \mathbf{e}_1$ and $\mathbf{x}_t \in \{\pm 1\}^N$, $\log N$ vs N.
    - Perceptron favorable in opposite situation.

Winnow Algorithm - Bound

- **Potential:** $\Phi_t = \sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|} \log \frac{v_i/\|\mathbf{v}\|}{w_{t,i}}$. (relative entropy)
- **Upper bound:** for each t in I,

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= \textstyle\sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{w_{t,i}}{w_{t+1,i}} \\
&= \textstyle\sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{Z_t}{\exp(\eta y_t x_{t,i})} \\
&= \log Z_t - \eta \textstyle\sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|_1} y_t x_{t,i} \\
&\leq \log \big[\textstyle\sum_{i=1}^{N} w_{t,i} \exp(\eta y_t x_{t,i}) \big] - \eta \rho_\infty \\
&= \log \operatorname*{E}_{\mathbf{w}_t} \big[\exp(\eta y_t x_t) \big] - \eta \rho_\infty \\
\text{(Hoeffding)} &\leq \log \big[\exp(\eta^2 (2R_\infty)^2/8) \big] + \underbrace{\eta y_t \mathbf{w}_t \cdot \mathbf{x}_t}_{\leq 0} - \eta \rho_\infty \\
&\leq \eta^2 R_\infty^2/2 - \eta \rho_\infty .
\end{aligned}$$

Winnow Algorithm - Bound

- Upper bound: summing up the inequalities yields

$$\Phi_{T+1} - \Phi_1 \leq M(\eta^2 R_\infty^2/2 - \eta\rho_\infty).$$

- Lower bound: note that

$$\Phi_1 = \sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{v_i/\|\mathbf{v}\|_1}{1/N} = \log N + \sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{v_i}{\|\mathbf{v}\|_1} \leq \log N$$

and for all $t, \Phi_t \geq 0$ (property of relative entropy).

Thus, $\Phi_{T+1} - \Phi_1 \geq 0 - \log N = -\log N.$

- Comparison: $-\log N \leq M(\eta^2 R_\infty^2/2 - \eta\rho_\infty)$. For $\eta = \frac{\rho_\infty}{R_\infty^2}$ we obtain

$$M \leq 2 \log N \frac{R_\infty^2}{\rho_\infty^2}.$$

Conclusion

- On-line learning:
  - wide and fast-growing literature.
  - many related topics, e.g., game theory, text compression, convex optimization.
  - online to batch bounds and techniques.
  - online version of batch algorithms, e.g., regression algorithms (see regression lecture).

References

- Aizerman, M. A., Braverman, E. M., & Rozonoer, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control, 25*, 821-837.

- Nicolò Cesa-Bianchi, Alex Conconi, Claudio Gentile: On the Generalization Ability of On-Line Learning Algorithms. *IEEE Transactions on Information Theory* 50(9): 2050-2057. 2004.

- Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

- Yoav Freund and Robert Schapire. Large margin classification using the perceptron algorithm. In *Proceedings of COLT 1998*. ACM Press, 1998.

- Nick Littlestone. From On-Line to Batch Learning. COLT 1989: 269-284.

- Nick Littlestone. "Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm" *Machine Learning* 285-318(2). 1988.

References

- Nick Littlestone, Manfred K. Warmuth: The Weighted Majority Algorithm. *FOCS 1989*: 256-261.
- Tom Mitchell. *Machine Learning*, McGraw Hill, 1997.
- Mehryar Mohri and Afshin Rostamizadeh. Perceptron Mistake Bounds. arXiv:1305.0208, 2013.
- Novikoff, A. B. (1962). On convergence proofs on perceptrons. *Symposium on the Mathematical Theory of Automata, 12*, 615-622. Polytechnic Institute of Brooklyn.
- Rosenblatt, Frank, *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386-408, 1958.
- Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

Appendix

SVMs - Leave-One-Out Analysis

(Vapnik, 1995)

- Theorem: let h_S be the optimal hyperplane for a sample S and let $N_{\mathrm{SV}}(S)$ be the number of support vectors defining h_S. Then,

$$\operatorname*{E}_{S \sim D^m}[R(h_S)] \leq \operatorname*{E}_{S \sim D^{m+1}}\left[\frac{\min(N_{\mathrm{SV}}(S), R_{m+1}^2/\rho_{m+1}^2)}{m+1}\right].$$

- Proof: one part proven in lecture 4. The other part due to $\alpha_i \geq 1/R_{m+1}^2$ for $\mathbf{x}_i$ misclassified by SVMs.

Comparison

- Bounds on expected error, not high probability statements.
- Leave-one-out bounds not sufficient to distinguish SVMs and perceptron algorithm. Note however:
  - same maximum margin ρ_{m+1} can be used in both.
  - but different radius R_{m+1} of support vectors.
- Difference: margin distribution.