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 $\hat{y}_t \in Y$.
  - receive label $y_t \in Y$.
  - incur loss $L(\hat{y}_t, y_t)$.

- **Classification:** $Y = \{0, 1\}$, $L(y, y') = |y' - y|$.
- **Regression:** $Y \subseteq \mathbb{R}$, $L(y, y') = (y' - y)^2$.
- **Objective:** minimize total loss $\sum_{t=1}^{T} L(\hat{y}_t, y_t)$. 
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 \( \hat{y}_t \in Y \).
  - receive label \( y_t \in Y \).
  - incur loss \( L(\hat{y}_t, y_t) \).

- **Objective**: minimize regret, i.e., difference of total loss incurred and that of best expert.

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

**HALVING**($H$)

1. $H_1 \leftarrow H$
2. **for** $t \leftarrow 1$ **to** $T$ **do**
3. \hspace{1em} RECEIVE($x_t$)
4. \hspace{1em} $\hat{y}_t \leftarrow \text{MAJORITYVOTE}(H_t, x_t)$
5. \hspace{1em} RECEIVE($y_t$)
6. \hspace{1em} **if** $\hat{y}_t \neq y_t$ **then**
7. \hspace{2em} $H_{t+1} \leftarrow \{c \in H_t : c(x_t) = y_t\}$
8. **return** $H_{T+1}$
Halving Algorithm - Bound

(Littlestone, 1988)

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

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

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,

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

- **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) \( \Rightarrow \ y_t, y_{t,i} \in \{0, 1\} \). \( \beta \in [0, 1) \).

1 \hspace{1em} \textbf{for} \ i \leftarrow 1 \hspace{1em} \textbf{to} \ N \hspace{1em} \textbf{do} \\
2 \hspace{1em} \hspace{1em} w_{1,i} \leftarrow 1 \\
3 \hspace{1em} \textbf{for} \ t \leftarrow 1 \hspace{1em} \textbf{to} \ T \hspace{1em} \textbf{do} \\
4 \hspace{1em} \hspace{1em} \textbf{RECEIVE}(x_t) \\
5 \hspace{1em} \hspace{1em} \hat{y}_t \leftarrow 1 \sum_{y_{t,i}=1}^N w_t \geq \sum_{y_{t,i}=0}^N w_t \hspace{1em} \Rightarrow \text{weighted majority vote} \\
6 \hspace{1em} \textbf{RECEIVE}(y_t) \\
7 \hspace{1em} \textbf{if} \ \hat{y}_t \neq y_t \textbf{ then} \\
8 \hspace{1em} \hspace{1em} \textbf{for} \ i \leftarrow 1 \hspace{1em} \textbf{to} \ N \hspace{1em} \textbf{do} \\
9 \hspace{1em} \hspace{1em} \hspace{1em} \textbf{if} \ (y_{t,i} \neq y_t) \textbf{ then} \\
10 \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} w_{t+1,i} \leftarrow \beta w_{t,i} \\
11 \hspace{1em} \hspace{1em} \hspace{1em} \textbf{else} \ w_{t+1,i} \leftarrow w_{t,i} \\
12 \hspace{1em} \textbf{return} \ w_{T+1}
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 $t$,

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

• 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,
$$\Phi_{t+1} \leq \left[ \frac{1}{2} + \frac{1}{2} \times \beta \right] \Phi_t = \left[ \frac{1 + \beta}{2} \right] \Phi_t.$$  
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: $\beta^{m^*_t} \leq \left[ \frac{1+\beta}{2} \right]^{m_t} N$

$$\Rightarrow m^*_t \log \beta \leq \log N + m_t \log \left[ \frac{1+\beta}{2} \right]$$

$$\Rightarrow m_t \log \left[ \frac{2}{1+\beta} \right] \leq \log N + m^*_t \log \frac{1}{\beta}.$$

Mehryar Mohri - Foundations of Machine Learning
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 loss.

$RT = o(T)$
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: $\hat{y}_t = \frac{\sum_{i=1}^{N} w_{t,i} y_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$.

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

$$\text{Regret}(T) \leq \sqrt{(T/2) \log N}.$$
Exponential Weighted Avg - Proof

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

- **Upper bound:**

$$
\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 \left( \mathbb{E}_{w_{t-1}} \left[ e^{-\eta L(y_{t,i}, y_t)} \right] \right)
= \log \left( \mathbb{E}_{w_{t-1}} \left[ \exp \left( -\eta \left( L(y_{t,i}, y_t) - \mathbb{E}_{w_{t-1}} [L(y_{t,i}, y_t)] \right) \right) - \eta \mathbb{E}_{w_{t-1}} [L(y_{t,i}, y_t)] \right) \right)
\leq -\eta \mathbb{E}_{w_{t-1}} [L(y_{t,i}, y_t)] + \frac{\eta^2}{8} \quad \text{(Hoeffding’s ineq.)}
\leq -\eta L \left( \mathbb{E}_{w_{t-1}} [y_{t,i}], y_t \right) + \frac{\eta^2}{8} \quad \text{(convexity of first arg. of } L) \)
= -\eta L(\hat{y}_t, y_t) + \frac{\eta^2}{8}.
$$
Exponential Weighted Avg - Proof

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

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

- **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:**

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

\[
\implies \sum_{t=1}^{T} L(\hat{y}_t, y_t) - \min_{i=1}^{N} L_{T,i} \leq \frac{\log N}{\eta} + \frac{\eta T}{8}.
\]
Advantage: bound on regret per bound is of the form $\frac{R_T}{T} = O\left(\sqrt{\frac{\log(N)}{T}}\right)$.

Disadvantage: choice of $\eta$ 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,

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

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

with

\[
\sum_{i=0}^{n} 2^{\frac{k}{2}} = \frac{\sqrt{2^{n+1}} - 1}{\sqrt{2} - 1} = \frac{2^{(n+1)/2} - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2\sqrt{T} + 1} - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2} (\sqrt{T} + 1) - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2\sqrt{T} + 1}}{\sqrt{2} - 1} + 1.
\]
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}(w_0)
1. \( w_1 \leftarrow w_0 \quad \triangleright \text{typically } w_0 = 0 \)
2. \textbf{for} \( t \leftarrow 1 \) \textbf{to} \( T \) \textbf{do}
3. \hspace{1em} \textbf{Receive}(x_t)
4. \hspace{1em} \hat{y}_t \leftarrow \text{sgn}(w_t \cdot x_t)
5. \hspace{1em} \textbf{Receive}(y_t)
6. \hspace{1em} \textbf{if} \ (\hat{y}_t \neq y_t) \ \textbf{then}
7. \hspace{2em} w_{t+1} \leftarrow w_t + y_t x_t \quad \triangleright \text{more generally } \eta y_t x_t, \eta > 0
8. \hspace{1em} \textbf{else} \ w_{t+1} \leftarrow w_t
9. \hspace{1em} \textbf{return} \ w_{T+1}
Separating Hyperplane

Margin and errors

\[ w \cdot x = 0 \]

\[ y_i \left( w \cdot x_i \right) - \frac{y_i (w \cdot x_i)}{||w||} \]
Perceptron $\equiv$ Stochastic Gradient Descent

- **Objective function**: convex but not differentiable.
  
  \[
  F(w) = \frac{1}{T} \sum_{t=1}^{T} \max \left( 0, -y_t(w \cdot x_t) \right) = \mathbb{E}_{x \sim \hat{D}} [f(w, x)]
  \]

  with \( f(w, x) = \max \left( 0, -y(w \cdot x) \right) \).

- **Stochastic gradient**: for each \( x_t \), the update is
  
  \[
  w_{t+1} \leftarrow \begin{cases} 
  w_t - \eta \nabla_w f(w_t, x_t) & \text{if differentiable} \\
  w_t & \text{otherwise,}
  \end{cases}
  \]

  where \( \eta > 0 \) is a learning rate parameter.

- **Here**:
  
  \[
  w_{t+1} \leftarrow \begin{cases} 
  w_t + \eta y_t x_t & \text{if } y_t(w_t \cdot x_t) < 0 \\
  w_t & \text{otherwise.}
  \end{cases}
  \]
Theorem: Assume that $\|x_t\| \leq R$ for all $t \in [1, T]$ and that for some $\rho > 0$ and $v \in \mathbb{R}^N$, for all $t \in [1, T]$,

$$\rho \leq \frac{y_t (v \cdot x_t)}{\|v\|}.$$

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

Proof: Let $I$ be the set of $t$s at which there is an update and let $M$ be the total number of updates.
Summing up the assumption inequalities gives:

\[ M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\| v \|} \]

\[ = \frac{v \cdot \sum_{t \in I} (w_{t+1} - w_t)}{\| v \|} \quad \text{(definition of updates)} \]

\[ = \frac{v \cdot w_{T+1}}{\| v \|} \]

\[ \leq \| w_{T+1} \| \quad \text{(Cauchy-Schwarz ineq.)} \]

\[ = \| w_{t_m} + y_{t_m} x_{t_m} \| \quad \text{\(t_m\) largest \(t\) in \(I\)} \]

\[ = \left[ \| w_{t_m} \|^2 + \| x_{t_m} \|^2 + 2 y_{t_m} w_{t_m} \cdot x_{t_m} \right]^{1/2} \]

\[ \leq \left[ \| w_{t_m} \|^2 + R^2 \right]^{1/2} \]

\[ \leq \left[ MR^2 \right]^{1/2} = \sqrt{MR}. \quad \text{(applying the same to previous \(t_s\) in \(I\))} \]
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

$$\text{sign} \left( \sum_{t \in I} c_t w_t \cdot x \right),$$

where $c_t$ is the number of iterations $w_t$ survives.

- $\{x_t : t \in I\}$ are the **support vectors** for the perceptron algorithm.

- non-separable case: **does not converge**.
**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,

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

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

$$
\hat{R}_{1oo}(\text{perceptron}) \leq \frac{M(S)}{m + 1}.
$$
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 \( x_1, \ldots, x_T \) and let \( M_T = |I| \). Then,

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

where \( R = \max_{t \in I} \|x_t\| \)

\[
L_\rho(u) = \sum_{t \in I} \left( 1 - \frac{y_t(u \cdot x_t)}{\rho} \right)_+.
\]
• **Proof:** for any \( t \), \( 1 - \frac{y_t(u \cdot x_t)}{\rho} \leq (1 - \frac{y_t(u \cdot x_t)}{\rho})_+ \), summing up these inequalities for \( t \in I \) yields:

\[
M_T \leq \sum_{t \in I} \left( 1 - \frac{y_t(u \cdot x_t)}{\rho} \right)_+ + \sum_{t \in I} \frac{y_t(u \cdot x_t)}{\rho} \\
\leq L_\rho(u) + \frac{\sqrt{M_T R}}{\rho},
\]

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

• solving the second-degree inequality

\[
M_T \leq L_\rho(u) + \frac{\sqrt{M_T R}}{\rho},
\]

gives

\[
\sqrt{M_T} \leq \frac{R}{\rho} + \sqrt{\frac{R^2}{\rho^2} + 4L_\rho(u)} \leq \frac{R}{\rho} + \sqrt{L_\rho(u)}.
\]
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 $x_1, \ldots, x_T$ and let $M_T = |I|$. Then,

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

- when $\|x_t\| \leq R$ for all $t \in I$, this implies

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

where $L_\rho(u) = \left[ \left( 1 - \frac{y_t(u \cdot x_t)}{\rho} \right) + \right]_{t \in I}$. 
• **Proof:** Reduce problem to separable case in higher dimension. Let 
\[ l_t = \left( 1 - \frac{y_t u \cdot x_t}{\rho} \right)_+ 1_{t \in I}, \text{ for } t \in [1, T]. \]

• **Mapping (similar to trivial mapping):**

\[(N + t)\text{th component}\]

\[
x_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \end{bmatrix} \rightarrow x'_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \\ 0 \\ \Delta \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\]

\[
u \rightarrow u' = \begin{bmatrix} \frac{u_1}{Z} \\ \vdots \\ \frac{u_N}{Z} \\ \frac{y_1 \rho l_1}{\Delta Z} \\ \vdots \\ \frac{y_T \rho l_T}{\Delta Z} \end{bmatrix}
\]

\[\|u'\| = 1 \implies Z = \sqrt{1 + \frac{\rho^2 \|L_\rho(u)\|^2}{\Delta^2}}.\]
• Observe that the Perceptron algorithm makes the same predictions and makes updates at the same rounds when processing \( x'_1, \ldots, x'_T \).

• For any \( t \in I \),

\[
 y_t(u' \cdot x'_t) = y_t \left( \frac{u \cdot x_t}{Z} + \Delta \frac{y_t \rho l_t}{Z \Delta} \right) \\
= \frac{y_t u \cdot x_t}{Z} + \frac{\rho l_t}{Z} \\
= \frac{1}{Z} \left( y_t u \cdot x_t + [\rho - y_t (u \cdot x_t)]_+ \right) \geq \frac{\rho}{Z}.
\]

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

\[
 MT \frac{\rho}{Z} \leq \sum_{t \in I} y_t(u' \cdot x'_t) \leq \sqrt{\sum_{t \in I} \|x'_t\|^2}.
\]
• The inequality can be rewritten as

\[ M_T^2 \leq \left( \frac{1}{\rho^2} + \frac{\|L_\rho(u)\|^2}{\Delta^2} \right) \left( r^2 + M_T \Delta^2 \right) = \frac{r^2}{\rho^2} + \frac{r^2 \|L_\rho(u)\|^2}{\Delta^2} + \frac{M_T \Delta^2}{\rho^2} + M_T \|L_\rho(u)\|^2, \]

where \( r = \sqrt{\sum_{t \in I} \|x_t\|^2}. \)

• Selecting \( \Delta \) to minimize the bound gives \( \Delta^2 = \frac{\rho \|L_\rho(u)\|_2 r}{\sqrt{M_T}} \)

and leads to

\[ M_T^2 \leq \frac{r^2}{\rho^2} + 2 \sqrt{M_T \|L_\rho(u)\|_2} \frac{r}{\rho} + M_T \|L_\rho(u)\|^2 = \left( \frac{r}{\rho} + \sqrt{M_T \|L_\rho(u)\|_2} \right)^2. \]

• Solving the second-degree inequality

\[ M_T - \sqrt{M_T \|L_\rho(u)\|_2} - \frac{r}{\rho} \leq 0 \]

yields directly the first statement. The second one results from replacing \( r \) with \( \sqrt{M_T R} \).
Dual Perceptron Algorithm

\[ \text{Dual-Perceptron}(\alpha^0) \]

1. \( \alpha \leftarrow \alpha^0 \quad \triangleright \text{typically } \alpha^0 = 0 \)
2. \textbf{for } t \leftarrow 1 \textbf{ to } T \textbf{ do} \\
3. \quad \text{Receive}(x_t) \\
4. \quad \hat{y}_t \leftarrow \text{sgn}\left(\sum_{s=1}^{T} \alpha_s y_s (x_s \cdot x_t)\right) \\
5. \quad \text{Receive}(y_t) \\
6. \quad \textbf{if } (\hat{y}_t \neq y_t) \textbf{ then} \\
7. \quad \quad \alpha_t \leftarrow \alpha_t + 1 \\
8. \quad \textbf{return } \alpha
Kernel Perceptron Algorithm

(Aizerman et al., 1964)

$K$ PDS kernel.

**Kernel-Perceptron($\alpha^0$)**

1. $\alpha \leftarrow \alpha^0$  \> typically $\alpha^0 = 0$
2. **for** $t \leftarrow 1$ **to** $T$ **do**
3. \hspace{1em} **Receive**($x_t$)
4. \hspace{1em} $\hat{y}_t \leftarrow \text{sgn}(\sum_{s=1}^{T} \alpha_s y_s K(x_s, x_t))$
5. \hspace{1em} **Receive**($y_t$)
6. \hspace{1em} **if** ($\hat{y}_t \neq y_t$) **then**
7. \hspace{2em} $\alpha_t \leftarrow \alpha_t + 1$
8. **return** $\alpha$
Winnow Algorithm

(Winnow, 1988)

Winnow(η)

1. \( w_1 \leftarrow 1/N \)
2. \textbf{for} \( t \leftarrow 1 \text{ to } T \) \textbf{do}
3. \hspace{0.5cm} \textbf{Receive}(x_t)
4. \hspace{0.5cm} \hat{y}_t \leftarrow \text{sgn}(w_t \cdot x_t) \quad \triangleright \quad y_t \in \{-1, +1\}
5. \hspace{0.5cm} \textbf{Receive}(y_t)
6. \hspace{0.5cm} \textbf{if} \ (\hat{y}_t \neq y_t) \ \textbf{then}
7. \hspace{1.5cm} Z_t \leftarrow \sum_{i=1}^{N} w_{t,i} \exp(\eta y_t x_{t,i})
8. \hspace{1.5cm} \textbf{for} \ i \leftarrow 1 \text{ to } N \ \textbf{do}
9. \hspace{2.5cm} w_{t+1,i} \leftarrow \frac{w_{t,i} \exp(\eta y_t x_{t,i})}{Z_t}
10. \hspace{1.5cm} \textbf{else} \ w_{t+1} \leftarrow w_t
11. \ \textbf{return} \ w_{T+1}
Winnow = weighted majority:

- for $y_{t,i} = x_{t,i}$ $\in \{-1, +1\}$, $\text{sgn}(w_t \cdot x_t)$ coincides with the majority vote.

- multiplying by $e^{\eta}$ 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 $v \in \mathbb{R}^N$, $v \geq 0$ for all $t \in [1, T]$, 

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

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

**Proof:** Let $I$ be the set of $t$s at which there is an update and let $M$ be the total number of updates.
Comparison with perceptron bound:

- dual norms: norms for $x_t$ and $v$.
- 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 $v = e_1$ and $x_t \in \{\pm 1\}^N$, $\log N$ vs $N$.
  - Perceptron favorable in opposite situation.
Winnow Algorithm - Bound

**Potential:** \( \Phi_t = \sum_{i=1}^{N} \frac{v_i}{\|v\|} \log \frac{v_i/\|v\|}{w_{t,i}} \). (relative entropy)

**Upper bound:** for each \( t \) in \( I \),

\[
\Phi_{t+1} - \Phi_t = \sum_{i=1}^{N} \frac{v_i}{\|v\|} \log \frac{w_{t,i}}{w_{t+1,i}} \\
= \sum_{i=1}^{N} \frac{v_i}{\|v\|} \log \frac{Z_t^i}{\exp(\eta y_t x_{t,i})} \\
= \log Z_t - \eta \sum_{i=1}^{N} \frac{v_i}{\|v\|} y_t x_{t,i} \\
\leq \log \left[ \sum_{i=1}^{N} w_{t,i} \exp(\eta y_t x_{t,i}) \right] - \eta \rho_\infty \\
= \log \mathbb{E}_{w_t} \left[ \exp(\eta y_t x_t) \right] - \eta \rho_\infty \\
(Hoeffding) \leq \log \left[ \exp(\eta^2 \left(2R_\infty \right)^2 / 8) \right] + \eta y_t w_t \cdot x_t - \eta \rho_\infty \\
\leq \eta^2 R_\infty^2 / 2 - \eta \rho_\infty. 
\]
Winnow Algorithm - Bound

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

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

**Lower bound:** note that

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

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:** $-\log N \leq M\left(\eta^2 R_\infty^2 / 2 - \eta \rho_\infty\right)$. 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


References


Appendix
SVMs - Leave-One-Out-Analysis

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

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

**Proof:** one part proven in lecture 4. The other part due to $\alpha_i \geq 1/R_{m+1}^2$ for $x_i$ 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 $\rho_{m+1}$ can be used in both.
  - but different radius $R_{m+1}$ of support vectors.

- Difference: margin distribution.