# Introduction to Machine Learning
# Lecture 6

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

# Perceptron and Winnow

# This Lecture

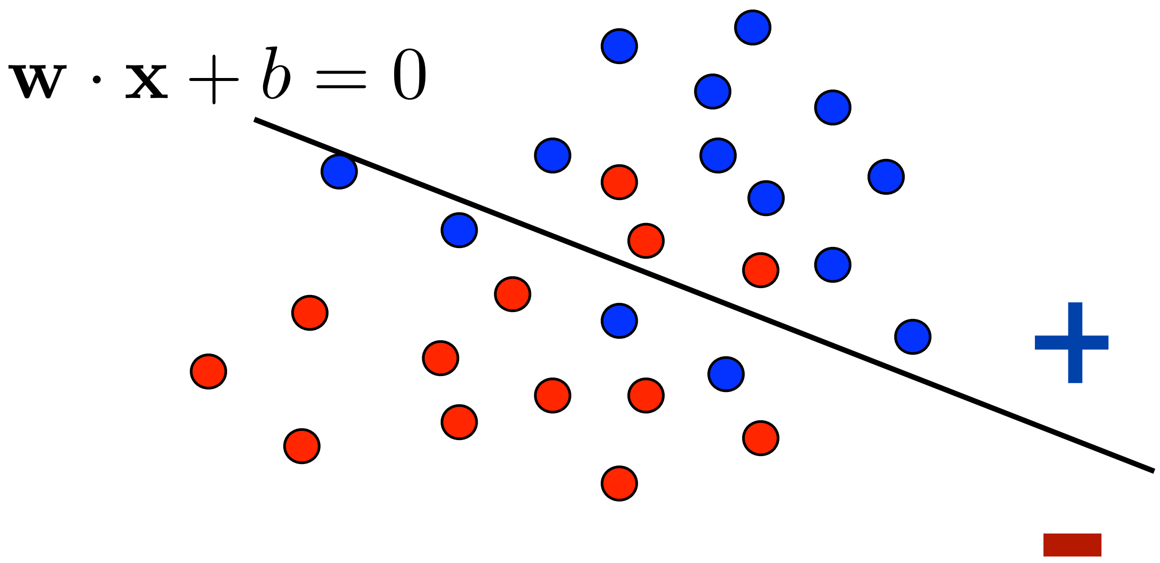
- On-Line linear classification: two algorithms.
  - Perceptron algorithm.
  - Winnow algorithm.

# Linear Classification

- Definition: a linear classifier is an algorithm that returns a hypothesis of the form

$$x \mapsto \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x} + b),$$

with $\mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}$.



# Margin Definitions

- Definition: the (geometric) margin of a point $\mathbf{x}$ with label y for a linear classifier $h\colon \mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} + b$ is its algebraic distance to the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$,

$$\rho(x) = \frac{y(\mathbf{w} \cdot \mathbf{x} + b)}{\|\mathbf{w}\|}.$$

- Definition: the margin of a linear classifier $h$ for a sample $S = (x_1, \ldots, x_m)$ is the minimum margin of the points in that sample:

$$\rho = \min_{1 \leq i \leq m} \frac{y_i(\mathbf{w} \cdot \mathbf{x}_i + b)}{\|\mathbf{w}\|}.$$

# 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**

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

$\quad \widehat{y}_t \leftarrow \operatorname{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$

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

$\quad$ **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** $\mathbf{w}_{T+1}$

# Perceptron - Notes

- Update: if $y_t(\mathbf{w}_t \cdot \mathbf{x}_t) < 0$, then

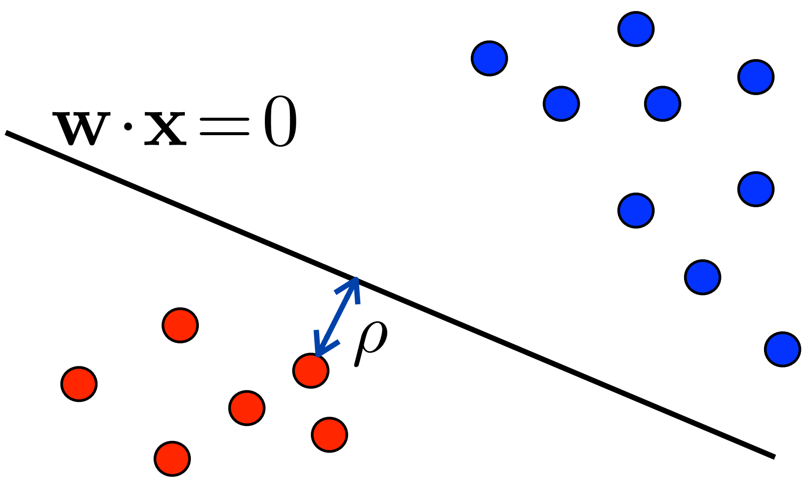
$$y_t(\mathbf{w}_{t+1} \cdot \mathbf{x}_t) = y_t(\mathbf{w}_t \cdot \mathbf{x}_t) + \underbrace{\eta \|\mathbf{x}_t\|^2}_{\geq 0}.$$

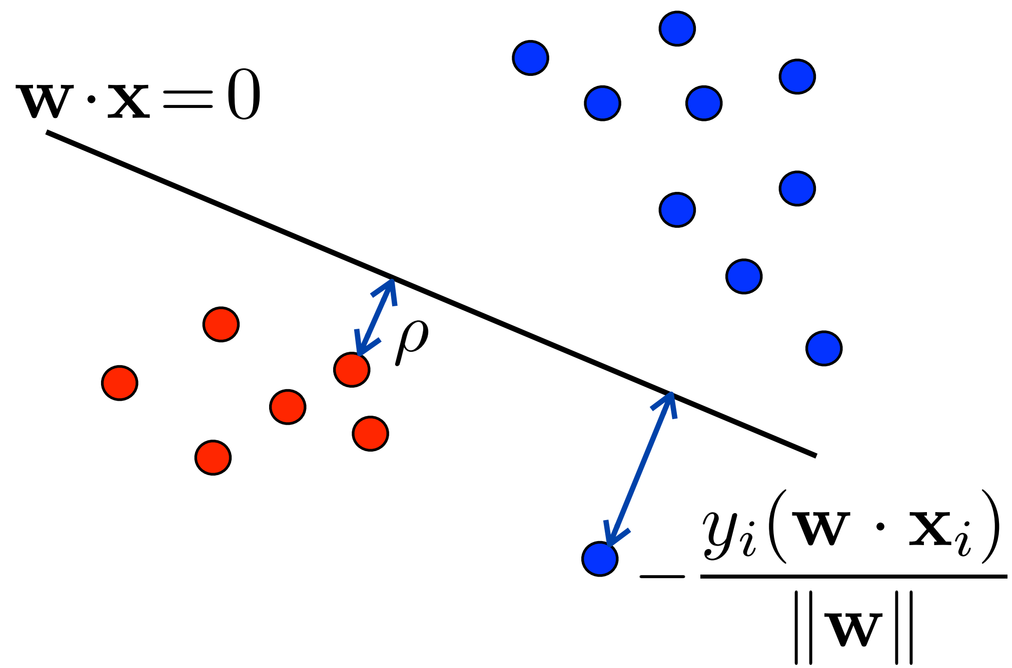
→ change in the desired direction.

- Different modes of applications:
  - repeated passes over sample of size $m$ drawn according to some distribution $D$.
  - infinite sample drawn according to $D$.
  - no distributional assumption.

# 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 $t$s at which there is an update and let $M$ be the total number of updates.

- Summing up the assumption inequalities gives:

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

- 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

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

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

# Leave-One-Out Error

- **Definition**: let $h_S$ be the hypothesis output by learning algorithm $L$ after receiving sample $S$ of size $m$. Then, the leave-one-out error of $L$ over $S$ is:

$$\widehat{R}_{\text{loo}}(L) = \frac{1}{m} \sum_{i=1}^{m} 1_{h_{S-\{x_i\}}(x_i) \neq f(x_i)}.$$

- **Property**: unbiased estimate of expected error of hypothesis trained on sample of size $m-1$,

$$\boxed{\mathop{\mathrm{E}}_{S \sim D^m}[\widehat{R}_{\text{loo}}(L)]} = \frac{1}{m} \sum_{i=1}^{m} \mathop{\mathrm{E}}_{S}[1_{h_{S-\{x_i\}}(x_i) \neq f(x_i)}] = \mathop{\mathrm{E}}_{S}[1_{h_{S-\{x\}}(x) \neq f(x)}]$$

$$= \mathop{\mathrm{E}}_{S' \sim D^{m-1}}[\mathop{\mathrm{E}}_{x \sim D}[1_{h_{S'}(x) \neq f(x)}]] \boxed{= \mathop{\mathrm{E}}_{S' \sim D^{m-1}}[R(h_{S'})].}$$

# Perceptron - Leave-One-Out Analysis

- Theorem: Assume that the data is separable. Let $h_S$ be the hypothesis returned by the Perceptron algorithm after training on sample $S \sim D^{m+1}$ (repeated passes) and let $M(S)$ be the number of updates made and let $R(h_S)$ be the error of $h_S$. 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 $\mathbf{x}$ be a point in sample $S$. Then, If $h_{S-\{\mathbf{x}\}}$ misclassifies $\mathbf{x}$, there must have been an update at $\mathbf{x}$ during training to obtain $h_S$. Thus,

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

# Dual Perceptron Algorithm

DUAL-PERCEPTRON($\boldsymbol{\alpha}_0$)

$\boldsymbol{\alpha}_1 \leftarrow \boldsymbol{\alpha}_0$ $\qquad \triangleright$ typically $\boldsymbol{\alpha}_0 = \mathbf{0}$
**for** $t \leftarrow 1$ **to** $T$ **do**
    RECEIVE($\mathbf{x}_t$)
    $\widehat{y}_t \leftarrow \operatorname{sgn}\left(\sum_{s=1}^{T} \alpha_s y_s (\mathbf{x}_s \cdot \mathbf{x}_t)\right)$
    RECEIVE($y_t$)
    **if** $(\widehat{y}_t \neq y_t)$ **then**
        $\alpha_{t+1} \leftarrow \alpha_t + 1$
    **else** $\alpha_{t+1} \leftarrow \alpha_t$
**return** $\boldsymbol{\alpha}$

# Kernel Perceptron Algorithm

(Aizerman et al., 1964)

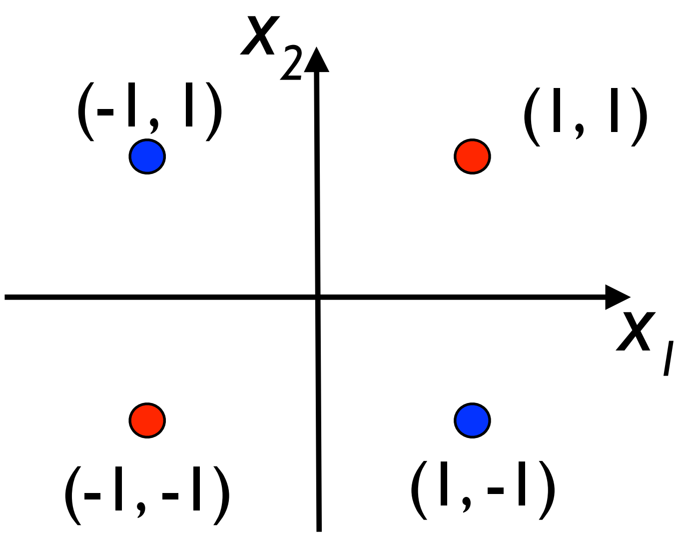
$K$ PDS kernel.

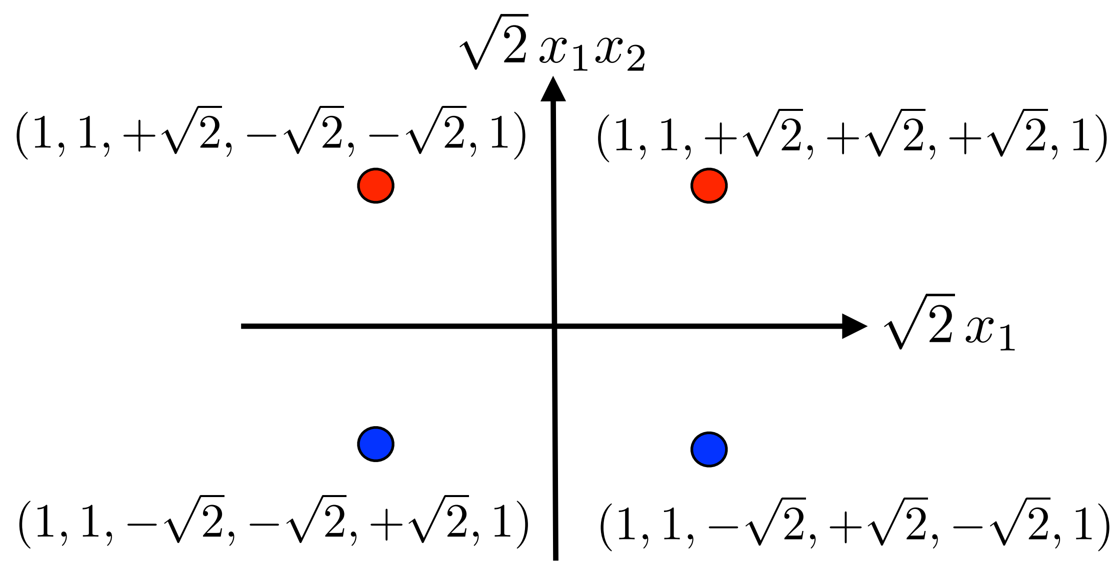
$\textsc{Kernel-Perceptron}(\boldsymbol{\alpha}_0)$

$\boldsymbol{\alpha}_1 \leftarrow \boldsymbol{\alpha}_0 \qquad \triangleright$ typically $\boldsymbol{\alpha}_0 = \mathbf{0}$
**for** $t \leftarrow 1$ **to** $T$ **do**
  $\textsc{Receive}(x_t)$
  $\widehat{y}_t \leftarrow \text{sgn}(\sum_{s=1}^{T} \alpha_s y_s K(x_s, x_t))$
  $\textsc{Receive}(y_t)$
  **if** $(\widehat{y}_t \neq y_t)$ **then**
    $\alpha_{t+1} \leftarrow \alpha_t + 1$
  **else** $\alpha_{t+1} \leftarrow \alpha_t$
**return** $\boldsymbol{\alpha}$

# XOR Problem

- Use second-degree polynomial kernel with $c = 1$:




Linearly non-separable

Linearly separable by $x_1 x_2 = 0.$

# Non-Separable Case

(Freund and Schapire, 1998)

- Theorem: Let $\mathbf{v}$ be any vector with $\|\mathbf{v}\| = 1$ and let $\rho > 0$. Define the deviation of $\mathbf{x}_t$ by:

$$d_t = \max\{0, \rho - y_t(\mathbf{v} \cdot \mathbf{x}_t)\},$$

and let $D = \sqrt{\sum_{t=1}^{T} d_t^2}$. Then, the number of perceptron updates after processing $\mathbf{x}_1, \ldots, \mathbf{x}_T$ is bounded by

$$\left[\frac{R + D}{\rho}\right]^2.$$

- Proof: Reduce problem to separable case in higher dimension.
  - Mapping (similar to trivial mapping):

$(N+t)$th component

$$\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \end{bmatrix} \rightarrow \mathbf{x}'_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \\ 0 \\ \vdots \\ 0 \\ \Delta \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \mathbf{v} \rightarrow \mathbf{v}' = \begin{bmatrix} v_1/Z \\ \vdots \\ v_N/Z \\ y_1 d_1/(\Delta Z) \\ \vdots \\ y_T d_T/(\Delta Z) \end{bmatrix}$$

$$\|\mathbf{v}'\| = 1 \Rightarrow Z = \sqrt{1 + \frac{D^2}{\Delta^2}}.$$

- Now, $y_t(\mathbf{v}' \cdot \mathbf{x}'_t) = y_t\Big(\frac{\mathbf{v} \cdot \mathbf{x}_t}{Z} + \Delta \frac{y_t d_t}{Z\Delta}\Big)$

$$= \frac{y_t \mathbf{v} \cdot \mathbf{x}_t}{Z} + \frac{d_t}{Z}$$

$$\geq \frac{y_t \mathbf{v} \cdot \mathbf{x}_t}{Z} + \frac{\rho - y_t(\mathbf{v} \cdot \mathbf{x}_t)}{Z} = \frac{\rho}{Z}.$$

- Since $||\mathbf{x}'_t||^2 \leq R^2 + \Delta^2$, the bound of the separable case applies: $\frac{(R^2+\Delta^2)(1+D^2/\Delta^2)}{\rho^2}$.
- With $\Delta = \sqrt{RD}$, this bound is minimized and equal to: $\frac{(R+D)^2}{\rho^2}$.
- Predictions made by the perceptron in the higher-dimension coincide with those of the perceptron in the original space.

# Winnow Algorithm

(Littlestone, 1988)

Winnow($\eta$)

$w_1 \leftarrow \mathbf{1}/N$
**for** $t \leftarrow 1$ **to** $T$ **do**
  Receive($\mathbf{x}_t$)
  $\widehat{y}_t \leftarrow \operatorname{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$  $\triangleright \quad y_t \in \{-1, +1\}$
  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** $\mathbf{w}_{T+1}$

# Winnow - 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 $e^{\eta}$ 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).
- Motivation: large number of irrelevant features.

# 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 $\boxed{2\left(R_\infty^2/\rho_\infty^2\right)\log N}$.

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

# 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 $I$,

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

# 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 or 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

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

# 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.