

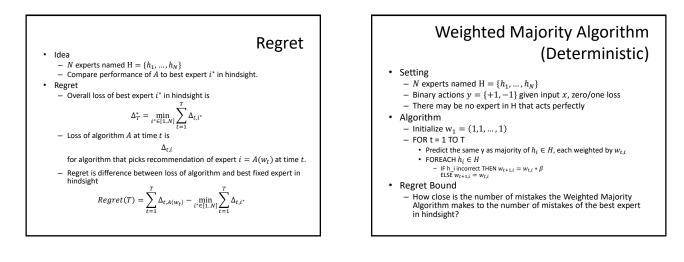
### (Online) Perceptron Algorithm • Input: $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n)), \vec{x}_i \in \Re^N, y_i \in \{-1, 1\}$ • Algorithm: $-\vec{w}_0 = \vec{0}, k = 0$ - FOR i=1 TO n $* \text{ IF } y_i(\vec{w}_k \cdot \vec{x}_i) \le 0 \#\#\# \text{ makes mistake}$ $\cdot \vec{w}_{k+1} = \vec{w}_k + y_i \vec{x}_i$ $\cdot k = k + 1$ \* ENDIF - ENDFOR• Output: $\vec{w}_k$ .

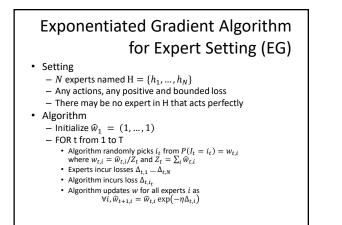
## Perceptron Mistake Bound

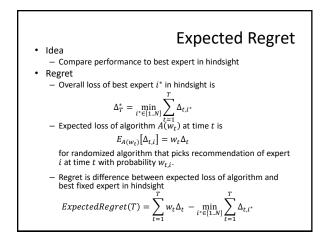
Theorem: For any sequence of training examples  $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n) \text{ with } R = \max \|\vec{x}_i\|$ , if there exists a weight vector  $\vec{w}_{opt}$  with  $\|\vec{w}_{opt}\| = 1$ and  $y_i (\vec{w}_{opt} \cdot \vec{x}_i) \ge \delta$ for all  $1 \le i \le n$ , then the Perceptron makes at most  $\frac{R^2}{\delta^2}$ errors.

# $\begin{array}{l} \textbf{Expert Learning Model} \\ \textbf{e. Setting} \\ & - \textit{N} \text{ experts named } H = \{h_1, ..., h_N\} \\ & - \text{ Each expert } h_i \text{ takes an action } y = h_i(x_t) \text{ in each round t} \\ & \text{ and incurs loss } \Delta_{t,i} \\ & - \text{ Algorithm can select which expert's action to follow in} \\ & \text{ each round} \\ \textbf{e. Interaction Model} \\ & - \text{ FOR t from 1 to T} \\ & \text{ Algorithm selects expert } h_{i_t} \text{ according to strategy } A_{w_t} \text{ and follows} \\ & \text{ its action } y \\ & \text{ experts incur losses } \Delta_{t,i} \\ & \text{ Algorithm incurs loss } \Delta_{t,i_t} \\ & \text{ Algorithm incurs loss } \Delta_{t,i_t} \\ & \text{ Algorithm updates } w_t \text{ to } w_{t+1} \text{ based on } \Delta_{t,1} \dots \Delta_{t,N} \end{array}$

## $\begin{array}{l} \textbf{Halving Algorithm} \\ \bullet & \text{Setting} \\ \bullet & N \text{ experts named } H = \{h_1, \dots, h_N\} \\ \bullet & \text{Binary actions } y = \{\pm 1, -1\} \text{ given input } x, \text{ zero/one loss} \\ \bullet & \text{Perfect expert exists in } H \\ \bullet & \text{Algorithm} \\ \bullet & VS_1 = H \\ \bullet & \text{FOR t = 1 TO T} \\ \bullet & \text{Predict the same y as majority of } h_i \in VS_t \\ \bullet & VS_{t+1} = VS_t \text{ minus those } h_i \in VS_t \text{ that were wrong} \\ \bullet & \text{Mistake Bound} \\ \bullet & \text{How many mistakes can the Halving algorithm make before predicting perfectly?} \end{array}$







## Regret Bound for Exponentiated Gradient Algorithm

Theorem

The expected regret of the exponentiated gradient algorithm in the expert setting is bounded by

 $\operatorname{Expected} Regret(T) \leq \sqrt{2 T \log(|H|)}$ 

where  $\Delta \in [0,1]$  and  $\eta = \sqrt{2 \log(|H|) / T}$  and  $T > 2 \log(|H|)$ .