A more general setting

Example	Prediction of alg A	Label	Loss of alg A
$\boldsymbol{x}_1$	$\hat{y}_1$	y_1	$L(y_1, \hat{y}_1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{x}_t$	$\hat{y}_t$	y_t	$L(y_t, \hat{y}_t)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{x}_T$	$\hat{y}_T$	y_T	$L(y_T, \hat{y}_T)$
		Total Loss	$L_A(S)$

Sequence of examples $S = (\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)$

Comparison class $\{\boldsymbol{u}\}$

Relative loss $\quad L_A(S) - \inf_{\{\boldsymbol{u}\}} L_{\boldsymbol{u}}(S)$

Goal: Bound relative loss for arbitrary sequence of examples

Example: Learning Disjunctions of Experts

variables/experts

E_1	E_2	E_3	E_4	*true label*	$E_1 \vee E_3$	$E_3 \vee E_4$
1	1	0	0	0	1	0
1	0	1	0	1	1	1
0	1	1	1	0	1	1
0	1	0	0	1	0	0
$x_{t,1}$	$x_{t,2}$	$x_{t,3}$	$x_{t,4}$		$\uparrow$	$\uparrow$
					3	2

mistakes

$E_1 \vee E_3$ becomes $\boldsymbol{u} = (1, 0, 1, 0)$

$E_1 \vee E_3$ is one on $\boldsymbol{x}_t \in \{0,1\}^n$ iff $\boldsymbol{u} \cdot \boldsymbol{x}_t \geq 1$

Weighted Majority on k-literal Disjunctions

Do as well as best k out of n literal (monotone) disjunction

Each disjunction is an expert

Keep one weight per disjunction: $\binom{n}{k}$ weights

$$\begin{array}{c}\#\text{ of mistakes}\\ \text{of WM}\end{array} \leq 2.63\, M + 2.63\, k \ln \frac{n}{k}$$

M is # of mistakes of best

Time (and space) **exponential** in k

Efficient algorithm have only one weight per literal

The Perceptron Algorithm

In trial t: Get instance $\boldsymbol{x}_t \in \{0,1\}^n$

If $\boldsymbol{w}_t \cdot \boldsymbol{x}_t \geq 1/2$ then $\hat{y}_t = 1$

else $\hat{y}_t = 0$

Get label $y_t \in \{0,1\}$

If mistake then

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\,(\hat{y}_t - y_t)\,\boldsymbol{x}_t$$

k-literal Disjunctions with Perceptron

Perceptron Convergence Theorem ($\eta = \frac{1}{2n}$)

$$\text{\# of mistakes} \leq 4\ A\ +\ 4\ k\ n$$

where A is # of attribute errors of best disjunction of size k, i.e., the minimum # of attributes that need to be flipped to make the disjunction consistent

$$A \leq kM$$

Lower bound for rotation invariant algorithms: [KWA]

$$\text{\#mistakes} = \Omega(n)$$

The Winnow Algorithm [L]

In trial t: Get instance $\boldsymbol{x}_t \in \{0,1\}^n$

If $\boldsymbol{w}_t \cdot \boldsymbol{x}_t \geq \theta$ then $\hat{y}_t = 1$

else $\hat{y}_t = 0$

Get label $y_t \in \{0,1\}$

If mistake then

$$w_{t+1,i} = w_{t,i}\, e^{-\eta\,(\hat{y}_t - y_t)\, x_{t,i}}$$

Mistake bound ($e^{-\eta} = 1/3$, $\theta = \frac{3\ln 3}{8}$) [AW]

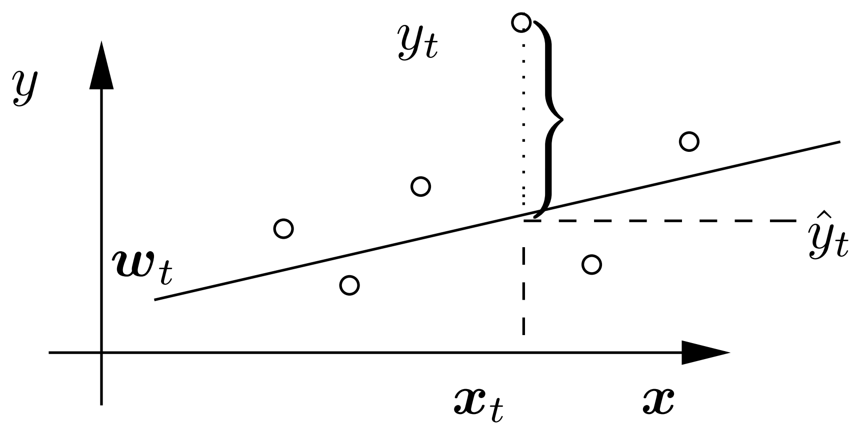
$$\#\text{ of mistakes} \leq 4\, A \;+\; 3.6\, k \ln \frac{n}{k}$$

Not rotation invariant!

On-line Linear Regression

For $t = 1, \ldots, T$ do

Get instance	$\boldsymbol{x}_t \in \mathbf{R}^n$
Predict	$\hat{y}_t = \boldsymbol{w}_t \cdot \boldsymbol{x}_t$
Get label	$y_t \in \mathbf{R}$
Incur loss	$L_t(\boldsymbol{w}_t) = (y_t - \hat{y}_t)^2$
Update	$\boldsymbol{w}_t$ to $\boldsymbol{w}_{t+1}$



Assume comparison class $\{\boldsymbol{u}\}$ is a set of linear predictors

$$\boldsymbol{u} \;:\; \boldsymbol{x} \to \boldsymbol{u} \cdot \boldsymbol{x}$$

Examples of Updates

Gradient descent

($\boldsymbol{w} \in \mathbf{R}^n$)

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_t(\boldsymbol{w}_t)$$

$$= \boldsymbol{w}_t - \eta(\boldsymbol{w}_t \cdot \boldsymbol{x}_t - y_t)\,\boldsymbol{x}_t \qquad \text{[WH]}$$

Exponentiated Gradient Algorithm [KW]

($\boldsymbol{w}$ is probability vector)

$$\boldsymbol{w}_{t+1,i} = w_{t,i}\ \exp\left[-\eta \frac{\partial L_t(\boldsymbol{w}_t)}{\partial w_{t,i}}\right] / \text{ normalization}$$

Motivation of Updates [KW]

Gradient descent

$$\begin{aligned}\boldsymbol{w}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{w}} \left(||\boldsymbol{w} - \boldsymbol{w}_t||_2^2/2 + \eta(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2/2\right) \\ &= \boldsymbol{w}_t - \eta(\underbrace{\boldsymbol{w}_{t+1} \cdot \boldsymbol{x}_t}_{\approx \boldsymbol{w}_t \cdot \boldsymbol{x}_t} - y_t)\, \boldsymbol{x}_t\end{aligned}$$

Exponentiated Gradient Algorithm

$$\begin{aligned}\boldsymbol{w}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{w}} \left(\sum_{i=1}^{n} w_i \ln \frac{w_i}{w_{t,i}} + \eta(y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2/2\right) \\ &= w_{t,i}\ \exp\left[-\eta\,(\underbrace{\boldsymbol{w}_{t+1} \cdot \boldsymbol{x}_t}_{\approx \boldsymbol{w}_t \cdot \boldsymbol{x}_t} - y_t)\, x_{t,i}\right] / \text{ normalization}\end{aligned}$$

Families of update algorithms

parameter "divergence"	name of family	update algorithms
$\|\|\boldsymbol{w} - \boldsymbol{w}_t\|\|_2^2$	Gradient Descent	Widrow Hoff (LMS) Linear Least Squares. Backpropagation Perceptron Algorithms kernel based algorithms,...
$\sum_{i=1}^{n} w_i \ln \frac{w_i}{w_{t,i}}$	Exponentiated Gradient Algorithm	expert algs Normalized Winnow "AdaBoost"

Families of update algorithms (cont)

parameter "divergence"	name of family	update algorithms
$\sum_{i=1}^{n} w_i \ln \frac{w_i}{w_{t,i}}$ $+ w_{t,i} - w_i$	Unnormalized Exp. Grad. Alg.	Winnow
$\sum_{i=1}^{n} w_i \ln \frac{w_i}{w_{t,i}}$ $+(1 - w_i) \ln \frac{1-w_i}{1-w_{t,i}}$	Binary Exp. Grad. Alg.	
any "Bregman divergence"	-	-

Members of different families exhibit different behavior

Loss bounds

Assume Example Sequence is

$$(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_t, y_t), \ldots \text{ where } y_t = \mathbf{u} \cdot \boldsymbol{x}_t$$

(the zero-error case)

Gradient Descent:

$$L_{GD}(S) \leq \left(\|\mathbf{u}\|_2 \max_t \|\boldsymbol{x}_t\|_2 \right)^2$$

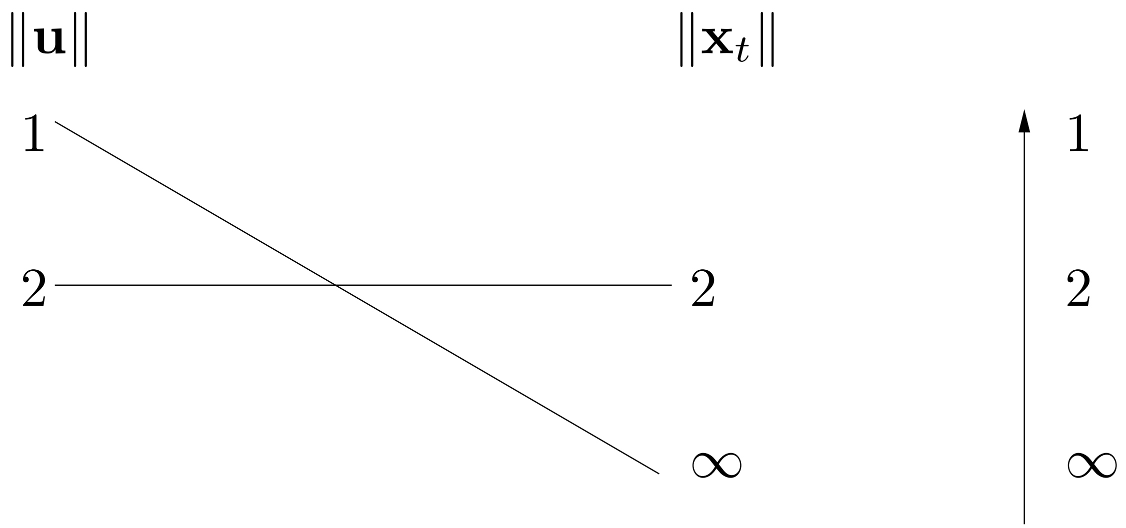
Exponentiated Gradients:

$$L_{EG\pm}(S) \leq \left(\|\mathbf{u}\|_1 \max_t \|\boldsymbol{x}_t\|_\infty \right)^2 \log(2n)$$

Incomparable Loss bounds

$$
\begin{aligned}
L_{GD}(S) &\leq \left(\|\mathbf{u}\|_2 \max_t \|\boldsymbol{x}_t\|_2 \right)^2 \\
L_{EG\pm}(S) &\leq \left(\|\mathbf{u}\|_1 \max_t \|\boldsymbol{x}_t\|_\infty \right)^2 \log(2n)
\end{aligned}
$$

Products of two norms:



Summary of Comparison

- EG better when:

- Instances $\boldsymbol{x}_t$ are dense ($\|\boldsymbol{x}_t\|_\infty \ll \|\boldsymbol{x}_t\|_2$)

- best weight vector is sparse ($\|\boldsymbol{u}_t\|_1 \approx \|\boldsymbol{u}_t\|_2$)

- GD better when:

- instances are sparse ($\|\boldsymbol{x}_t\|_\infty \approx \|\boldsymbol{x}_t\|_2$)

- best weight vector is dense ($\|\boldsymbol{u}_t\|_2 \ll \|\boldsymbol{u}_t\|_2$)

GD can be exponentially worse than EG