Proving Relative Loss Bounds for On-Line Learning Algorithms Using Bregman Divergences

Claudio Gentile
Universita' di Milano, Italy

Manfred K. Warmuth
UC Santa Cruz, USA

July 13, 2000

Early contributors:

Nick Littlestone
Volodya Vovk
Tom Cover

On-Line Learning

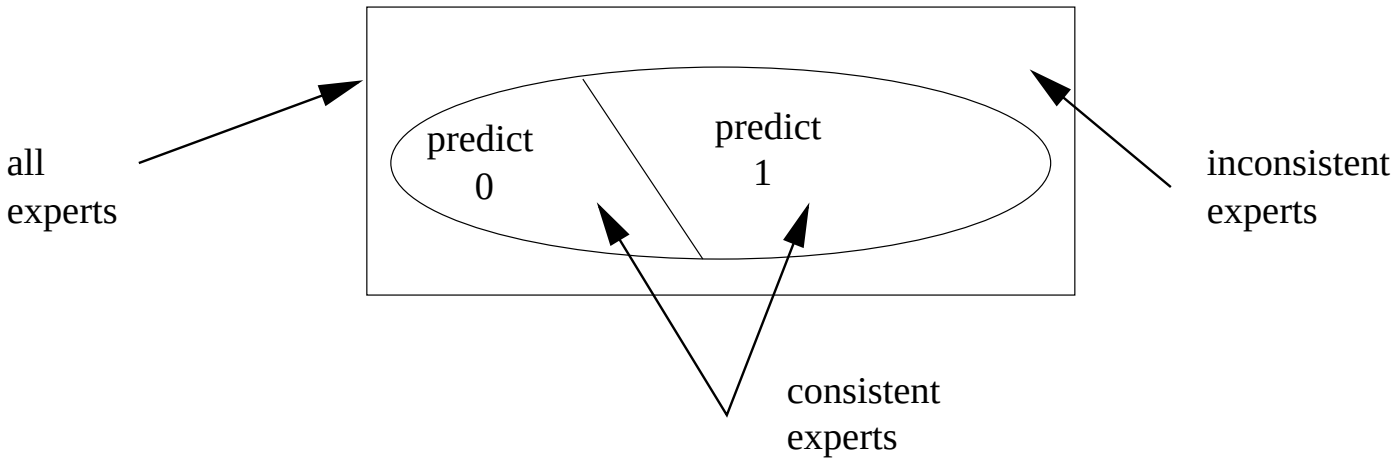
	experts						
	E_1	E_2	E_3	E_n	predic tion	*true label*	loss
day 1	1	1	0	0	0	1	1
day 2	1	0	1	0	1	0	1
day 3	0	1	1	1	1	1	0
day t	$x_{t,1}$	$x_{t,2}$	$x_{t,3}$	$x_{t,n}$	$\widehat{y}_t$	y_t	$\lvert y_t - \widehat{y}_t \rvert$

Protocol of the Master Algorithm

For $t = 1$ To T Do

Get instance	$\boldsymbol{x}_t \in \{0,1\}^n$
Predict	$\widehat{y}_t \in \{0,1\}$
Get label	$y_t \in \{0,1\}$
Incur loss	$\lvert y_t - \widehat{y}_t \rvert$

Halving Algorithm [BF]



- Predicts with majority

- If mistake is made then number of consistent experts is (at least) halved

A run of the Halving Algorithm

E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	$majority$	$true\ label$	loss
1	1	0	0	1	1	0	0	1	0	1
x	x	0	1	x	x	1	1	1	1	0
x	x	x	1	x	x	0	0	0	1	1
x	x	x	↑	x	x	x	x			
			consistent							

For any sequence with a consistent expert HA makes $\leq \log_2 n$ mistakes

What if no expert is consistent?

Sequence of examples $S = (x_1, y_1), \ldots, (x_T, y_T)$

- $L_A(S)$ be the total loss of alg. A
- $L_i(S)$ be the total loss of i-th expert E_i

Want bounds of the form:

$$\forall S: \quad L_A(S) \leq a \min_i L_i(S) + b \log(n)$$

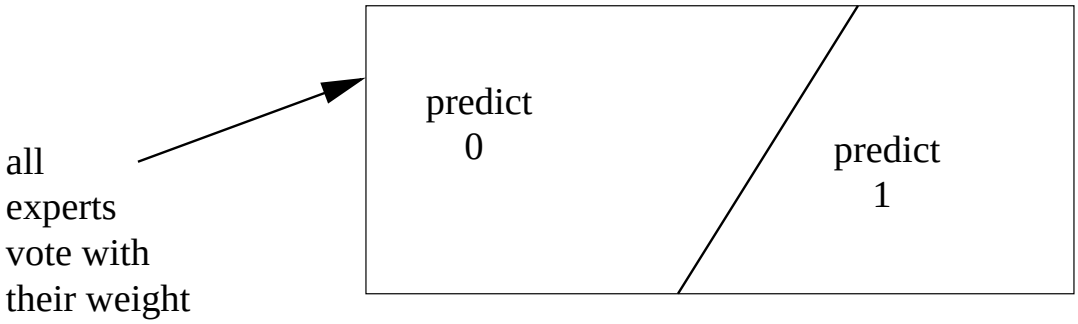
where a, b are constants

Bounds loss of algorithm relative to loss of best expert

Can't wipe out experts!

One weight per expert

Weighted Majority Algorithm [LW]



- Predicts with larger side

- Weights of wrong experts are multiplied by $\beta \in [0, 1)$

Number of mistakes of the WM algorithm

$M_{t,i}$ = # of mistakes of E_i before trial t

$w_{t,i}$ = $\beta^{M_{t,i}}$ weight of E_i at beginning of trial t

$W_t = \sum_{i=1}^{n} w_{t,i}$ total weight at trial t

Minority $\leq \frac{1}{2} W_t$

Majority $\geq \frac{1}{2} W_t$

If no mistake then

minority multiplied by β:

$W_{t+1} \leq 1\, W_t$

If mistake then

majority multiplied by β:

$$W_{t+1} \leq \underbrace{1 \quad \frac{1}{2} W_t}_{\text{minority}} + \underbrace{\beta \quad \frac{1}{2} W_t}_{\text{majority}}$$

$$= \frac{1+\beta}{2}\, W_t$$

Hence

$$\underset{\substack{total final\\ weight}}{W_{T+1}} \leq \left(\frac{1+\beta}{2}\right)^M W_1$$

$$W_{T+1} = \sum_{j=1}^{n} w_{T+1,j} = \sum_{j=1}^{n} \beta^{M_j} \geq \beta^{M_i}$$

We got:

$$\left(\frac{1+\beta}{2}\right)^M \underbrace{W_1}_{n} \geq \beta^{M_i}$$

Solving for M:

$$M \leq \frac{\ln \frac{1}{\beta}}{\ln \frac{2}{1+\beta}} M_i + \frac{1}{\ln \frac{2}{1+\beta}} \ln n$$

$$M \underset{\beta = 1/e}{\leq} \underbrace{2.63}_{a} \min_i M_i + \underbrace{2.63}_{b} \ln n$$

For all sequences, loss of master alg. is comparable to loss of best expert
Relative loss bounds [Fr]

Other Loss Functions

absolute loss $L(y, \widehat{y}) = |y - \widehat{y}|$

square loss $L(y, \widehat{y}) = (y - \widehat{y})^2$

entropic loss $L(y, \widehat{y}) = y \ln \frac{y}{\widehat{y}} + (1 - y) \ln \frac{1-y}{1-\widehat{y}}$

$y, \widehat{y} \in [0, 1]$

One weight per expert: [V]

$w_{t,i} = \beta^{\,L_{t,i}} = e^{-\eta\, L_{t,i}}$

where $L_{t,i}$ is total loss of E_i before trial t
and η is a positive learning rate

Master predicts with the weighted average [KW]

$v_{t,i} = \dfrac{w_{t,i}}{\sum_{i=1}^{n} w_{t,i}}$ normalized weights

$\widehat{y}_t = \sum_{i=1}^{n} v_{t,i}\, x_{t,i} = \boldsymbol{v}_t \cdot \boldsymbol{x}_t$

where $x_{t,i}$ is the prediction of E_i in trial t

$\forall$ sequences S of examples $\langle(\boldsymbol{x}_t, y_t)\rangle_{1\le t\le T}$ where $\boldsymbol{x}_t \in [0,1]^n$ and $y_t \in [0,1]$

$$L_{WA}(S) \le \min_i \underbrace{1}_{a} L_i(S) + \underbrace{1/\eta}_{b} \ln(n)$$

$1/\eta$	dot pred	fancy
entropic	1	1
square	2	1/2
hellinger	1	.71

- Slightly improved constants of $1/\eta$ when Master uses fancier prediction [V]

- For the discrete loss and the absolute loss $a > 1$

Potential: $\frac{1}{\eta} \ln W_t$

Key inequality: $L(y, \boldsymbol{v}_t \cdot \boldsymbol{x}_t) \leq \frac{1}{\eta} \ln W_t - \frac{1}{\eta} \ln W_{t+1}$
$= -\frac{1}{\eta} \ln \frac{W_{t+1}}{W_t}$

Telescoping:

$$\begin{aligned} L_{\mathsf{WA}}(S) &\leq -\frac{1}{\eta} \ln \frac{W_{T+1}}{W_1} \\ &= -\frac{1}{\eta} \ln \sum_{j=1}^{n} \frac{1}{n} e^{-\eta L_j(S)} \\ &\leq -\frac{1}{\eta} \ln \frac{1}{n} e^{-\eta L_i(S)} \\ &= -\frac{1}{\eta} \ln \frac{1}{n} e^{-\eta L_i(S)} \\ &= L_t(S) + \frac{1}{\eta} \ln n \end{aligned}$$

Usefulness:

- Easy to combine many pretty good experts (algorithms) so that Master is guaranteed to be almost as good as the best

- Bounds logarithmic
  in number of experts
  (multiplicative updates)

Questions:

- How to obtain algs. that do well compared to best linear combination or best thresholded linear combination of experts?
- How to motivate the updates?
- What are good measures of progress?
- What are good loss functions?
- Methods for proving relative loss bounds?

A more general setting

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

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

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

Relative loss

$$L_A(S) - \inf_{\{\boldsymbol{u}\}} Loss_{\boldsymbol{u}}(S)$$

Goal: Bound relative loss
for arbitrary sequence S

Learning Disjunctions of Experts

variables/experts

E_1	E_2	E_3	E_4	$\overset{true}{label}$	$E_1 \vee E_3$	$E_3 \vee E_4$
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
$x_{t,1}$	$x_{t,2}$	$x_{t,3}$	$x_{t,4}$		↑	↑
					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 $\boldsymbol{u} \cdot \boldsymbol{x}_t \geq 1$

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

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

M is # of mistakes of best

Time (and space) exponential in k

Efficient algs: 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

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

Perc. Conv. Th. ($\eta = \frac{1}{2n}$)

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

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 rot. inv. algs: [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!

k-term DNF via Feature Expansion [KW]

$$\boldsymbol{x} = \begin{matrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{matrix} \quad \Rightarrow \quad \Phi(\boldsymbol{x}) = \begin{matrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \\ x_1\, x_2 \\ \vdots \\ \vdots \\ x_1\, x_2 \ldots x_n \end{matrix}$$

n inputs $\qquad\qquad$ 2^n features

k-term DNF in input space
is k-literal disjunction in feature space

$$\underbrace{\Phi(\boldsymbol{x})}_{2^n} \cdot \underbrace{\Phi(\boldsymbol{y})}_{2^n} = \underbrace{\prod_{i=1}^{n} (1 + x_i\, y_i)}_{O(n) \text{ time}} = K(\underbrace{\boldsymbol{x}}_{n}, \underbrace{\boldsymbol{y}}_{n})$$

(Simple ANOVA kernel)

Perceptron: $\boldsymbol{w}_t = \sum_{q \text{ mistake}} \alpha_q \Phi(\boldsymbol{x}_i)$

Prediction:

$$\boldsymbol{w}_t \cdot \Phi(\boldsymbol{x}) = \left(\sum_{q \text{ mistake}} \alpha_q \Phi(\boldsymbol{x}_q) \right) \cdot \Phi(\boldsymbol{x})$$

$$= \sum_{q \text{ mistake}} \alpha_q \Phi(\boldsymbol{x}_q) \cdot \Phi(\boldsymbol{x})$$

$$= \underbrace{\sum_{q \text{ mistake}} \alpha_q K(\boldsymbol{x}_q, \boldsymbol{x})}_{\text{time: } O(n \cdot \#\text{ mistakes})}$$

Mistake bound: $O(k\, 2^n)$

Winnow: $w_{t,i} = \exp\left(-\eta \sum_{q \text{ mistake}} \alpha_q \, \Phi(\boldsymbol{x}_q)_i\right)$

log of weights is
linear comb of past examples
Mistake bound: $O(k \ln 2^n) = O(k\, n)$
prediction time: $\Omega(2^n \,\#$ mistakes$)$

No kernel trick with purely mult. updates!

So far

- Learning relative to best expert and best disjunction
- Various loss functions
- Perceptron versus Winnow and expansion into feature space

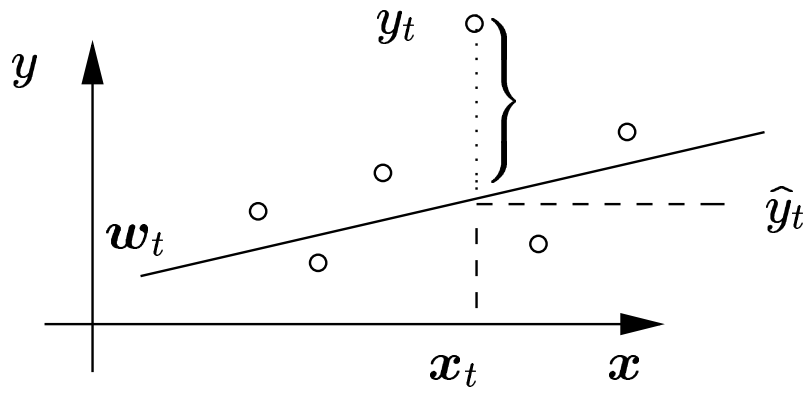
Rest of tutorial

- Motivation of updates
  with Bregman divergences
- Bregman divergences as loss functions
- Pythogorean Theorem
- Proving relative loss bounds
- Conversions to batch model
- Bregman divergences and
  the exponential family
- Comparator shifts with time

On-line Linear Regression

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

Get instance	$\boldsymbol{x}_t \in \mathbf{R}^n$
Predict	$\widehat{y}_t = \boldsymbol{w}_t \cdot \boldsymbol{x}_t$
Get label	$y_t \in \mathbf{R}$
Incur loss	$L_t(\boldsymbol{w}_t) = (y_t - \widehat{y}_t)^2$
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)

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

More examples of Updates

Unnormalized Exponentiated Gradient Alg.[KW]
$(\boldsymbol{w} \geq \mathbf{0})$

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

Binary Exponentiated Gradient Algorithm [By]
$(\boldsymbol{w} \in [0,1]^n)$

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

p-norm Algorithms [GLS,GL]

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

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

where

$$f(\boldsymbol{w}) = \nabla \tfrac{1}{2}||\boldsymbol{w}||_q^2 = \nabla \tfrac{1}{2} \left(\sum_i |w_i|^q\right)^{2/q}$$

and q dual to p (i.e., $\frac{1}{p} + \frac{1}{q} = 1$)

- $p = 2$ gets gradient descent
- $p = O(\log n)$ gets EG-like algs
- $2 < p < O(\log n)$ interpolates between the two extremes

Motivation of Updates [KW]

Gradient descent

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

Exponentiated Gradient Algorithm

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

Families of update algorithms

parameter divergence	name of family	update algs.
$\|\|\boldsymbol{w} - \boldsymbol{w}_t\|\|_2^2$	Grad. Desc.	Widrow Hoff (LMS) Lin. Least Squ. Backprop. Perceptron Alg. kernel based algs.,...
$\sum_{i=1}^{n} w_i \ln \frac{w_i}{w_{t,i}}$	Exponentiated Gradient Alg.	expert algs Normalized Winnow "AdaBoost"

Families of update algorithms (cont)

$\sum_{i=1}^{n} w_i \ln \frac{w_i}{w_{t,i}} + w_{t,i} - w_i$	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	-	-

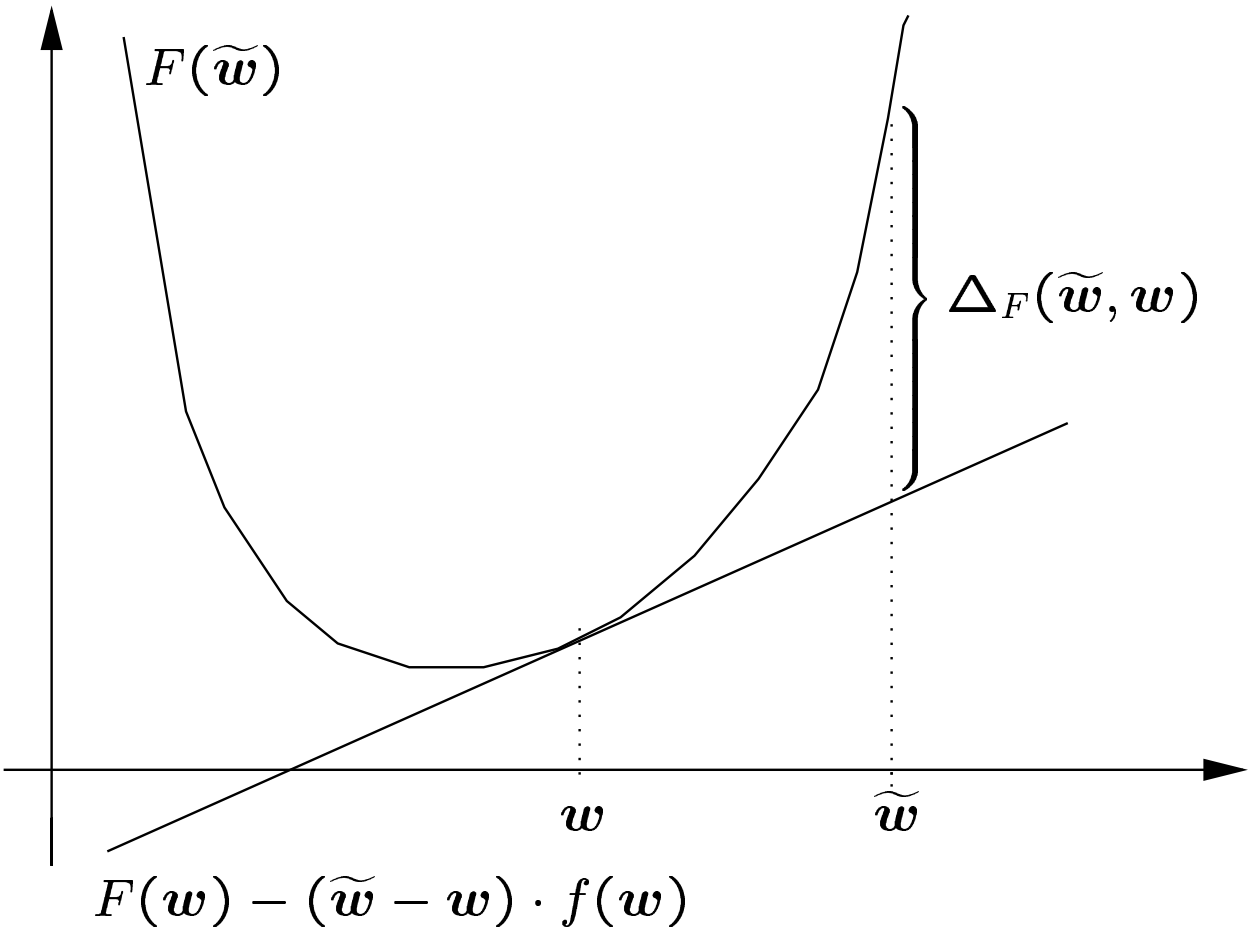
Members of different families exhibit different behavior

Bregman Divergences [Br,CL,Cs]

For any differentiable convex function F

$$\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) = F(\widetilde{\boldsymbol{w}}) - F(\boldsymbol{w}) - (\widetilde{\boldsymbol{w}} - \boldsymbol{w}) \cdot \underbrace{\nabla_{\boldsymbol{w}} F(\boldsymbol{w})}_{f(\boldsymbol{w})}$$

$$= F(\widetilde{\boldsymbol{w}}) - \begin{array}{c}\text{supporting hyperplane} \\ \text{through } (\boldsymbol{w}, F(\boldsymbol{w}))\end{array}$$



Bregman Divergences, Simple Properties [AW]

1. $\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w})$ is convex in $\widetilde{\boldsymbol{w}}$

2. $\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) \geq 0$
   If F convex equality holds iff $\widetilde{\boldsymbol{w}} = \boldsymbol{w}$

3. $\nabla_{\widetilde{\boldsymbol{w}}} \Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) = f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w})$

4. Usually not symmetric: $\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) \neq \Delta_F(\boldsymbol{w}, \widetilde{\boldsymbol{w}})$

5. Linearity (for $a \geq 0$):
   $\Delta_{F+a\,H}(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) = \Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) + a\,\Delta_H(\widetilde{\boldsymbol{w}}, \boldsymbol{w})$

6. Unaffected by linear terms ($a \in \mathbf{R}$, $\boldsymbol{b} \in \mathbf{R}^n$):
   $\Delta_{H+a\widetilde{\boldsymbol{w}}+\boldsymbol{b}}(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) = \Delta_H(\widetilde{\boldsymbol{w}}, \boldsymbol{w})$

7. $\Delta_F(\boldsymbol{w}_1, \boldsymbol{w}_2) + \Delta_F(\boldsymbol{w}_2, \boldsymbol{w}_3)$
   $= \Delta_F(\boldsymbol{w}_1, \boldsymbol{w}_3) + (\boldsymbol{w}_1 - \boldsymbol{w}_2) \cdot (f(\boldsymbol{w}_3) - f(\boldsymbol{w}_2))$

Examples

Squared Euclidean Distance

$$
\begin{aligned}
F(\boldsymbol{w}) &= ||\boldsymbol{w}||_2^2/2 \\
f(\boldsymbol{w}) &= \boldsymbol{w} \\
\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) &= ||\widetilde{\boldsymbol{w}}||_2^2/2 - ||\boldsymbol{w}||_2^2/2 - (\widetilde{\boldsymbol{w}} - \boldsymbol{w}) \cdot \boldsymbol{w} \\
&= ||\widetilde{\boldsymbol{w}} - \boldsymbol{w}||_2^2/2
\end{aligned}
$$

(Unnormalized) Relative Entropy

$$
\begin{aligned}
F(\boldsymbol{w}) &= \sum_i (w_i \ln w_i - w_i) \\
f(\boldsymbol{w}) &= \ln \boldsymbol{w} \\
\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) &= \sum_i \left(\widetilde{w_i} \ln \frac{\widetilde{w_i}}{w_i} + w_i - \widetilde{w_i} \right)
\end{aligned}
$$

Examples (cont) [GLS,GL]

p-norm Algs (q is dual to p)

$$F(\boldsymbol{w}) = \tfrac{1}{2}||\boldsymbol{w}||_q^2$$

$$f(\boldsymbol{w}) = \nabla \tfrac{1}{2}||\boldsymbol{w}||_q^2$$

$$\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) = \tfrac{1}{2}||\widetilde{\boldsymbol{w}}||_q^2 + \tfrac{1}{2}||\boldsymbol{w}||_q^2 - \widetilde{\boldsymbol{w}} \cdot f(\boldsymbol{w})$$

When $p = q = 2$ this reduces to squared Euclidean distance (Widrow-Hoff).

General Motivation of Updates [KW]

Trade-off between two divergences:

$$\boldsymbol{w}_{t+1} = \text{argmin}_{\boldsymbol{w}}(\underbrace{\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)}_{weight\ \ domain} + \eta_t \underbrace{L_t(\boldsymbol{w})}_{label\ \ domain})$$

$\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)$ is regularization term and serves as measure of progress in the analysis.

When loss L is convex (in $\boldsymbol{w}$)

$$\nabla_{\boldsymbol{w}}(\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) + \eta_t L_t(\boldsymbol{w})) = 0$$

iff

$$f(\boldsymbol{w}) - f(\boldsymbol{w}_t) + \eta_t \underbrace{\nabla L_t(\boldsymbol{w})}_{\approx \nabla L_t(\boldsymbol{w}_t)} = 0$$

$\Rightarrow$

$$\boldsymbol{w}_{t+1} = f^{-1}\left(f(\boldsymbol{w}_t) - \eta_t \nabla L_t(\boldsymbol{w}_t)\right)$$

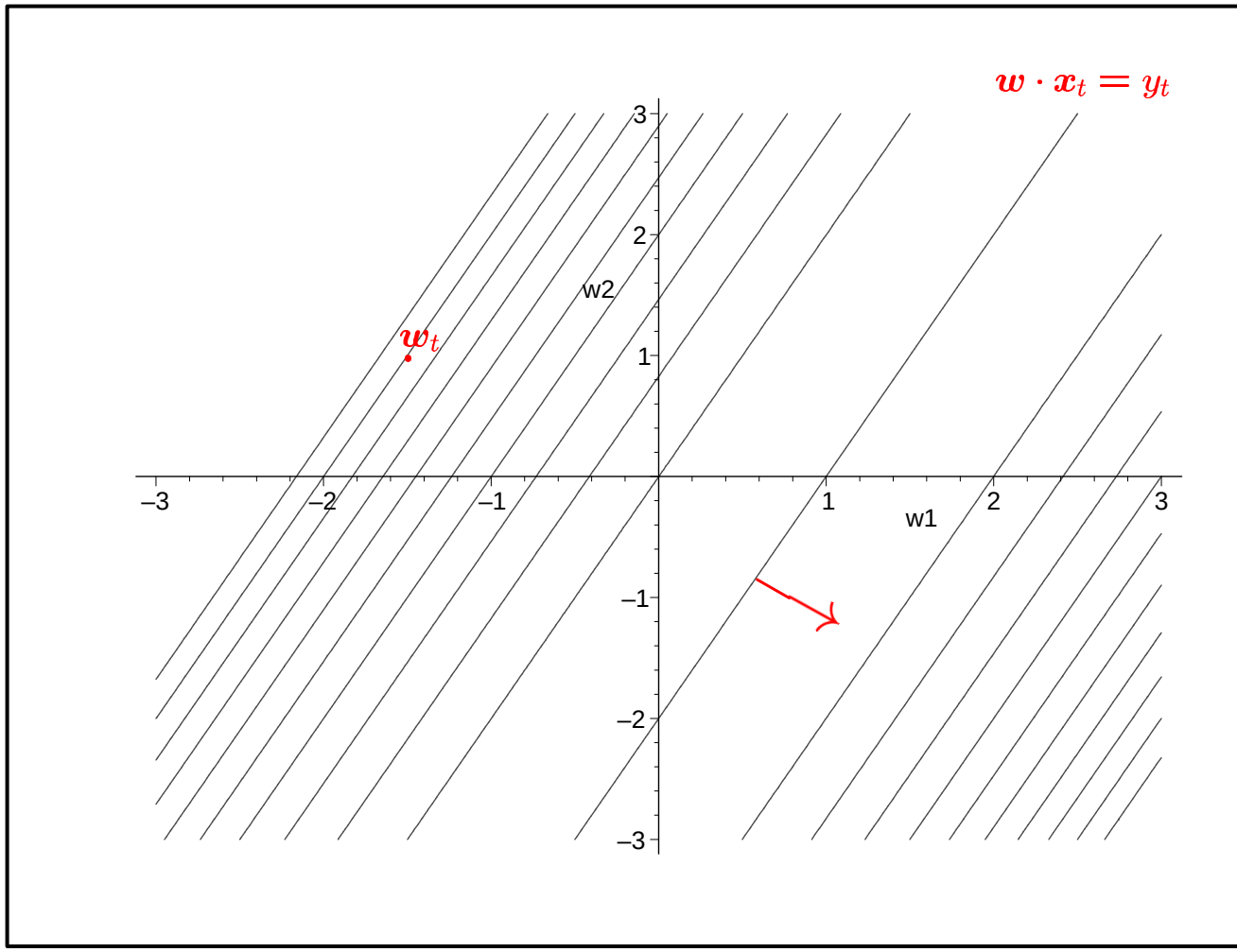
Square Loss

$$L_t(\boldsymbol{w}) = (y_t - \boldsymbol{w} \cdot \boldsymbol{x}_t)^2/2$$

$\boldsymbol{w}_t = (-3/2, 1)$

$\boldsymbol{x}_t = (1, -0.5)$

$y_t = 1$



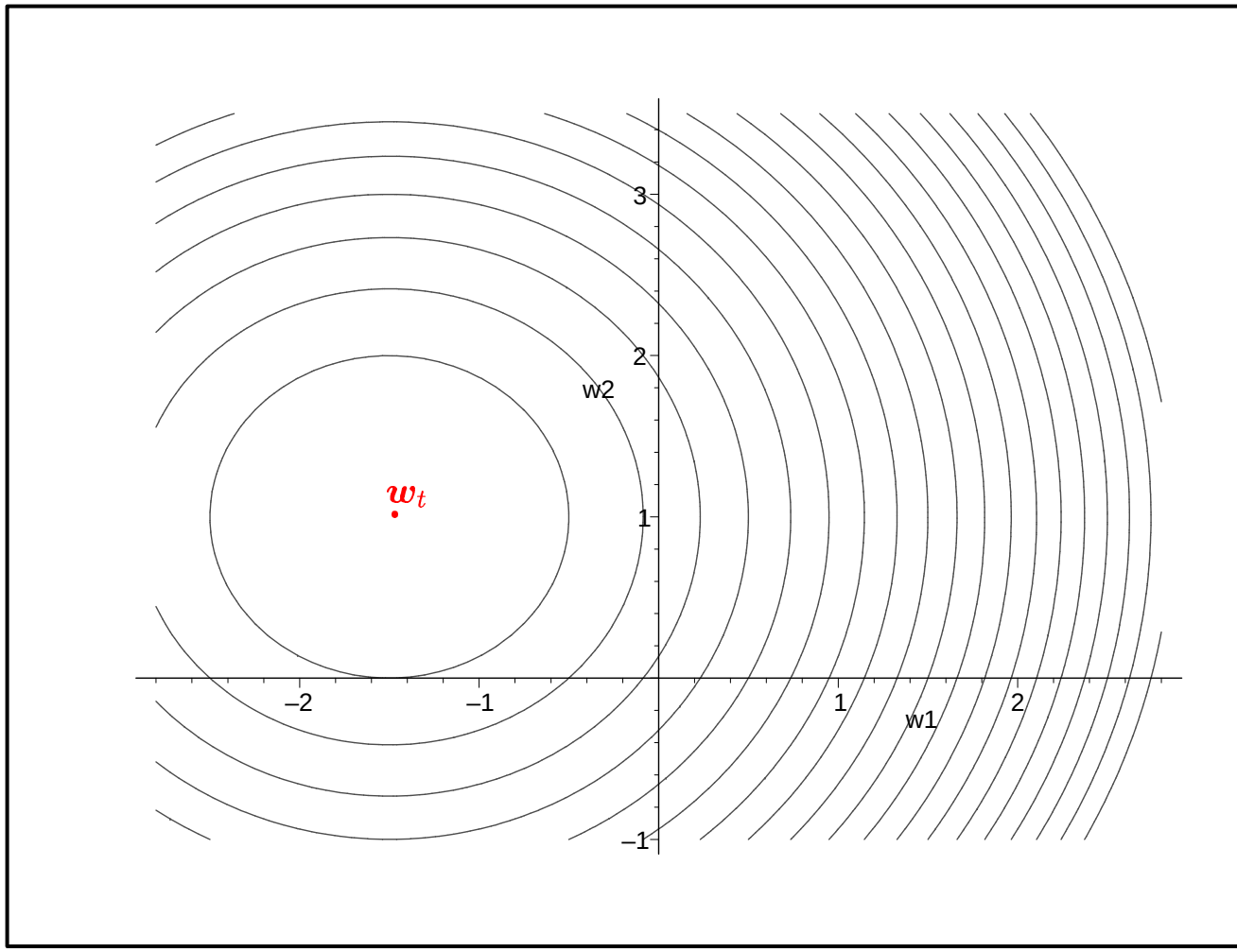
Divergence: Euclidean Distance Squared

$$\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) = ||\boldsymbol{w} - \boldsymbol{w}_t||_2^2/2$$

$\boldsymbol{w}_t = (-3/2, 1)$

$\boldsymbol{x}_t = (1, -0.5)$

$y_t = 1$



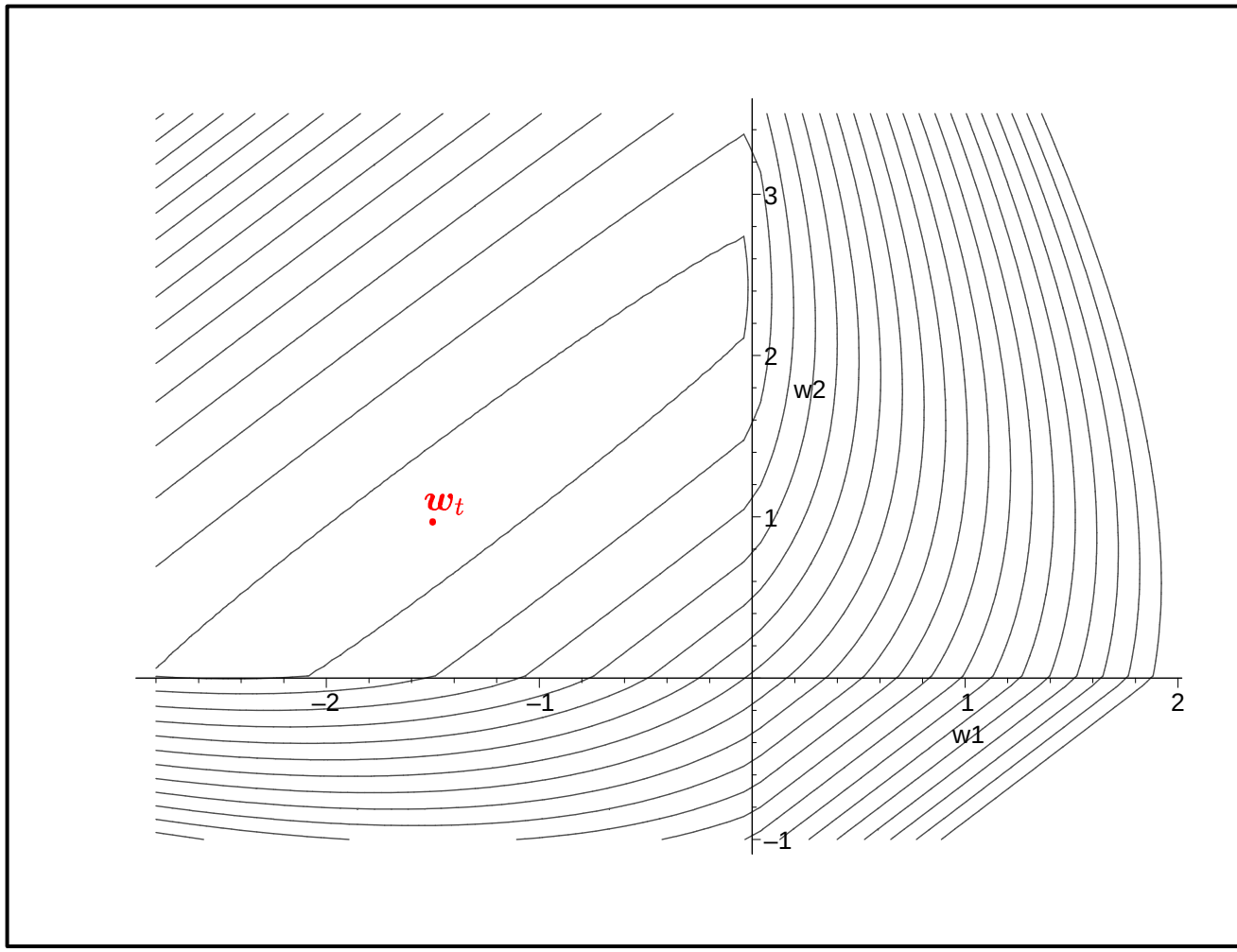
Divergence: 0-norm alg. divergence

$\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)$ is 10-norm alg. divergence

$\boldsymbol{w}_t = (-3/2, 1)$

$\boldsymbol{x}_t = (1, -0.5)$

$y_t = 1$



Loss + η Divergence

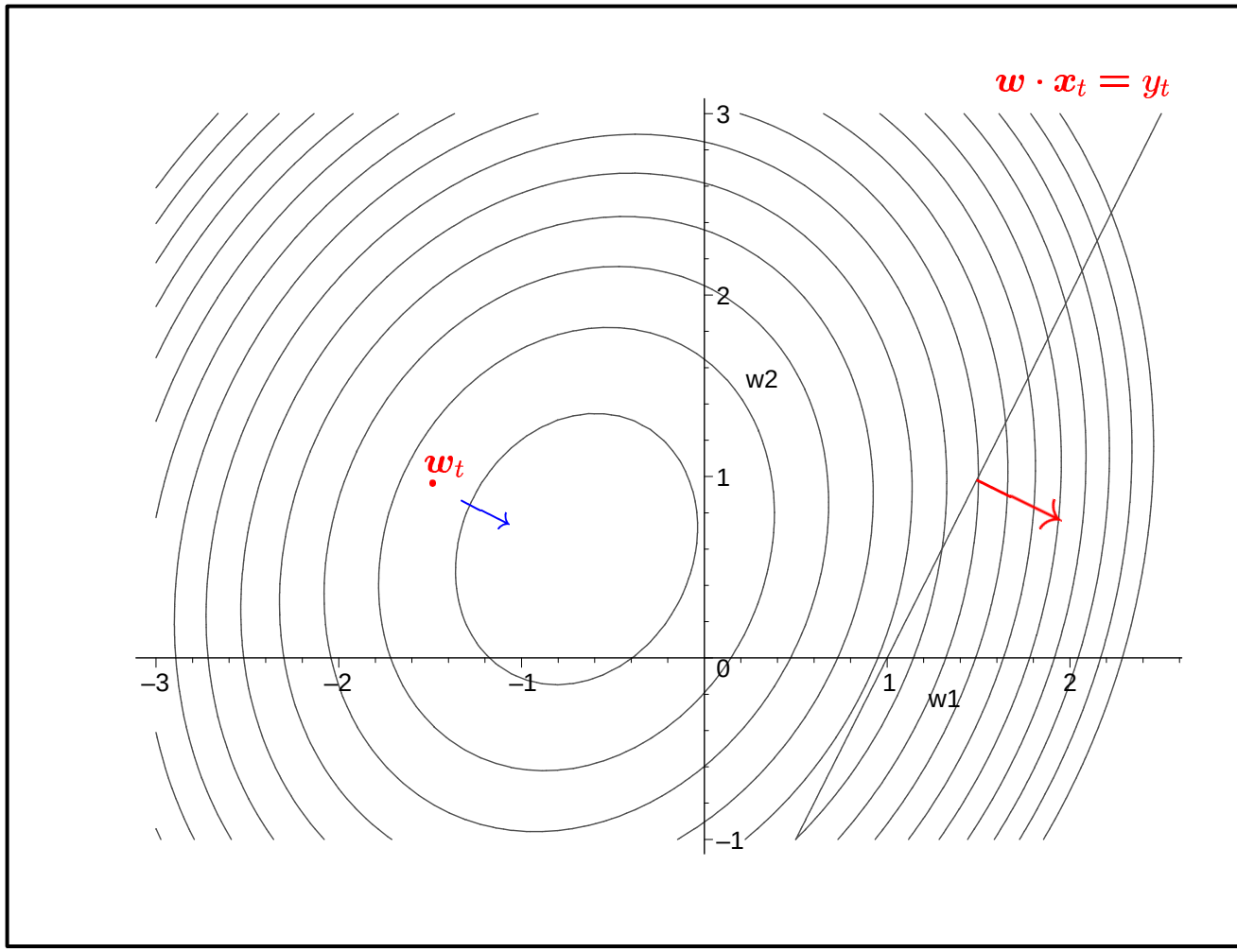
$\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)$
$= ||\boldsymbol{w} - \boldsymbol{w}_t||_2^2/2$

$\boldsymbol{w}_t = (-3/2, 1)$

$\boldsymbol{x}_t = (1, -0.5)$

$y_t = 1$

$\eta = 0.2$



Loss + η Divergence

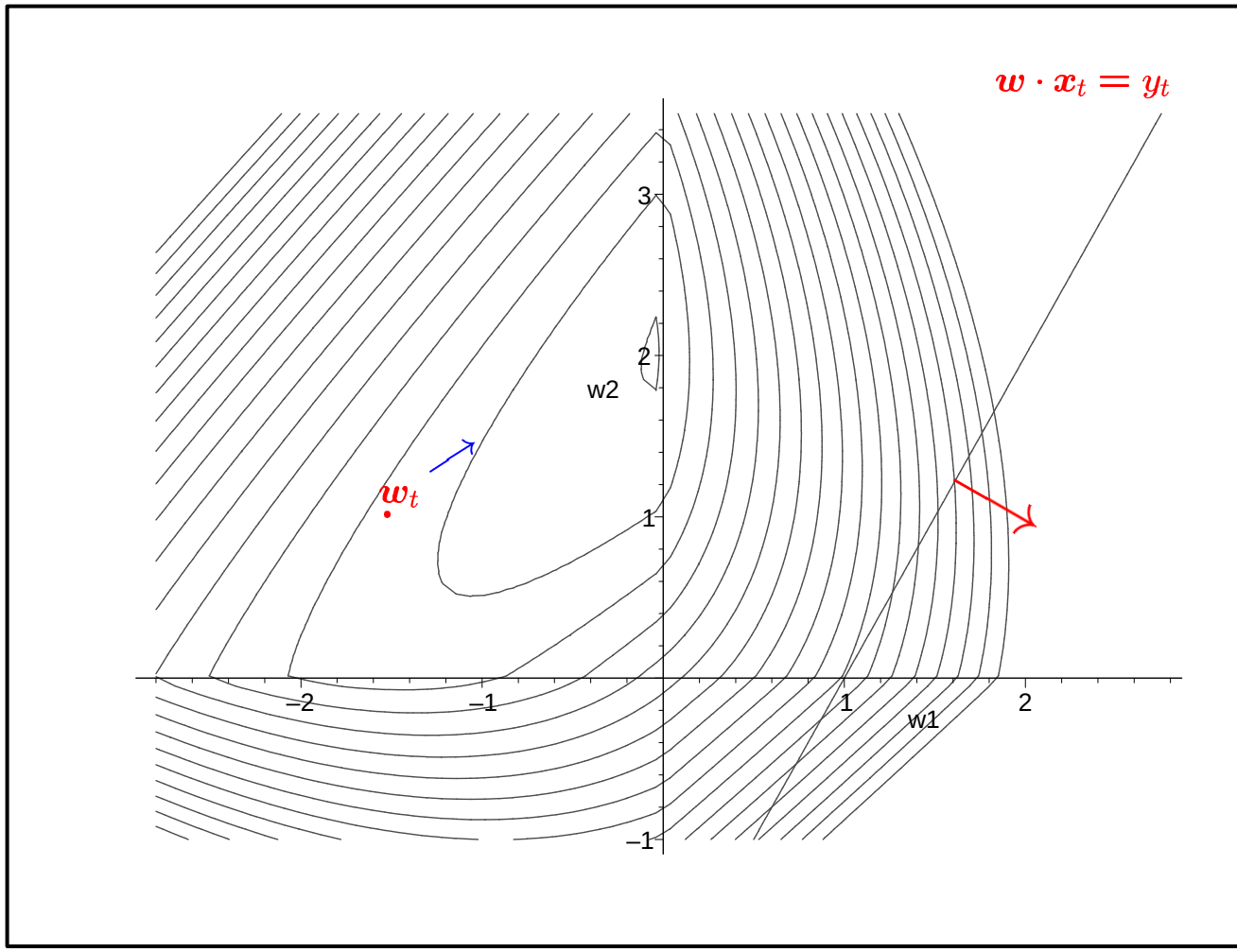
$\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)$
is 10-norm alg.
divergence

$\boldsymbol{w}_t = (-3/2, 1)$

$\boldsymbol{x}_t = (1, -0.5)$

$y_t = 1$

$\eta = 0.2$

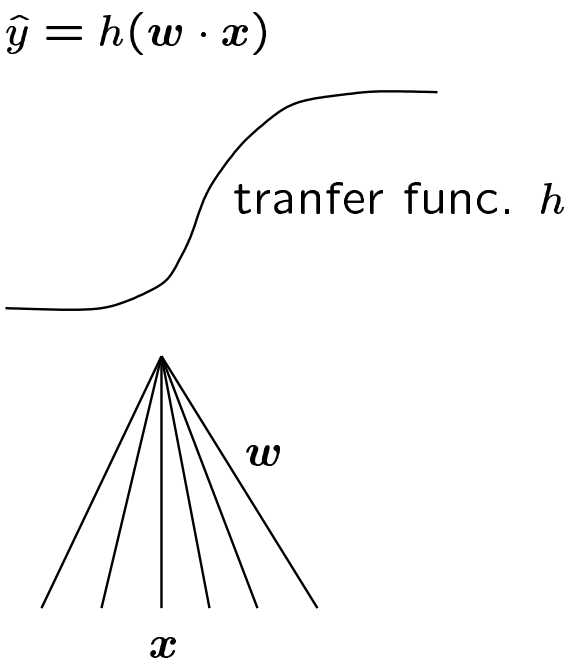


Characterization of algs. i.t.o. link function $f = \nabla F$ [WJ]

$$\boldsymbol{w}_{t+1} = f^{-1}\left(f(\boldsymbol{w}_t) - \eta_t \nabla L_t(\boldsymbol{w}_t)\right)$$

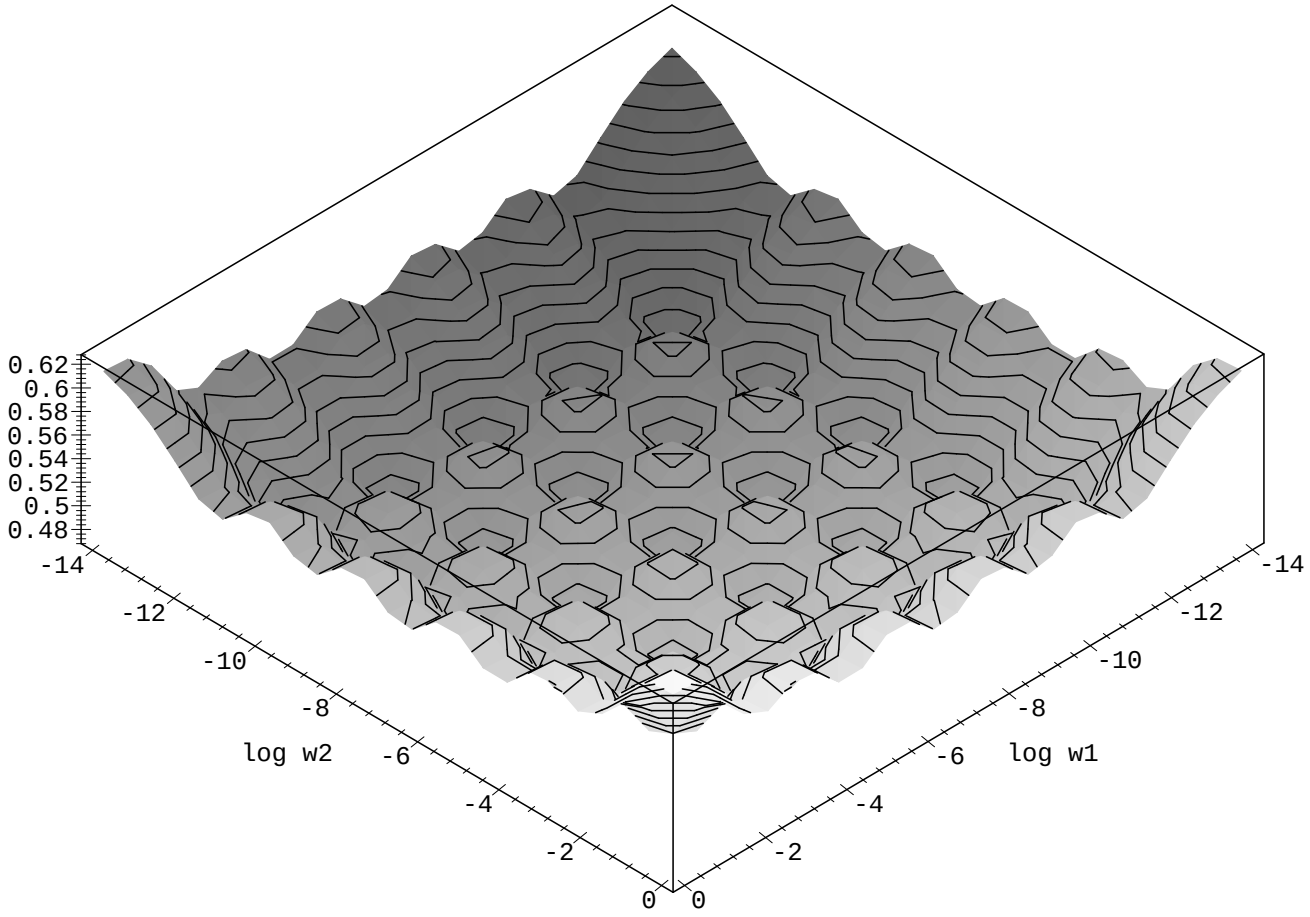
Alg.	$f(\boldsymbol{w})$	Domain of $\boldsymbol{w}$
GD	$f(\boldsymbol{w}) = \boldsymbol{w}$	$\boldsymbol{w} \in \mathbf{R}^n$
BEG	$f(\boldsymbol{w}) = \ln \frac{\boldsymbol{w}}{\mathbf{1}-\boldsymbol{w}}$	$\boldsymbol{w} \in [0,1]^n$
EG	$f(\boldsymbol{w}) = \ln \frac{\boldsymbol{w}}{1-\|\|\boldsymbol{w}\|\|_1}$	$\boldsymbol{w} \in [0,1]^{n-1}$, $\|\|\boldsymbol{w}\|\|_1 \leq 1$
p-norm	$f(\boldsymbol{w}) = \nabla \frac{1}{2}\|\|\boldsymbol{w}\|\|_q^2$ where $\frac{1}{p} + \frac{1}{q} = 1$	$\boldsymbol{w} \in \mathbf{R}^n$

Bregman Divergences
Lead to Good Loss Functions



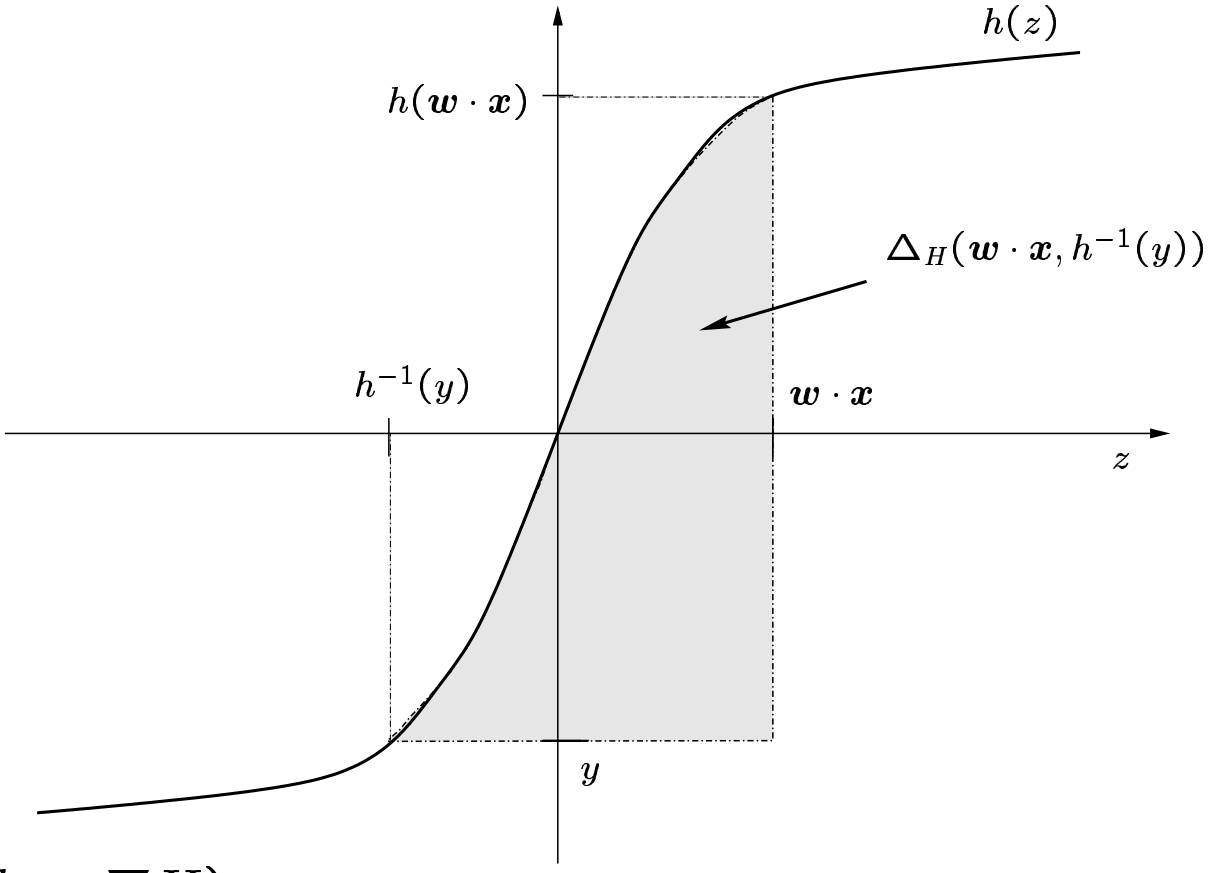
- Sigmoid function $h(z) = \frac{1}{1+e^{-z}}$

- For a set of examples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)$ total loss $\sum_{t=1}^{T} (h(\boldsymbol{w} \cdot \boldsymbol{x}) - y_t)^2/2$ can have exponentially many minima in weight space [Bu,AHW]



Want loss that is convex in $\boldsymbol{w}$

Bregman Divergences Lead to Good Loss Functions (cont)



$(h = \nabla H)$

$$\int_{h^{-1}(y)}^{\boldsymbol{w}\cdot\boldsymbol{x}} (h(z) - y)\, dz$$

$$= H(\boldsymbol{w}\cdot\boldsymbol{x}) - H(h^{-1}(y)) - (\boldsymbol{w}\cdot\boldsymbol{x} - h^{-1}(y))\, y$$

$$= \Delta_H(\boldsymbol{w}\cdot\boldsymbol{x}, h^{-1}(y))$$

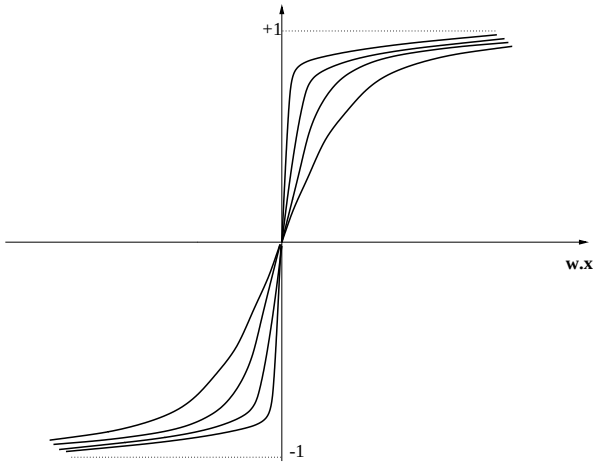
Use $\Delta_H(\boldsymbol{w} \cdot \boldsymbol{x}, h^{-1}(y))$ as loss of $\boldsymbol{w}$ on $(\boldsymbol{x}, y)$

Called matching loss for h [AHW,HKW]

Matching loss is convex in $\boldsymbol{w}$

transfer f. $h(z)$	$H(z)$	match. loss $d_H(\boldsymbol{w} \cdot \boldsymbol{x}, h^{-1}(y)$
z	$\frac{1}{2}z^2$	$\frac{1}{2}(\boldsymbol{w} \cdot \boldsymbol{x} - y)^2$ square loss
$\frac{e^z}{1+e^z}$	$\ln(1+e^z)$	$\ln(1+e^{\boldsymbol{w}\cdot\boldsymbol{x}}) - y\boldsymbol{w} \cdot \boldsymbol{x}$ $+y \ln y + (1-y)\ln(1-y)$ logistic loss
$\text{sign}(z)$	$\lvert z \rvert$	$\max\{0, -y\boldsymbol{w} \cdot \boldsymbol{x}\}$ hinge loss

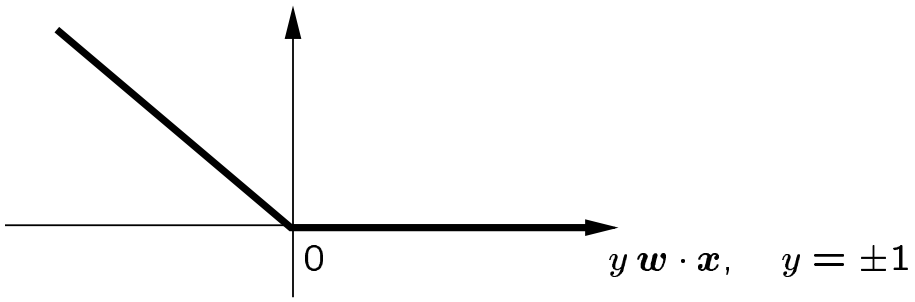
For transfer function $h(z) = \text{sign}(z)$



$H(z) = |z|$

Matching loss is hinge loss [GW]

$$HL(\boldsymbol{w} \cdot \boldsymbol{x}, h^{-1}(y)) = \max\{0, -y\, \boldsymbol{w} \cdot \boldsymbol{x}\}$$



Convex in $\boldsymbol{w}$ but not differentiable

Motivation of linear threshold algs

Gradient descent with Hinge Loss	Perceptron
Expon. gradient with Hinge Loss	Normalized Winnow

Known linear threshold algs for ± 1-class are gradient-based algs with hinge loss

Trade-off between two divergences [KW]

$$\boldsymbol{w}_{t+1} = \operatorname{argmin}_{\boldsymbol{w}}\Big(\underbrace{\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)}_{\substack{\text{parameter}\\ \text{divergence}}} + \eta_t \underbrace{\Delta_H(\boldsymbol{w} \cdot \boldsymbol{x}_t, h^{-1}(y_t))}_{\substack{\text{matching}\\ \text{loss}}}\Big)$$

Both divergences are convex in $\boldsymbol{w}$

$$\boldsymbol{w}_{t+1} = f^{-1}\left(f(\boldsymbol{w}_t) - \eta_t(h(\boldsymbol{w}_t \cdot \boldsymbol{x}_t) - y_t)\boldsymbol{x}_t\right)$$

Generalization of the "delta"-rule

Projections

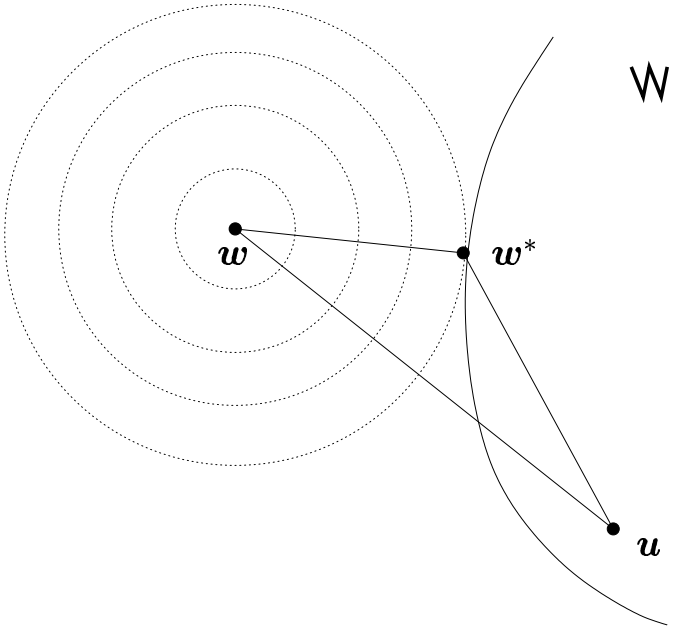
$$\boldsymbol{w}_{t+1} = \text{argmin}_{\boldsymbol{w}}(\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) + \eta(\boldsymbol{w} \cdot \boldsymbol{x}_t - y_t)^2)$$

When η is large then $\boldsymbol{w}_{t+1}$ is projection of $\boldsymbol{w}_t$ onto plane $\boldsymbol{w} \cdot \boldsymbol{x}_t = y_t$

$$\boldsymbol{w}_{t+1} = \text{argmin}_{\{\boldsymbol{w} \,:\, \boldsymbol{w}_t \cdot \boldsymbol{x}_t = y_t\}} \; \Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)$$

The AdaBoost update of the probability vector $\boldsymbol{w}_t$ on the examples is a projection w.r.t. divergence $\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) = \sum_i w_i \ln \frac{w_i}{w_{t,i}}$ [CKW,La,KW,CSS]

A Pythagorean Theorem [Br,Cs,A,HW]



$\boldsymbol{w}^*$ is projection of $\boldsymbol{w}$ onto convex set $\mathcal{W}$ w.r.t. Bregman divergence Δ_F:

$$\boldsymbol{w}^* = \text{argmin}_{\boldsymbol{u}\in\mathcal{W}} \Delta_F(\boldsymbol{u}, \boldsymbol{w})$$

Th:

$$\Delta_F(\boldsymbol{u}, w) \geq \Delta_F(\boldsymbol{u}, \boldsymbol{w}^*) + \Delta_F(\boldsymbol{w}^*, \boldsymbol{w})$$

How do we prove relative loss bounds?

Loss: $L_t(\boldsymbol{w}) = L((\boldsymbol{x}_t, y_t), \boldsymbol{w})$ convex in $\boldsymbol{w}$

Divergence: $\Delta_F(\boldsymbol{u}, \boldsymbol{w})$
$= F(\boldsymbol{u}) - F(\boldsymbol{w}) - (\boldsymbol{u} - \boldsymbol{w}) \cdot f(\boldsymbol{w})$

Update: $f(\boldsymbol{w}_{t+1}) - f(\boldsymbol{w}_t) = -\eta \nabla_{\boldsymbol{w}} L_t(\boldsymbol{w}_t)$

$$L_t(\boldsymbol{u})$$

$$\overbrace{\geq}^{\text{convexity}} \quad L_t(\boldsymbol{w}_t) + (\boldsymbol{u} - \boldsymbol{w}_t) \cdot \underbrace{\nabla_{\boldsymbol{w}} L_t(\boldsymbol{w}_t)}_{\text{update}}$$

$$= L_t(\boldsymbol{w}_t) - \frac{1}{\eta} \underbrace{(\boldsymbol{u} - \boldsymbol{w}_t) \cdot (f(\boldsymbol{w}_{t+1}) - f(\boldsymbol{w}_t))}_{\text{prop. 7 of } \Delta_F}$$

$$= L_t(\boldsymbol{w}_t) + \frac{1}{\eta} \left(\Delta_F(\boldsymbol{u}, \boldsymbol{w}_{t+1}) - \Delta_F(\boldsymbol{u}, \boldsymbol{w}_t) - \Delta_F(\boldsymbol{w}_t, \boldsymbol{w}_{t+1}) \right)$$

Summing over t [WJ,KW]

$$\sum_t L_t(\boldsymbol{w}_t) \leq \sum_t L_t(\boldsymbol{u}) + \frac{1}{\eta}\sum_t \Big(\Delta_F(\boldsymbol{u}, \boldsymbol{w}_t) - \Delta_F(\boldsymbol{u}, \boldsymbol{w}_{t+1}) + \Delta_F(\boldsymbol{w}_t, \boldsymbol{w}_{t+1}) \Big)$$

$$\leq \sum_t L_t(\boldsymbol{u}) + \frac{1}{\eta}\Big(\Delta_F(\boldsymbol{u}, \boldsymbol{w}_1) - \underbrace{\Delta_F(\boldsymbol{u}, \boldsymbol{w}_{T+1})}_{\geq 0} \Big) + \frac{1}{\eta}\sum_t \Delta_F(\boldsymbol{w}_t, \boldsymbol{w}_{t+1})$$

$$\sum_t L_t(\boldsymbol{w}_t) \leq \sum_t L_t(\boldsymbol{u}) + \frac{1}{\eta}\Delta_F(\boldsymbol{u}, \boldsymbol{w}_1) + \frac{1}{\eta}\sum_t \Delta_F(\boldsymbol{w}_t, \boldsymbol{w}_{t+1})$$

Any convex loss and any Bregman divergence!

Key step:

Relate $\Delta_F(\boldsymbol{w}_t, \boldsymbol{w}_{t+1})$ to loss $L_t(\boldsymbol{w}_t)$

Loss & divergence dependent

Get $\Delta_F(\boldsymbol{w}_t, \boldsymbol{w}_{t+1}) \leq \text{const}\, L_t(\boldsymbol{w}_t)$

Then solve for $\sum_t L_t(\boldsymbol{w}_t)$

Yield bounds of the form

$$\sum_t L_t(\boldsymbol{w}_t) \leq a \sum_t L_t(\boldsymbol{u}) + b\, \Delta_F(\boldsymbol{u}, \boldsymbol{w}_1)$$

a, b constants, $a > 1$.

Regret bounds ($a = 1$):

time changing η, subtler analysis [AG]

Some Bounds [KW,GLS,GL]
(Linear Regression with Square Loss)

Gradient Descent

$$\sum_t L_t(\boldsymbol{w}_t) \leq (1+c) \sum_t L_t(\boldsymbol{u}) + \frac{1+c}{c} X_2^2 U_2^2$$

$||\boldsymbol{x}_t||_2 \leq X_2,\ ||\boldsymbol{u}||_2 \leq U_2,\ c > 0$

Scaled Exponentiated Gradient

$$\sum_t L_t(\boldsymbol{w}_t) \leq (1+c) \sum_t L_t(\boldsymbol{u}) + \frac{1+c}{c} \ln n\, X_\infty^2 U_1^2$$

$||\boldsymbol{x}_t||_\infty \leq X_\infty,\ ||\boldsymbol{u}||_1 \leq U_1,\ c > 0$

p-norm alg

$$\sum_t L_t(\boldsymbol{w}_t) \leq (1+c) \sum_t L_t(\boldsymbol{u}) + \frac{1+c}{c} (p-1)\, X_p^2\, U_q^2$$

$||\boldsymbol{x}_t||_p \leq X_p,\ ||\boldsymbol{u}||_q \leq U_q,\ c > 0$

Generalization Bounds

Loss bounds for on-line algs

$\Rightarrow$

Bounds on expected loss in i.i.d. case

- On-line to batch [L]
- Leave-one out [HW,CB+,KW]

On-line to batch

Simplest case: 0-1 Loss, binary y_t

$S = (\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T) \sim D^T$

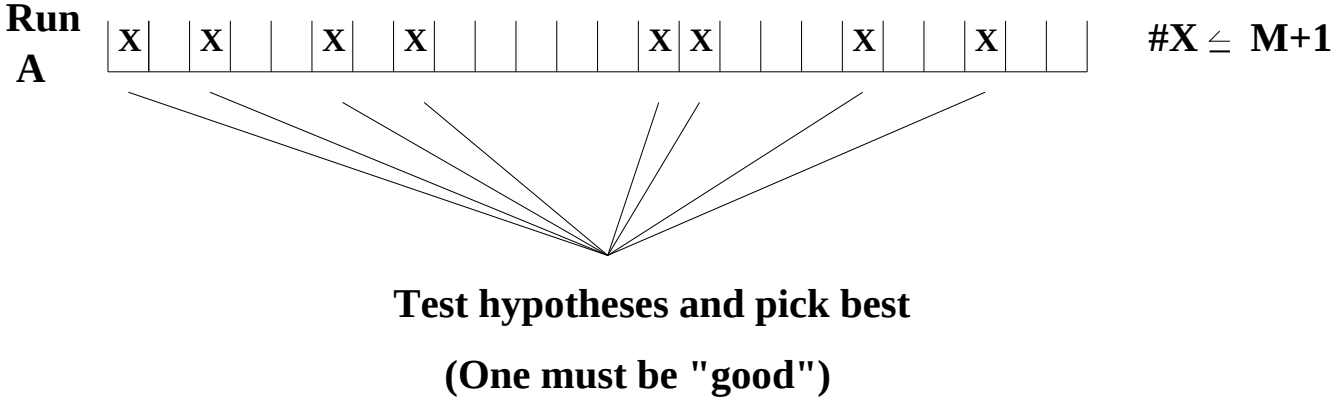
Alg A updates only if mistake

If A has mistake bound M then $\exists$ alg A' :

$$Pr_{S \sim D^T}\left(err_D(A'(S)) \leq \epsilon\right) \geq 1 - \delta$$

$$err_D(A'(S)) = Pr_{(\boldsymbol{x},y)\sim D}\left(A'(S)(\boldsymbol{x}) \neq y\right)$$

$$T = O\left(\frac{M}{\epsilon} + \frac{1}{\epsilon}\ln\frac{1}{\delta}\right)$$



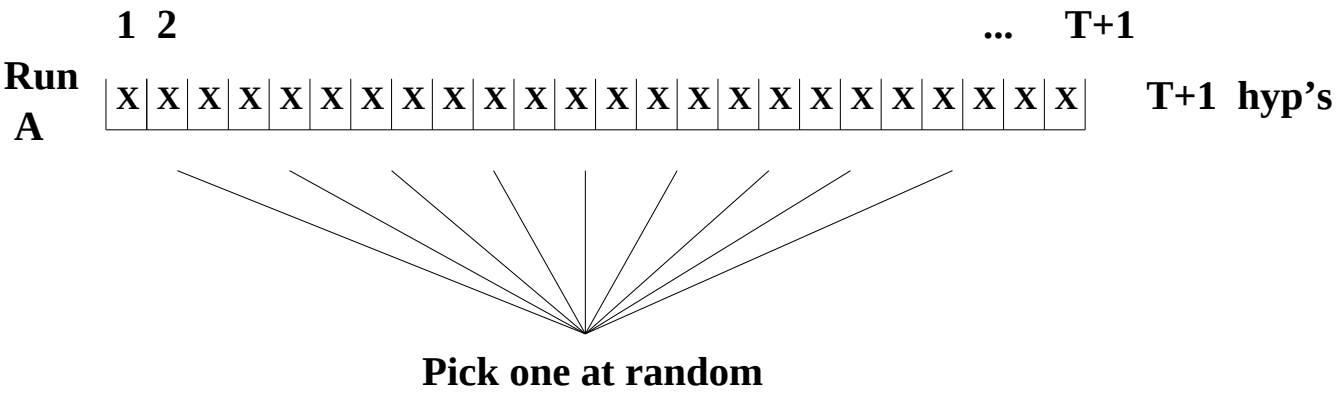
(Simplest) Leave One Out [HW]

Loss $L : \mathbf{R}^2 \to \mathbf{R}$

$S = (\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T) \sim D^T$

Given alg A, want to bound

$$E_{(\boldsymbol{x},y)\sim D}[L(y, A(\boldsymbol{x}))]$$



call it h_i and predict
on new instance $\boldsymbol{x}$ as $h_i(\boldsymbol{x})$

(Simplest) Leave One Out (cont)

$$Loo_A(S)(\boldsymbol{x}) = h_i(\boldsymbol{x}) \quad \sim \quad \frac{1}{T+1}$$

Then:

$$E\left[L(\boldsymbol{y}, Loo_A(\boldsymbol{S})(\boldsymbol{x}))\right]$$

$$\leq \frac{E_{S \sim D^{T+1}}[\text{cum. loss of } A]}{T+1}$$

$$\leq \frac{\text{worst-case loss bound}}{T+1}$$

Applied to the Perceptron Alg. [FS]

Where do Bregman divergences come from?

- Exponential family of distributions
- Inherent duality

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

primal param.		dual param.
$\boldsymbol{w}_t$	$\xrightarrow{f}$	$f(\boldsymbol{w}_t)$
$\boldsymbol{w}_{t+1}$	$\xleftarrow{f^{-1}}$	$-\eta \nabla L_t(\boldsymbol{w}_t)$

Exponential Family of Distributions

- Parametric density functions

$$P_G(\boldsymbol{x}|\boldsymbol{\theta}) = e^{\boldsymbol{\theta}\cdot\boldsymbol{x} - G(\boldsymbol{\theta})}\, P_0(\boldsymbol{x})$$

- $\boldsymbol{\theta}$ and $\boldsymbol{x}$ vectors in R^d

- Cumulant function $G(\boldsymbol{\theta})$
  assures normalization

$$G(\boldsymbol{\theta}) = \ln \int e^{\boldsymbol{\theta}\cdot\boldsymbol{x}}\, P_0(\boldsymbol{x})\, d\boldsymbol{x}$$

- $G(\boldsymbol{\theta})$ is convex function
  on convex set $\boldsymbol{\Theta} \subseteq R^d$

- G characterizes members of the family

- $\boldsymbol{\theta}$ is *natural* parameter

- Expectation parameter

$$\boldsymbol{\mu} = \int_{\boldsymbol{x}} \boldsymbol{x} P_G(\boldsymbol{x}|\boldsymbol{\theta}) d\boldsymbol{x} = E_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta})$$

  where $g(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$

- Second convex function $F(\boldsymbol{\mu})$ on space $g(\boldsymbol{\Theta})$

$$F(\boldsymbol{\mu}) \;\; = \;\; \boldsymbol{\theta} \cdot \boldsymbol{\mu} - G(\boldsymbol{\theta})$$

- $G(\boldsymbol{\theta})$ and $F(\boldsymbol{\mu})$ are *dual* convex functions

- Let $f(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}} F(\boldsymbol{\mu})$

- $f(\boldsymbol{\mu}) = g^{-1}(\boldsymbol{\mu})$

Summary

natural par.		expectation par.
$\boldsymbol{\theta}$	$\xrightarrow{g}$ $\xleftarrow{f}$	$\boldsymbol{\mu}$
$G(\boldsymbol{\theta})$		$F(\boldsymbol{\mu})$

- $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ are dual parameters

- Parameter transformations
  $g(\boldsymbol{\theta}) = \boldsymbol{\mu}$ and $f(\boldsymbol{\mu}) = \boldsymbol{\theta}$ [A,BN]

Gaussian (unit variance)

$$\begin{aligned} P(\boldsymbol{x}|\boldsymbol{\theta}) &\sim e^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{x})^2} \\ &= e^{\boldsymbol{\theta}\cdot\boldsymbol{x}-\frac{1}{2}\boldsymbol{\theta}^2} e^{\frac{1}{2}\boldsymbol{x}^2} \end{aligned}$$

Cumulant function: $G(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^2$

Parameter transformations:

$g(\boldsymbol{\theta}) = \boldsymbol{\theta} = \boldsymbol{\mu}$ and $f(\boldsymbol{\mu}) = \boldsymbol{\mu} = \boldsymbol{\theta}$

Dual convex function: $\begin{aligned} F(\boldsymbol{\mu}) &= \boldsymbol{\theta}\cdot\boldsymbol{\mu} - G(\boldsymbol{\theta}) \\ &= \frac{1}{2}\boldsymbol{\mu}^2 \end{aligned}$

Square loss: $L_t(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta}_t - \boldsymbol{x}_t)^2$

Bernoulli

Examples x_t are coin flips in $\{0, 1\}$

$$P(x|\mu) = \mu^x(1-\mu)^{1-x}$$

μ is the probability (expectation) of 1

Natural parameter: $\theta = \ln \frac{\mu}{1-\mu}$

$$P(x|\theta) = \exp\left(\theta x - \ln(1+e^\theta)\right)$$

Cumulant function: $G(\theta) = \ln(1+e^\theta)$

Parameter transformations:

$$\mu = g(\theta) = \frac{e^\theta}{1+e^\theta} \text{ and } \theta = f(\mu) = \ln\frac{\mu}{1-\mu}$$

Dual function: $F(\mu) = \mu \ln \mu + (1-\mu)\ln(1-\mu)$

Log loss: $\begin{aligned} L_t(\theta) &= -x_t\theta + ln(1+e^\theta) \\ &= -x_t \ln \mu - (1-x_t)\ln(1-\mu) \end{aligned}$

Poisson

Examples x_t are natural numbers in $\{0, 1, \dots\}$

$$P(x|\mu) = \frac{e^{-\mu}\mu^x}{x!}$$

μ is expectation of x

Natural parameter: $\theta = \ln \mu$

$$P(x|\theta) = \exp\left(\theta x - e^{\theta}\right) \frac{1}{x!}$$

Cumulant function: $G(\theta) = e^{\theta}$

Parameter transformations:

$$\mu = g(\theta) = e^{\theta} \text{ and } \theta = f(\mu) = \ln \mu$$

Dual function: $F(\mu) = \mu \ln \mu - \mu$

$$\begin{aligned} \text{Loss:} \quad L_t(\theta) &= -x_t\theta + e^{\theta} + \ln x_t! \\ &= -x_t \ln \mu + \mu + \ln x_t! \end{aligned}$$

Bregman Divergences as Relative Entropies between Exponential Distributions

Let $P(\boldsymbol{x}|\boldsymbol{\theta})$ and $P(\boldsymbol{x}|\widetilde{\boldsymbol{\theta}})$ denote two distributions with cumulant function G

$$\Delta_G(\widetilde{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \int_{\boldsymbol{x}} P_G(\boldsymbol{x}|\boldsymbol{\theta}) \ln \frac{P_G(\boldsymbol{x}|\boldsymbol{\theta})}{P_G(\boldsymbol{x}|\widetilde{\boldsymbol{\theta}})} d\boldsymbol{x}$$

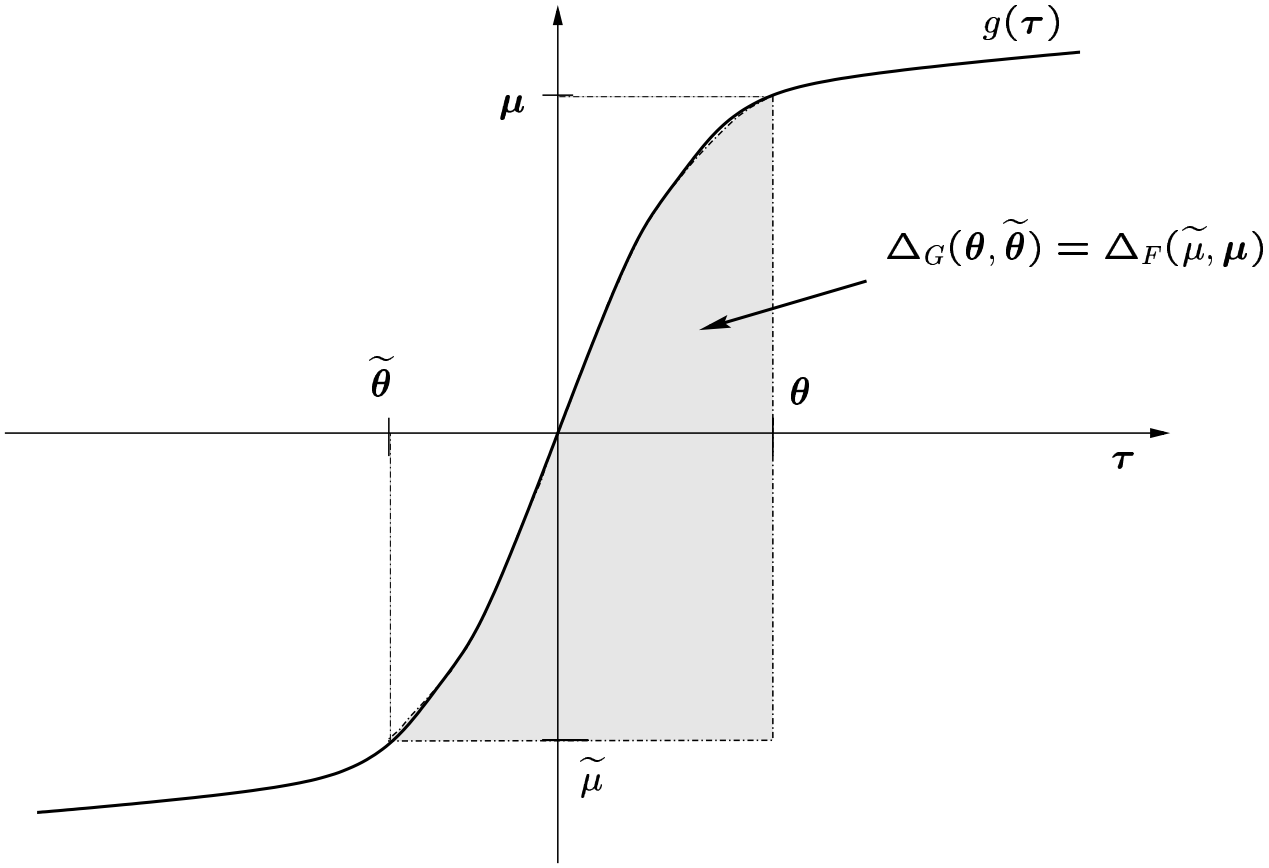
$$= G(\widetilde{\boldsymbol{\theta}}) - G(\boldsymbol{\theta}) - (\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}) \cdot \boldsymbol{\mu}$$

$$\overset{F(\boldsymbol{\mu})=\boldsymbol{\theta}\cdot\boldsymbol{\mu}-G(\boldsymbol{\theta})}{=} F(\boldsymbol{\mu}) - F(\widetilde{\boldsymbol{\mu}}) - (\boldsymbol{\mu} - \widetilde{\boldsymbol{\mu}}) \cdot \widetilde{\boldsymbol{\theta}}$$

$$= \Delta_F(\boldsymbol{\mu}, \widetilde{\boldsymbol{\mu}})$$

[A,BN,AW]

Area unchanged When Slide Flipped



$$
\begin{aligned}
\Delta_G(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) &= G(\boldsymbol{\theta}) - G(\widetilde{\boldsymbol{\theta}}) - (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}) \cdot g(\widetilde{\boldsymbol{\theta}}) \\
&= \int_{\widetilde{\boldsymbol{\theta}}}^{\boldsymbol{\theta}} (g(\boldsymbol{\tau}) - g(\widetilde{\boldsymbol{\theta}})) \cdot d\boldsymbol{\tau} \\
&\overset{\text{flip}}{=} \int_{\boldsymbol{\mu}}^{\widetilde{\boldsymbol{\mu}}} (f(\boldsymbol{\sigma}) - f(\boldsymbol{\mu})) \cdot d\boldsymbol{\sigma} \\
&= F(\widetilde{\boldsymbol{\mu}}) - F(\boldsymbol{\mu}) - (\widetilde{\boldsymbol{\mu}} - \boldsymbol{\mu}) \cdot f(\boldsymbol{\mu}) \\
&= \Delta_F(\widetilde{\boldsymbol{\mu}}, \boldsymbol{\mu})
\end{aligned}
$$

Dual divergence for Bernoulli

$$G(\boldsymbol{\theta}) = \ln(1 + e^{\theta}) \qquad F(\mu) = \mu \ln \mu + (1 - \mu) \ln(1 - \mu)$$

$$g(\theta) = \frac{e^{\theta}}{1+e^{\theta}} = \mu \qquad f(\mu) = \ln \frac{\mu}{1-\mu} = \theta$$

$$\Delta_G(\widetilde{\theta}, \theta) = \ln(1 + e^{\widetilde{\theta}}) - \ln(1 + e^{\theta}) - (\widetilde{\theta} - \theta)\frac{e^{\theta}}{1 + e^{\theta}}$$

$$\Delta_F(\mu, \widetilde{\mu}) = \mu \ln \frac{\mu}{\widetilde{\mu}} + (1 - \mu) \ln \frac{1 - \mu}{1 - \widetilde{\mu}}$$

Binary relative entropy

Dual divergence for Poisson

$$G(\boldsymbol{\theta}) = e^{\theta} \qquad F(\mu) = \mu \ln \mu - \mu$$

$$g(\theta) = e^{\theta} = \mu \quad f(\mu) = \ln \mu = \theta$$

$$\Delta_G(\widetilde{\theta}, \theta) = e^{\widetilde{\theta}} - e^{\theta} - (\widetilde{\theta} - \theta)e^{\theta}$$

$$\Delta_F(\mu, \widetilde{\mu}) = \mu \ln \frac{\mu}{\widetilde{\mu}} + \widetilde{\mu} - \mu$$

Unnormalized relative entropy

Dual matching loss for sigmoid tranfer func.

$$H(z) = \ln(1 + e^z) \qquad K(r) = r \ln r + (1 - r) \ln(1 - r)$$

$$h(z) = \frac{e^z}{1+e^z} = r \qquad k(r) = \ln \frac{r}{1-r} = z$$

$$K \text{ dual to } H \text{ and } k = h^{-1}$$

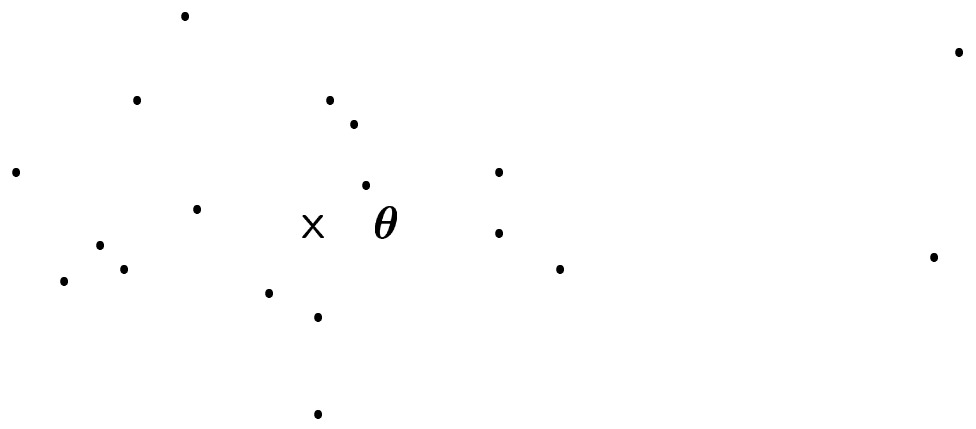
$$\Delta_H(\boldsymbol{w} \cdot \boldsymbol{x}, h^{-1}(y))$$
$$= \ln(1 + e^{\boldsymbol{w} \cdot \boldsymbol{x}}) - y \boldsymbol{w} \cdot \boldsymbol{x} + y \ln y + (1 - y) \ln(1 - y)$$

By duality logistic loss is same as entropic loss

$$\Delta_K(y, h(\boldsymbol{w} \cdot \boldsymbol{x}))$$
$$= y \ln \frac{y}{h(\boldsymbol{w} \cdot \boldsymbol{x})} + (1 - y) \ln \frac{1 - y}{1 - h(\boldsymbol{w} \cdot \boldsymbol{x})}$$

Gaussian density estimation

(Fixed variance)



Off-line versus on-line

- Loss on example $\boldsymbol{x}_t$

$$L_t(\boldsymbol{\theta}) = -\ln P(\boldsymbol{x}_t|\boldsymbol{\theta}) \quad = \quad \tfrac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\theta})^2$$

Derivation of Updates

- Want to bound

$$\sum_{t=1}^{T} L_t(\boldsymbol{\theta}_t) - \inf_{\boldsymbol{\theta}} L_{1..T}(\boldsymbol{\theta})$$

- Off-line algorithm has all T examples

$$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$$

- Setup for choosing best parameter setting

$$\boldsymbol{\theta}_B = \text{argmin}_{\boldsymbol{\theta}}(\eta_B^{-1} \underbrace{\Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_1)}_{\substack{\text{divergence} \\ \text{to initial}}} + \underbrace{L_{1..T}(\boldsymbol{\theta})}_{\substack{\text{total} \\ \text{loss}}})$$

  Here $\eta_B^{-1} > 0$ is a tradeoff parameter

- Equivalent to Bayesian MAP

  where $\eta_B^{-1}\Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_1)$ is the log of the conjugate prior

  and $L_{1..T}(\boldsymbol{\theta})$ is the log of the joint likelihood

- Alternate:
  $\eta_B^{-1}\Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_1)$ loss on initial set of examples

On-line Algorithm [AW]

- In trial t, the first t examples

  $$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$$

  have been presented

- Motivation for on-line parameter update: do as well as best off-line algorithm up to trial t

- At end of trial t algorithm minimizes

  $$\boldsymbol{\theta}_{t+1} = \text{argmin}_{\boldsymbol{\theta}} \, (\eta_1^{-1} \underbrace{\Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_1)}_{\substack{\text{divergence} \\ \text{to initial}}} + \underbrace{L_{1..t}(\boldsymbol{\theta})}_{\substack{\text{loss} \\ \text{so far}}})$$

  Tradeoff parameter $\eta_1^{-1} \geq 0$

Alternate Motivation of Same On-Line Update

$$\boldsymbol{\theta}_{t+1} = \text{argmin}_{\boldsymbol{\theta}}(\eta_t^{-1} \underbrace{\Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_t)}_{\substack{\text{divergence} \\ \text{to last}}} + \underbrace{L_t(\boldsymbol{\theta})}_{\substack{\text{current} \\ \text{loss}}})$$

where

$$\eta_t = \frac{1}{\eta_1^{-1} + t - 1}$$

Parameter Updates

Off-line

$$\boldsymbol{\mu}_B = \frac{\eta_B^{-1}\boldsymbol{\mu}_1 + \sum_{t=1}^{T} \boldsymbol{x}_t}{\eta_B^{-1} + T}$$

On-Line in trial t

$$\begin{aligned} \boldsymbol{\mu}_{t+1} &= \frac{\eta_1^{-1}\boldsymbol{\mu}_1 + \sum_{q=1}^{t} \boldsymbol{x}_q}{\eta_1^{-1} + t} \\ &= \boldsymbol{\mu}_t - \eta_{t+1}(\boldsymbol{\mu}_t - \boldsymbol{x}_t) \end{aligned}$$

$$\boldsymbol{\theta}_{t+1} = g^{-1}\left(g(\boldsymbol{\theta}_t) - \eta_{t+1}(\boldsymbol{\mu}_t - \boldsymbol{x}_t)\right)$$

- On-line algorithm has freedom to use a tradeoff parameter η_1^{-1} that could be different from the off-line parameter η_B^{-1}

- Two choices for η_1^{-1}

  Case $\eta_1^{-1} = \eta_B^{-1}$:
  Incremental Off-Line Algorithm

  Case $\eta_1^{-1} = \eta_B^{-1} + 1$:
  Forward Algorithm [V]

	Off-line	Forward on-line
Gauss $\boldsymbol{\mu}_1 = 0, \eta_B^{-1} = 0$	$\boldsymbol{\mu}_B = \frac{\sum_{t=1}^{T} \boldsymbol{x}_t}{T}$	$\boldsymbol{\mu}_t = \frac{\sum_{q=1}^{t-1} \boldsymbol{x}_q}{t}$
Bernoulli $\boldsymbol{\mu}_1 = \frac{1}{2}, \eta_B^{-1} = 0$	$\boldsymbol{\mu}_B = \frac{\sum_{t=1}^{T} \boldsymbol{x}_t}{T}$	$\boldsymbol{\mu}_t = \frac{\frac{1}{2}+\sum_{q=1}^{t-1} \boldsymbol{x}_q}{t}$

Key Lemma [AW]

For any example $\boldsymbol{x}_t$ and any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$

$$\underbrace{L_t(\boldsymbol{\theta}_t)}_{\substack{\text{loss of} \\ \text{algorithm}}} - \underbrace{L_t(\boldsymbol{\theta})}_{\substack{\text{loss of} \\ \text{comparator } \boldsymbol{\theta}}}$$

$$= \eta_t^{-1} \underbrace{\Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_t)}_{\substack{\text{divergence} \\ \text{to last par.}}} - \eta_{t+1}^{-1} \underbrace{\Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_{t+1})}_{\substack{\text{divergence} \\ \text{to updated par.}}}$$

$$+ \underbrace{\eta_{t+1}^{-1} \Delta_G(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+1})}_{\substack{\text{cost of} \\ \text{update}}}$$

Main Theorem

For any sequence of examples and any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$

$$\underbrace{\textstyle\sum_{t=1}^{T} L_t(\boldsymbol{\theta}_t)}_{\text{total loss of algorithm}} - \underbrace{L_{1..T}(\boldsymbol{\theta})}_{\text{total loss of comparator } \boldsymbol{\theta}}$$

$$= \underbrace{\eta_1^{-1} \Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_1)}_{\text{divergence to initial par.}} - \underbrace{\eta_{T+1}^{-1} \Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_{T+1})}_{\text{divergence to last par.}}$$

$$+ \underbrace{\textstyle\sum_{t=1}^{T} \eta_{t+1}^{-1} \Delta_G(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+1})}_{\text{cost of all updates}}$$

Proven by simply summing the Key Lemma

Bounds for the Forward Algorithm

$$\sum_{t=1}^{T} L_t(\boldsymbol{\theta}_t) - \inf_{\boldsymbol{\theta}} L_{1..T}(\boldsymbol{\theta})$$

$$\begin{aligned}
&\overset{\text{Gauss}}{=} && \sum_{t=1}^{T} \eta_t \, \boldsymbol{x}_t^2/2 - \sum_{t=1}^{T-1} \eta_t \, \boldsymbol{\mu}_{t+1}^2/2 && [AW] \\
&\leq && \frac{X^2}{2} \ln(1 + \frac{T}{\eta_1^{-1} - 1}) && \\
& && \text{where } X^2 = \max_{t=1}^{T} \boldsymbol{x}_t^2 &&
\end{aligned}$$

$$\begin{aligned}
&\overset{\text{Bernoulli}}{\leq} && \tfrac{1}{2} \ln(T+1) + 1 && [Fr, XB, AW]
\end{aligned}$$

$$\begin{aligned}
&\overset{\text{lin.regr.}}{\leq} && \tfrac{1}{2} Y^2 n \ln\left(1 + \frac{TX^2}{a}\right) && [V, Fo, AW] \\
& && \text{where } Y = \max_{t=1}^{T} y_t && \\
& && \text{and } \boldsymbol{w}_t = \left(a\boldsymbol{I} + \sum_{q=1}^{t} \boldsymbol{x}_q \boldsymbol{x}_q'\right)^{-1} \sum_{q=1}^{t-1} \boldsymbol{x}_q y_q &&
\end{aligned}$$

General Setup

- We hide some information from the learner
- The relative loss bound quantifies the price for hiding the information
- So far the future examples are hidden
  Off-line algorithm knows all examples
  On-line algorithm knows past examples

Minimax Algorithm for T Trials

Gaussian [TW]

Learner against adversary

$$\inf_{\boldsymbol{\theta}_1} \sup_{\boldsymbol{x}_1} \inf_{\boldsymbol{\theta}_2} \sup_{\boldsymbol{x}_2} \inf_{\boldsymbol{\theta}_3} \sup_{\boldsymbol{x}_3} \ldots \inf_{\boldsymbol{\theta}_T} \sup_{\boldsymbol{x}_T}$$

$$\underbrace{\sum_{t=1}^{T} \tfrac{1}{2}(\boldsymbol{\theta}_t - \boldsymbol{x}_t)^2}_{\substack{\text{total loss of} \\ \text{on-line} \\ \text{algorithm}}} - \underbrace{\inf_{\boldsymbol{\theta}} \left(\sum_{t=1}^{T} \tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{x}_t)^2 \right)}_{\substack{\text{total loss of} \\ \text{off-line} \\ \text{algorithm}}}$$

Instances must be bounded: $||\boldsymbol{x}_t||_2 \leq X$

Minimax algorithm usually intractable

Bernoulli is another exception [Sh]

Gaussian

Forward Alg.	$\boldsymbol{\theta}_t = \frac{\sum_{q=1}^{t-1} \boldsymbol{x}_q}{t}$
Bound	$\frac{1}{2}X^2(1 + \ln T)$
Minimax Alg.	$\boldsymbol{\theta}_t = \frac{\sum_{q=1}^{t-1} \boldsymbol{x}_q}{t+\ln T-\ln(t+O(\ln T))}$
Bound	$\frac{1}{2}X^2(\ln T- \ln\ln T) + o(1)$

Minimax alg. needs to know T

Last-step Minimax

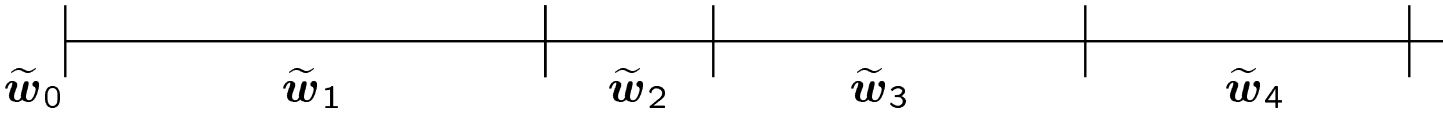
Assumes that current trial is last trial [Fo, TW]

$$\begin{aligned}\boldsymbol{\theta}_t &= \operatorname*{arginf}_{\boldsymbol{\theta}} \sup_{\boldsymbol{x}_t} \sum_{q=1}^{t} L_q(\boldsymbol{\theta}_q) - \inf_{\boldsymbol{\theta}} L_{1..t}(\boldsymbol{\theta}) \\ &= \operatorname*{arginf}_{\boldsymbol{\theta}} \sup_{\boldsymbol{x}_t} L_t(\boldsymbol{\theta}_t) - \inf_{\boldsymbol{\theta}} L_{1..t}(\boldsymbol{\theta})\end{aligned}$$

For Gaussian and linear regression
Last-step Minimax is same as Forward Alg.

For Bernoulli Last-step Minimax slightly better than Forward Alg (Laplace Estimator)

Comparator shifts with time



On-line examples and on-line comparator

$$\underbrace{\sum_{t=1}^{T} L_t(\boldsymbol{w}_t)}_{\substack{\text{total loss of} \\ \text{on-line} \\ \text{algorithm}}} - \inf_{\widetilde{\boldsymbol{w}}_t} \underbrace{\sum_{t=1}^{T} \left(L_t(\widetilde{\boldsymbol{w}}_t) + \Delta(\widetilde{\boldsymbol{w}}_{t-1}, \widetilde{\boldsymbol{w}}_t)\right)}_{\substack{\text{total loss of} \\ \text{shifting off-line} \\ \text{comparator}}}$$

Modifications to the Expert Alg. [HW]

Predict $\hat{y}_t = \boldsymbol{v}_t \cdot \boldsymbol{x}_t$,
where $v_{t,i} = \frac{w_{t,i}}{\sum_{i=1}^{n} w_{t,i}}$

Loss Update $w_{t,i} := w_{t,i} e^{-\eta L_{y_t, x_{t,i}}}$

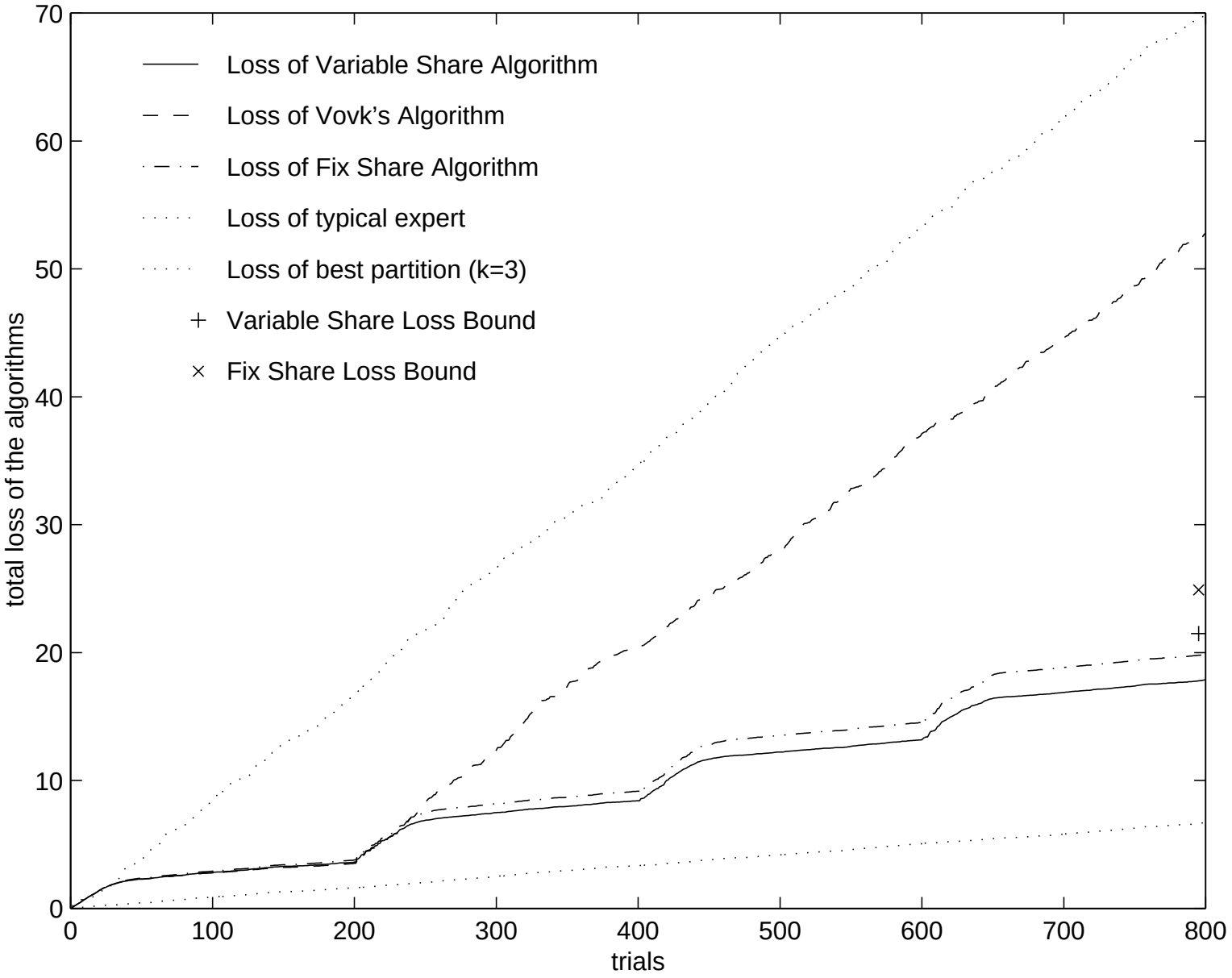
Share Updates ($\alpha \in [0, 1)$)

- Static-expert: Blank

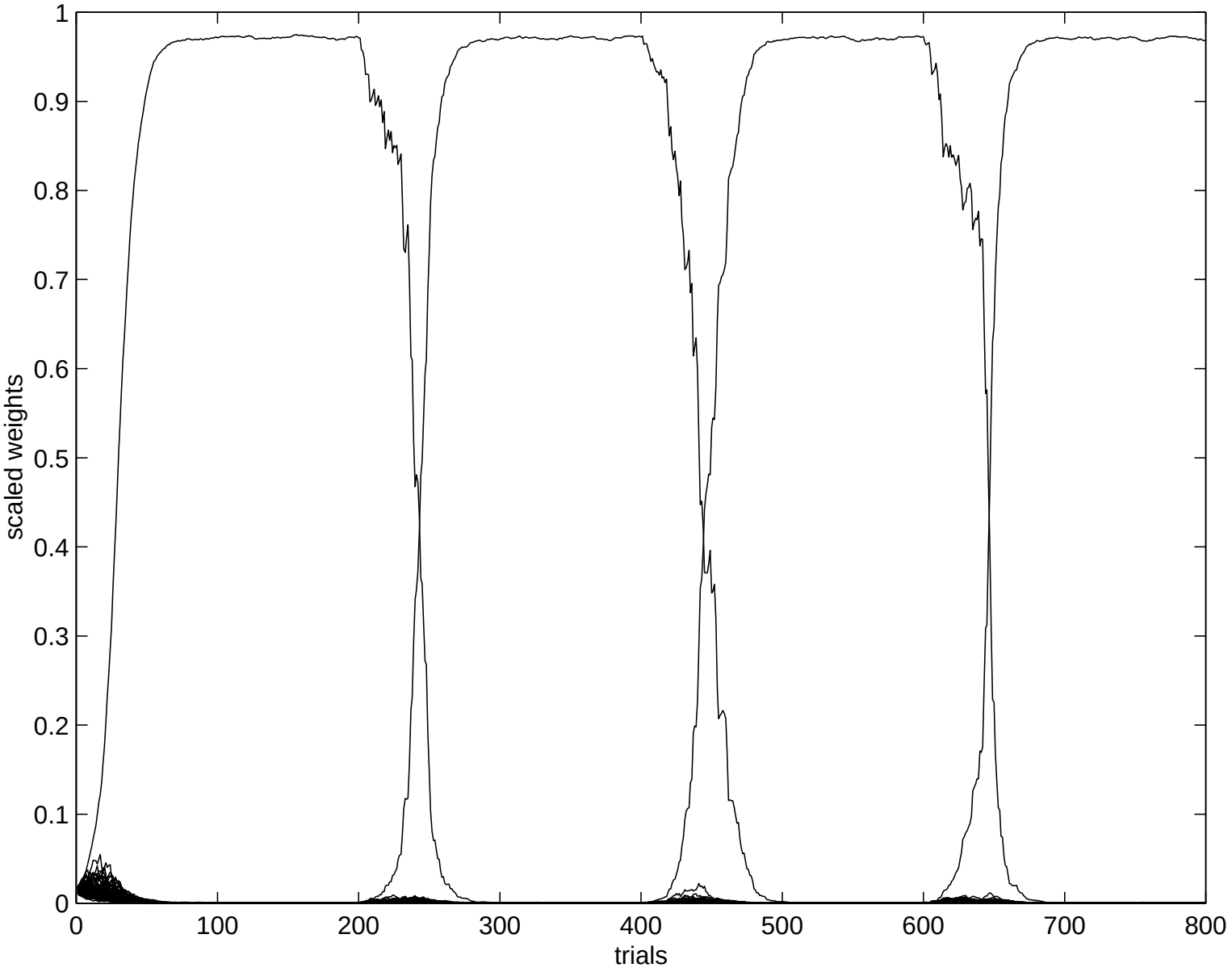
- Fixed-share:
  Each expert sends $\frac{\alpha}{n-1}$ of its weight to the other $n-1$ experts

- Variable-share: Replace $\frac{\alpha}{n-1}$ by

$$\frac{1}{n-1}(1 - (1-\alpha)^{L(y_t, x_{t,i})})$$



Loss of the share algorithms
versus Static Expert Algorithm



Relative weigths of the Fixed Share Algorithm

Shifting bounds

- The Static Expert bounds

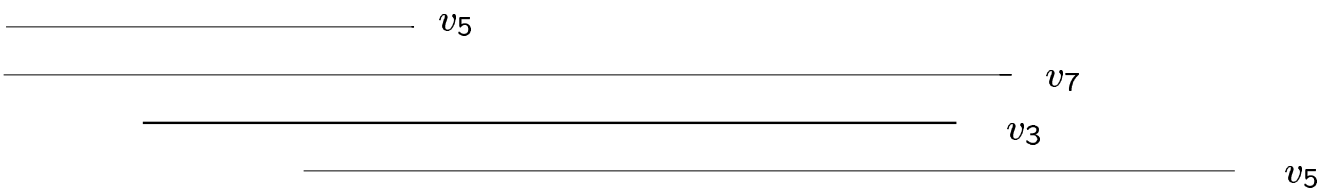
$$L_{\mathsf{Alg}}(S) \leq \min_i L_i(S) + O(\log n))$$

become [HW]

$$L_{\mathsf{Alg}}(S) \leq \min_P L_i(S) + O(\mathsf{size}(\mathsf{P}) \log n)$$

where size(P) is # of shifts in partition P

- For shifting disjunctions [AW]



Schedule τ

$$L_{\mathsf{Alg}}(S) \leq O(\min_{\boldsymbol{\tau}} A_{\boldsymbol{\tau}}(S) + \mathsf{size}(\boldsymbol{\tau}) \log n)$$

where size(τ) is # of literals in τ
and $A_{\boldsymbol{\tau}}(S)$ is # of attrib. errors w.r.t. τ

Applications

- Calendar managing
  Many features (sleeping experts) [BI,FSSW]

- Text categorization [LSCP]
  One attribute per word in text

- Spelling correction [Ro]

- Portfolio prediction [Co,CO,HSSW,BK]

- Boosting [Sc,Fr,SS]

- Load Balancing based on shifting expert algorithms [BB]

Future

- Apply clean setup for density estimation to regression and classification problems

- Other frameworks for deriving on-line algorithms such as Last-Step Minimax Alg.

- Shifting [H]

- More applications for multiplicative updates