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

Claudio Gentile
Università di Milano, Italy

Manfred K. Warmuth
UC Santa Cruz, USA

July 13, 2000

Early contributors:

Nick Littlestone
Volodya Vovk
Tom Cover

On-Line Learning

\[ \begin{array}{cccc|cc|c}
\text{experts} & E_1 & E_2 & E_3 & E_n & \text{prediction} & \text{true label} & \text{loss} \\
\hline
day 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
day 2 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
day 3 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
day t & x_{t,1} & x_{t,2} & x_{t,3} & x_{t,n} & \hat{y}_t & y_t & |y_t - \hat{y}_t| \\
\end{array} \]

Protocol of the Master Algorithm

For \( t = 1 \) To \( T \) Do

Get instance \( x_t \in \{0,1\}^n \)
Predict \( \hat{y}_t \in \{0,1\} \)
Get label \( y_t \in \{0,1\} \)
Incur loss \( |y_t - \hat{y}_t| \)

Halving Algorithm

\[ \begin{array}{ccc}
\text{predict} & 0 & \text{predict} & 1 \\
\hline
\text{all experts} & \text{inconsistent experts} & \text{consistent experts} \\
\end{array} \]

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

A run of the Halving Algorithm

\[ \begin{array}{cccccc|cc|c}
\text{majority} & \text{true label} & \text{loss} \\
\hline
E_1 & E_2 & E_3 & E_4 & E_5 & E_6 & E_7 & E_8 & \text{majority} & \text{true label} & \text{loss} \\
\hline
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
x & x & 0 & 1 & x & x & 1 & 1 & 1 & 1 & 0 \\
x & x & x & 1 & x & x & 0 & 0 & 0 & 1 & 1 \\
x & x & x & ↑ & x & x & x & x & \text{consistent} & \text{consistent} & \text{consistent} \\
\end{array} \]

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**

\[
M_{t,i} = \text{# of mistakes of } E_i \text{ before trial } t
\]

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

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

Minority \( \leq \frac{1}{2} W_t \)

Majority \( \geq \frac{1}{2} W_t \)

If no mistake then

minority multiplied by \( \beta \):

\[
w_{t+1,i} \leq 1 W_t
\]

If mistake then

majority multiplied by \( \beta \):

\[
w_{t+1,i} \leq 1 \frac{1}{2} W_t + \beta \frac{1}{2} W_t
\]

\[
= \frac{1 + \beta}{2} W_t
\]

Hence

\[
W_{T+1} = \sum_{j=1}^{n} w_{T+1,j} = \sum_{j=1}^{n} \beta^{M_j} \geq \beta^{M_t}
\]

We got:

\[
\left( \frac{1 + \beta}{2} \right)^M W_1 \geq \beta^{M_t}
\]

Solving for \( M \):

\[
M \leq \frac{\ln \frac{1}{\beta}}{\ln \frac{2}{1+\beta}} M_t + \frac{1}{\ln \frac{2}{1+\beta}} \ln n
\]

\[
\beta = \frac{1}{e} \min_i M_i + \frac{2.63}{\ln n}
\]

For all sequences, loss of master alg.

is comparable to loss of best expert

Relative loss bounds [Fr]
Other Loss Functions

absolute loss \( L(y, \hat{y}) = |y - \hat{y}| \)
square loss \( L(y, \hat{y}) = (y - \hat{y})^2 \)
entropic loss \( L(y, \hat{y}) = y \ln \frac{y}{\hat{y}} + (1 - y) \ln \frac{1 - y}{1 - \hat{y}} \), \( y, \hat{y} \in [0,1] \)

One weight per expert: \([\checkmark]\)
\[ w_{t,i} = \beta L_{t,i} = e^{-\eta L_{t,i}} \]

where \( L_{t,i} \) is total loss of \( E_i \) before trial \( t \)
and \( \eta \) is a positive learning rate

Master predicts with the weighted average \([KW]\)
\[ v_{t,i} = \frac{w_{t,i}}{\sum_{i=1}^{n} w_{t,i}} \]
normalized weights
\[ \hat{y}_t = \sum_{i=1}^{n} v_{t,i} x_{t,i} = v_t \cdot x_t \]

where \( x_{t,i} \) is the prediction of \( E_i \) in trial \( t \)

Potential: \( \frac{1}{\eta} \ln W_t \)

Key inequality: \( L(y, v_t \cdot x_t) \leq \frac{1}{\eta} \ln W_t - \frac{1}{\eta} \ln W_{t+1} \)
\[ = -\frac{1}{\eta} \ln \frac{W_{t+1}}{W_t} \]

Telescoping:
\[ L_{WA}(S) \leq \frac{1}{\eta} \ln \frac{W_{T+1}}{W_1} \]
\[ = -\frac{1}{\eta} \ln \sum_{i=1}^{n} \frac{1}{\eta} e^{-\eta L_i(S)} \]
\[ \leq -\frac{1}{\eta} \ln \frac{1}{\eta} e^{-\eta L_i(S)} \]
\[ = -\frac{1}{\eta} \ln \frac{1}{\eta} e^{-\eta L_i(S)} \]
\[ = L_i(S) + \frac{1}{\eta} \ln n \]

\[ \forall \ \text{sequences } S \ \text{of examples } \langle (x_l, y_l) \rangle_{1 \leq l \leq T} \text{ where } x_t \in [0,1]^n \text{ and } y_t \in [0,1] \]
\[ L_{WA}(S) \leq \min_i \frac{1}{\eta} L_i(S) + \frac{1}{\eta} \ln \eta n \]

\[
\begin{array}{c|c|c}
1/\eta & \text{dot pred} & \text{fancy} \\
\hline
\text{entropic} & 1 & 1 \\
\text{square} & 2 & 1/2 \\
\text{hellinger} & 1 & .71 \\
\end{array}
\]

- Slightly improved constants of \( 1/\eta \) when Master uses fancier prediction \([\checkmark]\)
- For the discrete loss and the absolute loss \( \alpha > 1 \)

Usefulness:

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

- Bounds \text{logarithmic} in number of experts (\text{multiplicative updates})
Questions:

- How to obtain alg. 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

<table>
<thead>
<tr>
<th>Instance of alg $A$</th>
<th>Prediction of alg $A$</th>
<th>Label</th>
<th>Loss of alg $A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$\hat{y}_1$</td>
<td>$y_1$</td>
<td>$L(y_1, \hat{y}_1)$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$\hat{y}_2$</td>
<td>$y_2$</td>
<td>$L(y_2, \hat{y}_2)$</td>
</tr>
<tr>
<td>$x_T$</td>
<td>$\hat{y}_T$</td>
<td>$y_T$</td>
<td>$L(y_T, \hat{y}_T)$</td>
</tr>
</tbody>
</table>

Total Loss $L_A(S)$

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

Comparison class $\{u\}$

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

**Goal:** Bound relative loss for arbitrary sequence $S$
The Perceptron Algorithm

In trial $t$:
- Get instance $x_i \in \{0,1\}^n$
  - If $w_i \cdot x_i \geq 1/2$ then $\hat{y}_i = 1$
  - else $\hat{y}_i = 0$
- Get label $y_i \in \{0,1\}$
- If mistake then
  $$w_{i+1} = w_i - \eta (\hat{y}_i - y_i)x_i$$

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

# of mistakes $\leq 4A + 4kn$ 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]

# mistakes $= \Omega(n)$

---

The Winnow Algorithm

In trial $t$:
- Get instance $x_i \in \{0,1\}^n$
  - If $w_i \cdot x_i \geq \theta$ then $\hat{y}_i = 1$
  - else $\hat{y}_i = 0$
- Get label $y_i \in \{0,1\}$
- If mistake then
  $$w_{i+1,i} = w_{i,i} e^{-\eta (\hat{y}_i - y_i)x_{l,i}}$$

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

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

Not rotation invariant!

---

$k$-term DNF via Feature Expansion [KW]

$$x = x_1 x_2 \cdots x_n$$

$n$ inputs, $2^n$ features

$$\Phi(x) = x_1 x_2 \cdots x_n$$

$k$-term DNF in input space

is $k$-literal disjunction in feature space

$$\Phi(x) \cdot \Phi(y) = \prod_{i=1}^{n} (1 + x_i y_i) = K(\frac{x}{n}, \frac{y}{n})$$

(Simple ANOVA kernel)

Perceptron:

$$w_i = \sum_q \alpha_q \Phi(x_q)$$

Prediction:

$$w_i \cdot \Phi(x) = \left( \sum_q \alpha_q \Phi(x_q) \right) \cdot \Phi(x)$$

$$= \sum_q \alpha_q \Phi(x_q) \cdot \Phi(x)$$

$$= \sum_{q \text{ mistake}} \alpha_q K(x_q, x)$$

time: $O(n \cdot \# \text{ mistakes})$

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

Winnow:

$$w_{i,i} = \exp \left( -\eta \sum_q \alpha_q \Phi(x_q);i \right)$$

log of weights is linear comb of past examples

Mistake bound: $O(k \ln 2^n) = O(kn)$

prediction time: $\Omega(2^n \# \text{ mistakes})$

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
- Pythagorean 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 $x_t \in \mathbb{R}^n$
- Predict $\hat{y}_t = w_t \cdot x_t$
- Get label $y_t \in \mathbb{R}$
- Incur loss $L_t(w_t) = (y_t - \hat{y}_t)^2$
- Update $w_t$ to $w_{t+1}$

Assume comparison class $\{u\}$ is a set of linear predictors $u : x \mapsto u \cdot x$

Examples of Updates

Gradient descent ($w \in \mathbb{R}^n$)

$$w_{t+1} = w_t - \eta \nabla L_t(w_t)$$

$$= w_t - \eta (w_t \cdot x_t - y_t) x_t$$

[WH]

Exponentiated Gradient Algorithm ($w$ is probability vector)

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

[KW]
More examples of Updates

Unnormalized Exponentiated Gradient Alg. [KW]
\( (w \geq 0) \)

\[
w_{i+1,i} = w_{i,i} \exp \left[ -\frac{\eta \partial L_i(w_i)}{\partial w_{i,i}} \right]
\]

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

\[
w_{i+1,i} = \frac{w_{i,i} \exp \left[ -\frac{\partial L_i(w_i)}{\partial w_{i,i}} \right]}{1 - w_{i,i} \exp \left[ -\frac{\partial L_i(w_i)}{\partial w_{i,i}} \right]}
\]

\[ \text{Motivation of Updates} \quad [KW] \]

Gradient descent

\[
w_{i+1} = \arg\min_w \left\{ \|w - w_i\|_2^2 / 2 + \eta (y_i - w \cdot x_i)^2 / 2 \right\}
\]

\[
= w_i - \eta (w_{i+1} \cdot x_i - y_i) x_i
\]

Exponentiated Gradient Algorithm

\[
w_{i+1} = \arg\min_w \left\{ \sum_{i=1}^n w_i \ln \frac{w_i}{w_{i,i}} + \eta (y_i - w \cdot x_i)^2 / 2 \right\}
\]

\[
= w_{i,i} \exp \left[ -\eta \left( \frac{w_{i+1} \cdot x_i - y_i}{\approx w_i \cdot x_i} \right) x_{i,i} \right] / \text{normalize}
\]

\[ \text{Families of update algorithms} \]

<table>
<thead>
<tr>
<th>parameter divergence</th>
<th>name of family</th>
<th>update algs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( |w - w_i|_2^2 )</td>
<td>Grad. Desc.</td>
<td>Widrow Hoff (LMS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lin. Least Squ.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Backprop.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Perceptron Alg.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>kernel based alg...</td>
</tr>
</tbody>
</table>

\[ \sum_{i=1}^n w_i \ln \frac{w_i}{w_{i,i}} \] Exponentiated Gradient Alg. expert algs

\[ \text{normalized Winnow} \] “AdaBoost”
Families of update algorithms (cont)

\[ \sum_{i=1}^{n} w_i \ln \frac{w_i}{\bar{w}_i} \]  
\[ + w_i - \bar{w}_i \]

\[ \sum_{i=1}^{n} w_i \ln \frac{w_i}{\bar{w}_i} \]  
\[ + (1 - w_i) \ln \frac{1 - w_i}{1 - \bar{w}_i} \]

\[ \Delta_F(\bar{w}, w) = F(\bar{w}) - F(w) - (\bar{w} - w) \cdot \nabla w F(w) \]
\[ \frac{f(w)}{f(\bar{w})} \]
\[ = F(\bar{w}) - \text{supporting hyperplane through } (w, F(w)) \]

\[ F(w) - (\bar{w} - w) \cdot f(w) \]

Members of different families exhibit different behavior

Bregman Divergences

For any differentiable convex function \( F \),

\[ \Delta_F(\bar{w}, w) = F(\bar{w}) - F(w) - (\bar{w} - w) \cdot \nabla w F(w) \]
\[ \frac{f(w)}{f(\bar{w})} \]

\[ = F(\bar{w}) - \text{supporting hyperplane through } (w, F(w)) \]

\[ F(w) - (\bar{w} - w) \cdot f(w) \]

Examples

Bregman Divergences, Simple Properties \([\text{AW}]\)

1. \( \Delta_F(\bar{w}, w) \) is convex in \( \bar{w} \)

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

3. \( \nabla \bar{w} \Delta_F(\bar{w}, w) = f(\bar{w}) - f(w) \)

4. Usually not symmetric: \( \Delta_F(\bar{w}, w) \neq \Delta_F(w, \bar{w}) \)

5. Linearity (for \( a \geq 0 \)):
   \( \Delta_F + a \Delta_H(\bar{w}, w) = \Delta_F(\bar{w}, w) + a \Delta_H(\bar{w}, w) \)

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

7. \( \Delta_F(w_1, w_2) + \Delta_F(w_2, w_3) \)
   \[ = \Delta_F(w_1, w_3) + (w_1 - w_2) \cdot (f(w_3) - f(w_2)) \]

Squared Euclidean Distance

\[ F(w) = \|w\|_2^2 / 2 \]

\[ f(w) = w \]

\[ \Delta_F(\bar{w}, w) = \|\bar{w}\|_2^2 / 2 - \|w\|_2^2 / 2 - (\bar{w} - w) \cdot w \]
\[ = \|\bar{w} - w\|_2^2 / 2 \]

(Unnormalizesed) Relative Entropy

\[ F(w) = \sum_i (w_i \ln w_i - w_i) \]

\[ f(w) = \ln w \]

\[ \Delta_F(\bar{w}, w) = \sum_i (\bar{w}_i \ln \frac{\bar{w}_i}{w_i} + w_i - \bar{w}_i) \]
Examples (cont) [GLS, GL]

p-norm Algs ($q$ is dual to $p$)

\[ F(w) = \frac{1}{2}||w||^2_q \]
\[ f(w) = \nabla \frac{1}{2}||w||^2_q \]
\[ \Delta_F(\bar{w}, w) = \frac{1}{2}||\bar{w}||^2_q + \frac{1}{2}||w||^2_q - \bar{w} \cdot f(w) \]

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

General Motivation of Updates [KW]

Trade-off between two divergences:
\[ w_{t+1} = \arg \min_w (\Delta_F(w, w_t) + \eta_t L_t(w)) \]
\[ \Delta_F(w, w_t) \text{ is regularization term and serves as measure of progress in the analysis.} \]

When loss $L$ is convex (in $w$)
\[ \nabla w(\Delta_F(w, w_t) + \eta_t L_t(w)) = 0 \]
iff
\[ f(w) - f(w_t) + \eta_t \nabla L_t(w) = 0 \]
\[ \Rightarrow \]
\[ w_{t+1} = f^{-1}(f(w_t) - \eta_t \nabla L_t(w_t)) \]
Characterization of algs. i.t.o. link function \( f = \nabla F \) \[ [WJ] \]

\[
 w_{t+1} = f^{-1}(f(w_t) - \eta_t \nabla L_t(w_t))
\]

<table>
<thead>
<tr>
<th>Alg.</th>
<th>( f(w) )</th>
<th>Domain of ( w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD</td>
<td>( f(w) = w )</td>
<td>( w \in \mathbb{R}^n )</td>
</tr>
<tr>
<td>BEG</td>
<td>( f(w) = \ln \frac{w}{1-w} )</td>
<td>( w \in [0,1]^n )</td>
</tr>
<tr>
<td>EG</td>
<td>( f(w) = \ln \frac{w}{1-</td>
<td></td>
</tr>
<tr>
<td>( p )-norm</td>
<td>( f(w) = \nabla \frac{1}{2}</td>
<td></td>
</tr>
</tbody>
</table>
Bregman Divergences Lead to Good Loss Functions

\[ \tilde{y} = h(w \cdot x) \]

\[ h(z) = \frac{1}{1 + e^{-z}} \]

- Sigmoid function

- For a set of examples \((x_1, y_1), \ldots, (x_T, y_T)\)

\[ \text{total loss } \sum_{t=1}^{T} (h(w \cdot x_t) - y_t)^2 / 2 \]

can have exponentially many minima in weight space \([Bu, AHW]\)

Want loss that is convex in \(w\)

Bregman Divergences Lead to Good Loss Functions (cont)

\[ H(z) = \int_{h^{-1}(y)}^{w \cdot x} (h(z) - y) dz \]

\[ = H(w \cdot x) - H(h^{-1}(y)) - (w \cdot x - h^{-1}(y)) y \]

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

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

Called matching loss for \(h\) \([AHW, HKW]\)

Matching loss is convex in \(w\)

\[
\begin{array}{c|c|c}
\text{transfer f. } h(z) & H(z) & \text{match. loss } d_H(w \cdot x, h^{-1}(y)) \\
\hline
z & \frac{1}{2}z^2 & \frac{1}{2}(w \cdot x - y)^2 \\
\frac{e^z}{1+e^z} & \ln(1 + e^z) & \ln(1 + e^{w \cdot x}) - yw \cdot x + y \ln y + (1 - y) \ln(1 - y) \\
\text{sign}(z) & |z| & \max\{0, -yw \cdot x\} \\
\end{array}
\]

square loss

logistic loss

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

![Graph showing sign function](image)

$H(z) = |z|

Matching loss is \textit{hinge loss} \cite{GW}

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

Convex in $w$ but not differentiable

**Motivation of linear threshold alg\s**

- Gradient descent
  - Perceptron
  - Hinge Loss
- Exponential gradient
  - Normalized
  - Winnow
  - Hinge Loss

Known linear threshold alg\s for ±1-class are gradient-based alg\s with hinge loss

**Trade-off between two divergences** \cite{KW}

$$w_{t+1} = \arg\min_{w} \left( \Delta_F(w, w_t) + \eta_t \Delta_H(w \cdot x_t, h^{-1}(y_t)) \right)$$

parameter divergence matching loss

Both divergences are convex in $w$

$$w_{t+1} = f^{-1}(f(w_t) - \eta_t(h(w_t \cdot x_t) - y_t)x_t)$$

Generalization of the “delta”-rule

**Projections**

$$w_{t+1} = \arg\min_{w} (\Delta_F(w, w_t) + \eta(w \cdot x_t - y_t)^2)$$

When $\eta$ is large then $w_{t+1}$ is projection of $w_t$ onto plane $w \cdot x_t = y_t$

$$w_{t+1} = \arg\min_{w} \{w : w_t x_t = y_t\} \Delta_F(w, w_t)$$

The \textit{AdaBoost} update of the probability vector $w_t$ on the examples is a projection w.r.t. divergence $\Delta_F(w, w_t) = \sum_i w_i \ln \frac{w_i}{w_{t+i}}$ \cite{CKW, La, KW, CSS}
A Pythagorean Theorem [Br,Cs,A.HW]

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

$$w^* = \arg\min_{u \in \mathcal{W}} \Delta_F(u, w)$$

**Th:**

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

---

**How do we prove relative loss bounds?**

**Loss:**

$$L_t(w) = L((x_t, y_t), w) \text{ convex in } w$$

**Divergence:**

$$\Delta_F(u, w) = F(u) - F(w) - (u - w) \cdot f(w)$$

**Update:**

$$f(w_{t+1}) - f(w_t) = -\eta \nabla_w L_t(w_t)$$

**$L_t(u)$ convexity**

$$\geq L_t(w_t) + (u - w_t) \cdot \nabla_w L_t(w_t)$$

**update**

$$= L_t(w_t) - \frac{1}{\eta} (u - w_t) \cdot (f(w_{t+1}) - f(w_t))$$

**prop. 7 of $\Delta_F$**

$$= L_t(w_t) + \frac{1}{\eta} (\Delta_F(u, w_{t+1}) - \Delta_F(u, w_t) - \Delta_F(w_t, w_{t+1}))$$

---

**Summing over $t$** [W.J.KW]

$$\sum_t L_t(w_t) \leq \sum_t L_t(u) + \frac{1}{\eta} \sum_t (\Delta_F(u, w_t) - \Delta_F(u, w_{t+1}))$$

$$+ \Delta_F(w_t, w_{t+1})$$

$$\leq \sum_t L_t(u)$$

$$+ \frac{1}{\eta} (\Delta_F(u, w_1) - \Delta_F(u, w_{T+1})) + \frac{1}{\eta} \sum_t \Delta_F(w_t, w_{t+1})$$

$$\sum_t L_t(w_t) \leq \sum_t L_t(u) + \frac{1}{\eta} \Delta_F(u, w_1)$$

$$+ \frac{1}{\eta} \sum \Delta_F(w_t, w_{t+1})$$

Any convex loss and any Bregman divergence!

---

**Key step:**

Relate $\Delta_F(w_t, w_{t+1})$ to loss $L_t(w_t)$

Loss & divergence dependent

Get $\Delta_F(w_t, w_{t+1}) \leq \text{const} \ L_t(w_t)$

Then solve for $\sum_t L_t(w_t)$

Yield bounds of the form

$$\sum_t L_t(w_t) \leq a \sum_t L_t(u) + b \Delta_F(u, w_1)$$

$a, b$ constants, $a > 1$.

Regret bounds ($a = 1$):

time changing $\eta$, subtler analysis [AG]
Some Bounds \([KW,GLS,GL]\)
(Linear Regression with Square Loss)

Gradient Descent
\[
\sum_{t} L_t(w_t) \leq (1 + c) \sum_{t} L_t(w) + \frac{1 + c}{c} X_2^2 U_2^2 \\
||x_t||_2 \leq X_2, ||u||_2 \leq U_2, c > 0
\]

Scaled Exponentiated Gradient
\[
\sum_{t} L_t(w_t) \leq (1 + c) \sum_{t} L_t(w) + \frac{1 + c}{c} \ln n X_\infty^2 U_1^2 \\
||x_t||_\infty \leq X_\infty, ||u||_1 \leq U_1, c > 0
\]

\(p\)-norm alg
\[
\sum_{t} L_t(w_t) \leq (1 + c) \sum_{t} L_t(w) + \frac{1 + c}{c} (p - 1) X_p^2 U_1^2 \\
||x_t||_p \leq X_p, ||u||_q \leq U_q, c > 0
\]

Generalization Bounds
Loss bounds for on-line algns
\(\Rightarrow\)
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 \(y_t\)

\[S = (x_1, y_1), \ldots, (x_T, y_T) \sim D^T\]

Alg \(A\) updates only if mistake

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

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

\[err_D(A'(S)) = Pr_{(x,y) \sim D} (A'(S)(x) \neq y)\]

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

(Simplest) Leave One Out \([HW]\)

Loss \(L: \mathbb{R}^2 \rightarrow \mathbb{R}\)

\[S = (x_1, y_1), \ldots, (x_T, y_T) \sim D^T\]

Given alg \(A\), want to bound

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

Run \(A\) \(1\) \(2\) \(\ldots\) \(T+1\)

Pick one at random

call it \(h_i\) and predict on new instance \(x\) as \(h_i(x)\)
(Simplest) Leave One Out (cont)

\[ L_{\text{loo}}(S)(x) = h_1(x) \sim \frac{1}{T+1} \]

Then:

\[ E[L(y, L_{\text{loo}}(S)(x))] \]
\[ \leq \frac{E_{S \sim Y^T}[\text{cum. loss of } A]}{T+1} \]
\[ \leq \text{worst-case loss bound} \]

Applied to the Perceptron Alg. \[ [FS] \]

Where do Bregman divergences come from?

- Exponential family of distributions
- Inherent duality

\[ w_{i+1} = f^{-1}(f(w_i) - \eta \nabla L_i(w_i)) \]

primal param. dual param.

\[ w_i \quad \overset{f}{\longrightarrow} \quad f(w_i) \]
\[ w_{i+1} \quad \overset{f^{-1}}{\longleftarrow} \quad -\eta \nabla L_i(w_i) \]

Exponential Family of Distributions

- Parametric density functions
  \[ P_G(x|\theta) = e^{\theta \cdot x - G(\theta)} P_0(x) \]
  - \( \theta \) and \( x \) vectors in \( \mathbb{R}^d \)

- Cumulant function \( G(\theta) \) assures normalization
  \[ G(\theta) = \ln \int e^{\theta \cdot x} P_0(x) dx \]
  - \( G(\theta) \) is convex function on convex set \( \Theta \subseteq \mathbb{R}^d \)
  - \( G \) characterizes members of the family
  - \( \theta \) is natural parameter

- Expectation parameter
  \[ \mu = \int x P_G(x|\theta) dx = E_{\theta}(x) = g(\theta) \]
  where \( g(\theta) = \nabla_{\theta} G(\theta) \)

- Second convex function \( F(\mu) \) on space \( g(\Theta) \)
  \[ F(\mu) = \theta \cdot \mu - G(\theta) \]

- \( G(\theta) \) and \( F(\mu) \) are dual convex functions
  - Let \( f(\mu) = \nabla_{\mu} F(\mu) \)
  - \( f(\mu) = g^{-1}(\mu) \)
Summary

natural par.           expectation par.

\[ \theta \quad \rightarrow \quad g, \quad \mu \quad \leftarrow \quad f \]

\[ G(\theta) \quad \rightarrow \quad F(\mu) \]

- \( \theta \) and \( \mu \) are dual parameters

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

\[ \text{Gaussian (unit variance)} \]

\[ P(x|\theta) \sim e^{-\frac{1}{2}(\theta - x)^2} \]

\[ = e^{\theta - \frac{1}{2} \theta^2} e^{-\frac{1}{2} x^2} \]

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

Parameter transformations:

\[ g(\theta) = \theta = \mu \quad \text{and} \quad f(\mu) = \mu = \theta \]

Dual convex function: \[ F(\mu) = \frac{\theta \cdot \mu - G(\theta)}{\frac{1}{2} \mu^2} \]

Square loss: \[ L_4(\theta) = \frac{1}{2}(\theta - x)^2 \]

\[ \text{Bernoulli} \]

Examples \( x_i \) are coin flips in \{0,1\}

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

\( \mu \) 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:

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

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

Log loss: \[ L_4(\theta) = -x_i \theta + \ln(1 + e^\theta) \]

\[ = -x_i \ln \mu - (1 - x_i) \ln(1 - \mu) \]

\[ \text{Poisson} \]

Examples \( x_i \) are natural numbers in \{0,1,\ldots\}

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

\( \mu \) 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:

\[ \mu = g(\theta) = e^\theta \quad \text{and} \quad \theta = f(\mu) = \ln \mu \]

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

Loss: \[ L_4(\theta) = -x_i \theta + e^\theta + \ln x_i! \]

\[ = -x_i \ln \mu + \mu + \ln x_i! \]
Bregman Divergences as Relative Entropies between Exponential Distributions

Let \( P(x|\theta) \) and \( P(x|\bar{\theta}) \) denote two distributions with cumulant function \( G \):

\[
\Delta_G(\bar{\theta}, \theta) = \int_x P_G(x|\theta) \ln \frac{P_G(x|\theta)}{P_G(x|\bar{\theta})} \, dx
\]

\[
= G(\bar{\theta}) - G(\theta) - (\bar{\theta} - \theta) \cdot \mu
\]

\[
F(\mu) = \theta \mu - G(\theta)
\]

\[
F(\mu) - F(\mu) - (\mu - \mu) \cdot \bar{\theta}
\]

\[
= \Delta_F(\mu, \bar{\mu})
\]

[A, BN, AW]

---

Area unchanged When Slide Flipped

Dual divergence for Bernoulli

\[
G(\theta) = \ln(1 + e^\theta) \quad F(\mu) = \mu \ln \mu + (1 - \mu) \ln(1 - \mu)
\]

\[
g(\theta) = \frac{e^\theta}{1 + e^\theta} = \mu \quad f(\mu) = \ln \frac{\mu}{1 - \mu} = \theta
\]

\[
\Delta_G(\bar{\theta}, \theta) = \ln(1 + e^\bar{\theta}) - \ln(1 + e^\theta) - (\bar{\theta} - \theta) \frac{e^\theta}{1 + e^\theta}
\]

\[
\Delta_F(\mu, \bar{\mu}) = \mu \ln \frac{\mu}{\bar{\mu}} + (1 - \mu) \ln \frac{1 - \mu}{1 - \bar{\mu}}
\]

Binary relative entropy
Dual divergence for Poisson

\[ g(\theta) = e^\theta \quad f(\mu) = \mu \ln \mu - \mu \]

\[ h(\theta) = e^\theta \quad f(\mu) = \ln \mu = \theta \]

\[ \Delta_G(\theta, \theta) = e^\theta - e^\theta - (\theta - \theta)e^\theta \]

\[ \Delta_F(\mu, \bar{\mu}) = \mu \ln \frac{\mu}{\bar{\mu}} + \bar{\mu} - \mu \]

Unnormalized relative entropy

Dual matching loss for sigmoid transfer func.

\[ H(z) = \ln (1 + e^z) \quad K(r) = r \ln r + (1 - r) \ln (1 - r) \]

\[ h(z) = \frac{e^z}{1 + e^z} = r \quad k(r) = \ln \frac{r}{1 - r} = z \]

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

\[ \Delta_H(w \cdot x, h^{-1}(y)) = \ln (1 + e^{w \cdot x}) - yw \cdot x + y \ln y + (1 - y) \ln (1 - y) \]

By duality \textit{logistic loss} is same as \textit{entropic loss}

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

Derivation of Updates

- Want to bound

\[ \sum_{t=1}^{T} L_t(\theta_t) - \inf_{\theta} L_{1..T}(\theta) \]

- Off-line algorithm has all \( T \) examples

\[ \{x_1, x_2, \ldots, x_T\} \]

- Setup for choosing best parameter setting

\[ \theta_B = \arg\min_{\theta} \eta_B^{-1} \Delta_G(\theta, \theta_1) + L_{1..T}(\theta) \]

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

Gaussian density estimation

(Fixed variance)

\[ \frac{1}{2}(x_i - \theta)^2 \]

Off-line versus on-line

- Loss on example \( x_i \)

\[ L_i(\theta) = -\ln P(x_i|\theta) = \frac{1}{2}(x_i - \theta)^2 \]
• Equivalent to Bayesian MAP

where \( \eta_1^{-1} \Delta_G(\theta, \theta_1) \) is the log of the conjugate prior

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

• Alternate:

\( \eta_B^{-1} \Delta_G(\theta, \theta_1) \) loss on initial set of examples

73

On-line Algorithm

• In trial \( t \), the first \( t \) examples \( \{x_1, x_2, \ldots, x_t\} \) 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

\[
\theta_{t+1} = \arg \min_{\theta} \left( \eta_1^{-1} \Delta_G(\theta, \theta_1) + L_{1:t}(\theta) \right)
\]

Tradeoff parameter \( \eta_1^{-1} \geq 0 \)

74

Alternate Motivation of Same On-Line Update

\[
\theta_{t+1} = \arg \min_{\theta} \left( \eta_t^{-1} \Delta_G(\theta, \theta_t) + L_t(\theta) \right)
\]

where

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

Parameter Updates

Off-line

\[
\mu_B = \frac{\eta_B^{-1} \mu_1 + \sum_{i=1}^T x_i}{\eta_B^{-1} + T}
\]

On-Line in trial \( t \)

\[
\mu_{t+1} = \frac{\eta_1^{-1} \mu_1 + \sum_{q=1}^t x_q}{\eta_1^{-1} + t} = \mu_t - \eta_t^{-1} (\mu_t - x_t)
\]

\[
\theta_{t+1} = \eta_1^{-1} \left( g(\theta_t) - \eta_{t+1}(\mu_t - x_t) \right)
\]

75

• On-line algorithm has freedom to use a tradeoff parameter \( \eta_1^{-1} \) that could be different from the off-line parameter \( \eta_B^{-1} \)

• Two choices for \( \eta_1^{-1} \)

Case \( \eta_1^{-1} = \eta_B^{-1} \):

Incremental Off-Line Algorithm

Case \( \eta_1^{-1} = \eta_B^{-1} + 1 \):

Forward Algorithm

[V]

76
Main Theorem

For any sequence of examples and any $\theta \in \Theta$

\[
\sum_{i=1}^{T} I_i(\theta_i) - \inf_{\theta} I_{1..T}(\theta) = \eta_1^{-1} \Delta_G(\theta, \theta_1) - \eta_{T+1}^{-1} \Delta_G(\theta, \theta_{T+1})
\]

total loss of algorithm
total loss of comparator $\theta$
divergence to initial par.
divergence to last par.

+ $\sum_{i=1}^{T} \eta_i^{-1} \Delta_G(\theta_i, \theta_{i+1})$

cost of all updates

Proven by simply summing the Key Lemma

Key Lemma

For any example $x_i$ and any $\theta \in \Theta$

\[
L_i(\theta) - L_i(\theta) = \eta_i^{-1} \Delta_G(\theta, \theta_i) - \eta_{i+1}^{-1} \Delta_G(\theta, \theta_{i+1})
\]

divergence to last par.
divergence to updated par.

+ $\eta_{i+1}^{-1} \Delta_G(\theta_i, \theta_{i+1})$

cost of update

Bounds for the Forward Algorithm

\[
\sum_{i=1}^{T} L_i(\theta_i) - \inf_{\theta} L_{1..T}(\theta) \leq \frac{1}{2} \ln(T + 1) + 1
\]

[Fr, XB, AW]

\[
\sum_{i=1}^{T} \eta_i \frac{x_i^2}{2} - \sum_{i=1}^{T-1} \eta_i \frac{\mu_{i+1}^2}{2}
\]

[AW]

where $X^2 = \max_{i=1}^{T} x_i^2$

lin.regr.

\[
\frac{1}{2} Y^{-2} n \ln \left(1 + \frac{T X^2}{a}\right)
\]

[V, Fo, AW]

where $Y = \max_{i=1}^{T} y_i$

and $w_i = \left( aI + \sum_{q=1}^{i} x_q x_q^T \right)^{-1} \sum_{q=1}^{i-1} x_q y_q$
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

Learner against adversary

$$\inf_{\theta_1} \sup_{x_1} \inf_{\theta_2} \sup_{x_2} \ldots \inf_{\theta_T} \sup_{x_T}$$

$$\sum_{t=1}^{T} \frac{1}{2}(\theta_t - x_t)^2 = \inf_\theta \left( \sum_{t=1}^{T} \frac{1}{2}(\theta - x_t)^2 \right)$$

Total loss of on-line algorithm

Total loss of off-line algorithm

Instances must be bounded: $\|x_t\|_2 \leq X$

Minimax algorithm usually intractable

Bernoulli is another exception

Gaussian

Forward Alg.

$$\theta_t = \frac{\sum_{q=1}^{t-1} x_q}{t}$$

Bound

$$\frac{1}{2} x^2 (1 + \ln T)$$

Minimax Alg.

$$\theta_t = \frac{\sum_{q=1}^{t-1} x_q}{t + \ln T - \ln(1 + O(\ln T))}$$

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]

$$\theta_t = \arg \inf_\theta \sup_{x_t} \sum_{q=1}^{t} L_q(\theta_t) - \underset{\theta}{\inf} L_{1..t}(\theta)$$

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

\[ \sum_{t=1}^{T} L_t(w_t) - \inf_{w_t} \sum_{t=1}^{T} (L_t(w_t) + \Delta(w_{t-1}, w_t)) \]

\[ \text{total loss of on-line algorithm} - \text{total loss of shifting off-line comparator} \]

---

**Modifications to the Expert Alg.** [HW]

Predict \( \hat{y}_t = v_t \cdot x_t \), where

\[ \hat{y}_t = \frac{v_{t,i}}{\sum_{i=1}^{n} w_{t,i}} \]

Loss Update \( w_{t,i} := w_{t,i} e^{-\eta L_{W_i, x_t,i}} \)

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
  \[ \frac{1}{n-1} (1 - (1 - \alpha)^L(y_t x_t)) \]

---

**Loss of the share algorithms versus Static Expert Algorithm**

**Relative weights of the Fixed Share Algorithm**
**Shifting bounds**

- The Static Expert bounds
  \[ L_{\text{Alg}}(s) \leq \min_i L_i(s) + O(\log n) \]
  become \([\text{HW}]\)

  \[ L_{\text{Alg}}(s) \leq \min_P L_i(s) + O(\text{size}(P) \log n) \]
  where \text{size}(P)\ is \#\ of\ shifts\ in\ partition\ P

- For shifting disjunctions \([\text{AW}]\)

\[
\begin{array}{c}
\text{Schedule } \tau \\
\hline
\text{size}(\tau) \\
\text{size}(\tau) \\
\text{size}(\tau) \\
\text{size}(\tau) \\
\text{size}(\tau) \\
\hline
\end{array}
\]

\[ L_{\text{Alg}}(s) \leq O(\min_\tau A_\tau(s) + \text{size}(\tau) \log n) \]

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

---

**Applications**

- Calendar managing
  Many features (sleeping experts) \([\text{BL,FSSW}]\)

- Text categorization \([\text{LSCP}]\)
  One attribute per word in text

- Spelling correction \([\text{Ro}]\)

- Portfolio prediction \([\text{Co,CO,HSSW,BK}]\)

- Boosting \([\text{Sc,Fr,SS}]\)

- Load Balancing based on shifting expert algorithms \([\text{BB}]\)

---

**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 \([\text{H}]\)

- More applications for multiplicative updates