Online Learning and Bregman Divergences

Part III

Gunnar Rätsch[&] and Manfred Warmuth[$]

[&] *Australian National University, Canberra, Australia*

[$] *University of California at Santa Cruz, USA*

[&] `http://mlg.anu.edu.au/~raetsch`

[$] `http://www.cse.ucsc.edu/~manfred`

Help with the tutorial: Claudio Gentile, Jyrki Kivinen

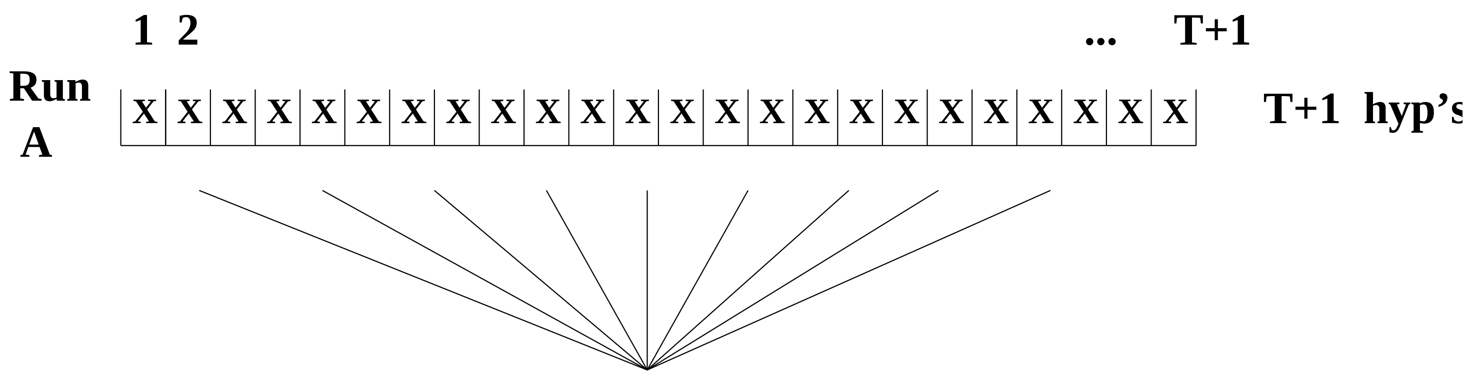
Content of this tutorial

- P I: Introduction to Online Learning
  - The Learning setting
  - Predicting as good as the best expert
  - Predicting as good as the best linear combination of experts
- P II: Bregman divergences and Loss bounds
  - Introduction to Bregman divergences
  - Relative loss bounds for the linear case
  - Nonlinear case & matching losses
  - Duality and relation to exponential families
- P III: Extensions, interpretations, applications
  - Asymptotic Results and Natural Gradients
  - Prior information on the weight vector
  - Some applications

Goal: How can we prove relative loss bounds?

Averaging: A ϵ-δ-Bound [CCG]

Convex Loss $L : \mathbf{R}^2 \longrightarrow [0, L_{\max}]$



$$\overline{h}(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} h_t(\boldsymbol{x})$$

Assume total loss is M

Then holds with probability $1 - \delta$:

$$err_D(\overline{h}) \leq \frac{M}{T} + L_{\max}\sqrt{\frac{2}{T}\log\frac{1}{\delta}}$$

A Refinement

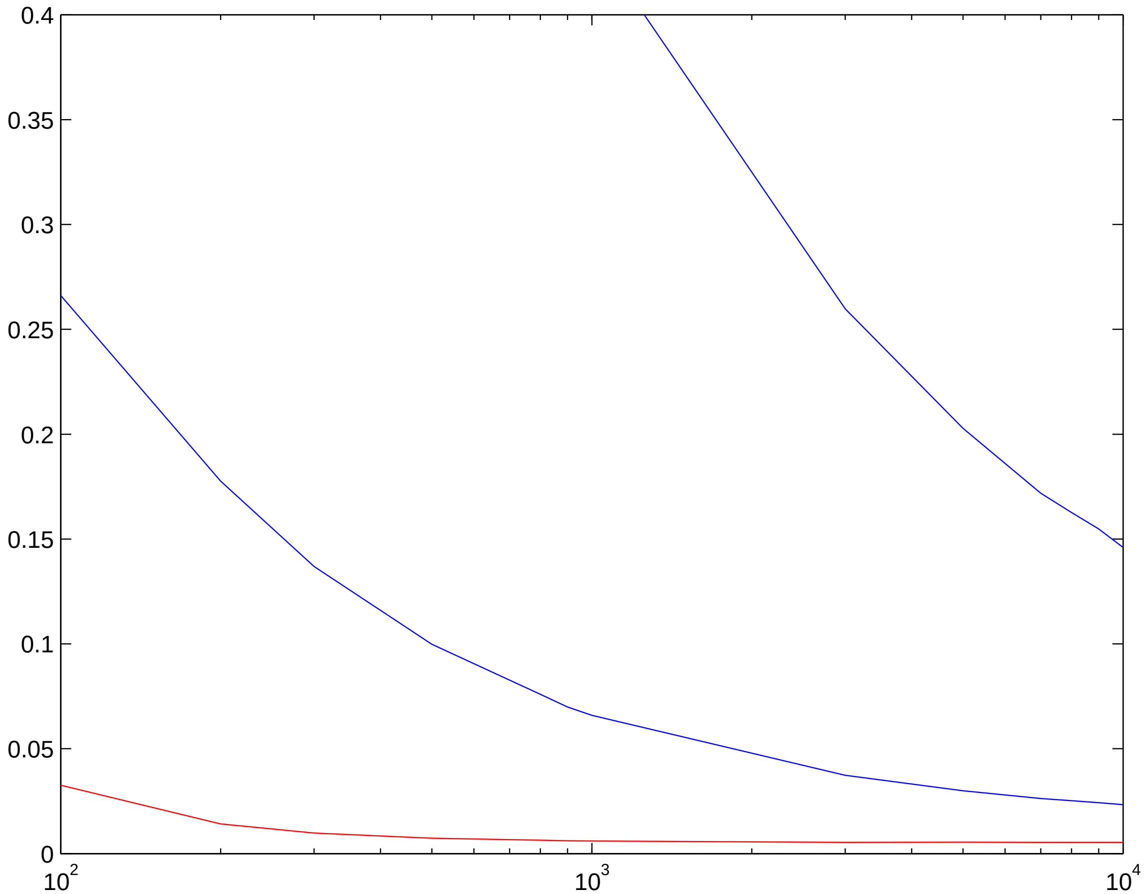
... for Gradient Descent with square loss

$$err_D(\overline{h}) \le \underbrace{\frac{M(\mathbf{u}, S)}{T}}_{\le err_S(\overline{h})} + \frac{\eta^{-1}\|\mathbf{u}\| + 2Y^2 \sum_{i=1}^{n} \log(1 + \lambda_i \eta)}{T} + 2Y\sqrt{\frac{2}{T}\log\frac{1}{\delta}},$$

where $Y = \max_{y,z} L(y, z)$ and λ_i are the eigenvalues of the Gram matrix

$\Rightarrow$ Applies e.g. to Kernel Regression

Illustration of the Bound



Squared loss, random target $\mathbf{u}$ and random $\boldsymbol{x}_t$'s ($\mathbf{u}, \boldsymbol{x}_t \in \mathbf{R}^2$)

$y_t = \mathbf{u} \cdot \boldsymbol{x}_t + n_i$, where $n_i \sim \mathcal{N}(0, 0.3)$, $\eta = 0.1$

On Consistency

What happens in the limit?

Is the estimate converging to the minimum?

$$\frac{L_A(S)}{T} \quad \overset{t\to\infty}{\longrightarrow} \quad \frac{\inf_{\mathbf{u}} L_{\mathbf{u}}(S)}{T}$$

Proved Loss bounds:

$$L_A(S) \quad \leq \quad a \inf_{\mathbf{u}} L_{\mathbf{u}} + b$$

if $a = 1$, then

$$\frac{L_A(S)}{T} \leq \frac{\inf_{\mathbf{u}} L_{\mathbf{u}}(S)}{T} + \frac{b}{T}$$

But when η fixed, then $a > 1$

$\Rightarrow$ not necessarily consistent

An asymptotic result [RM]

For

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w} \cdot x, y)$$

If $\eta_t \to 0$ such that

$$\sum_t \eta_t \quad \xrightarrow{t \to \infty} \quad \infty$$

$$\sum_t \eta_t^2 \quad \stackrel{t \to \infty}{\not\longrightarrow} \quad \infty$$

Then

$$\frac{L_A(S)}{T} \quad = \quad \frac{\inf_{\mathbf{u}} L_{\mathbf{u}}(S)}{T} + \mathcal{O}\left(\frac{1}{t}\right)$$

Best non-asymptotic result (so far) [ACG]

For GD, EG and p-norm algorithms with $\eta_t \sim \frac{1}{\sqrt{t}}$

$$L_A(S) \leq \inf_{\mathbf{u}} L_{\mathbf{u}}(S) + b + c\sqrt{\inf_{\mathbf{u}} L_{\mathbf{u}}(S)}$$

Hence

$$\frac{L_A(S_T)}{T} = \frac{\inf_{\mathbf{u}} L_{\mathbf{u}}(S_T)}{T} + \mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$$

Can this be improved?

Remark on the Geometry of Optimization

- Consider the case where one minimizes

$$\mathbf{E}_{\boldsymbol{x},y} L(\boldsymbol{w} \cdot \boldsymbol{x}, y)$$

- Error surface often looks like a taco shell
- Transformation of gradient helps to improve convergence speed:

$$H^{-1} \frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}_t \cdot \boldsymbol{x}_t, y_t)$$

- same as using another divergence:

$$\boldsymbol{\Delta}(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = (\boldsymbol{w} - \tilde{\boldsymbol{w}})^\top H (\boldsymbol{w} - \tilde{\boldsymbol{w}})$$

leading to

$$H \boldsymbol{w}_{t+1} = H \boldsymbol{w}_t - \eta \frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}_t \cdot \boldsymbol{x}_t, y_t)$$

A Special Case

Density Estimation in Exponential Families

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \exp(\boldsymbol{\theta} \cdot \boldsymbol{x} - G(\theta))p_0(\boldsymbol{x})$$

Minimize $\sum_t -\log p(\boldsymbol{x}|\boldsymbol{\theta})$

Measuring the distance between two parameters

$$\Delta_G(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = \int_{\boldsymbol{x}} p(\boldsymbol{x}|\theta) \log \left(\frac{p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)} \right) d\boldsymbol{x}$$

Update

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &\approx \min_{\boldsymbol{\theta}} \int_{\boldsymbol{x}} p(\boldsymbol{x}|\theta) \log \left(\frac{p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta_t)} \right) d\boldsymbol{x} + \eta \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\boldsymbol{x}_t|\boldsymbol{\theta}_t) \\ &= \min_{\boldsymbol{\theta}} \Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_t) + \eta(\boldsymbol{x}_t - g(\boldsymbol{\theta}_t)) \end{aligned}$$

A Special Case (cont)

$$\boldsymbol{\theta} \approx \min_{\boldsymbol{\theta}} \Delta_G(\boldsymbol{\theta}, \boldsymbol{\theta}_t) + \eta(\boldsymbol{x}_t - g(\boldsymbol{\theta}_t))$$

Update:

$$\underbrace{g(\boldsymbol{\theta}_{t+1})}_{\boldsymbol{\mu}_{t+1}} = \underbrace{g(\boldsymbol{\theta}_t)}_{\boldsymbol{\mu}_t}(1-\eta) + \eta \boldsymbol{x}_t$$

$\Rightarrow$ “Leaky” average of $\boldsymbol{x}_t$’s

$\Rightarrow$ update in the dual (=expectation) parameter

How does it work for other distributions?

$$\int_{\boldsymbol{x}} p(\boldsymbol{x}|\theta) \log \left(\frac{p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)} \right) d\boldsymbol{x} \neq \Delta_G(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}})$$

because G would not be convex

$$\begin{aligned} \int p(\boldsymbol{x}|\theta) \log \left(\frac{p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)} \right) d\boldsymbol{x} &= (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top I_\theta (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) + \mathcal{O}(\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|^2) \\ &\approx (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top I_{\theta^*} (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) \\ &= \Delta_{\tilde{G}}(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \end{aligned}$$

where

$$I_\theta = \mathbf{E}_{\boldsymbol{x}}[\nabla \log p(\boldsymbol{x}|\boldsymbol{\theta})^\top \nabla \log p(\boldsymbol{x}|\boldsymbol{\theta})]$$

is the Fisher Information matrix and

$$\tilde{G}(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top I_{\boldsymbol{\theta}^*} \boldsymbol{\theta}$$

Other distributions (cont)

$$\boldsymbol{\theta}_{t+1} \quad \approx \quad \min_{\boldsymbol{\theta}} \Delta_{\tilde{G}}(\boldsymbol{\theta}, \boldsymbol{\theta}_t) + \eta(\boldsymbol{x}_t - g(\boldsymbol{\theta}_t))$$

Update: $\tilde{g}(\boldsymbol{\theta}) = \nabla \tilde{G}(\boldsymbol{\theta}) = I\boldsymbol{\theta}$

$$I\boldsymbol{\theta}_{t+1} = I\boldsymbol{\theta}_t + \eta \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\boldsymbol{x}|\boldsymbol{\theta}_t) \quad \Rightarrow \quad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta I^{-1} \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\boldsymbol{x}|\boldsymbol{\theta}_t)$$

$\Rightarrow$ Natural Gradient

Under mild conditions holds [M]

$$I_{\boldsymbol{\theta}^*} = -H_{\boldsymbol{\theta}^*}$$

Minimizing the negative log-likelihood with natural gradients is equivalent to the Newton-method

Related updates & Information Geometry [MW]

Also often appears in literature:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \eta I_{\boldsymbol{w}}^{-1} \frac{\partial}{\partial \boldsymbol{w}} L(y, \boldsymbol{w} \cdot \boldsymbol{x}), \tag{1}$$

where $I_{\boldsymbol{w}}$ is the Fisher information matrix (at $\boldsymbol{w}$)

So far:

$$\boldsymbol{w}_{t+1} = f^{-1}\left(f(\boldsymbol{w}_t) + \eta \frac{\partial}{\partial \boldsymbol{w}} L(y, \boldsymbol{w} \cdot \boldsymbol{x}) \right) \tag{2}$$

Now: Approximate (2) for small η:

$$\boldsymbol{w}_{t+1} = f(\boldsymbol{w}_t) + \eta J_{f(\boldsymbol{w})} \frac{\partial}{\partial \boldsymbol{w}} L(y, \boldsymbol{w} \cdot \boldsymbol{x}) + \mathcal{O}(\eta^2) \tag{3}$$

where J is the Jacoby matrix for f^{-1} at $f(\boldsymbol{w})$: $J_{i,j} = \left. \frac{\partial f_i^{-1}}{\partial w_j} \right|_{f(\boldsymbol{w})}$

Prior Information [MW]

Similarity between (1) and (3) suggests probabilistic interpretation of (2)

Shown for a special case with prior density on $\boldsymbol{w}$ in product form:

$$\phi(\boldsymbol{w}) = \prod_{i=1}^{N} \phi_i(w_i)$$

Then the "preferential metric" ($\sim$ Fisher information matrix) is given by

$$I_{\boldsymbol{w}} = \begin{pmatrix} \phi_1(w_1)^2 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \phi_N(w_N)^2 \end{pmatrix}$$

Interpretation

The Jacobian is diagonal if $f(\boldsymbol{w}) = (f_1(w_1), \ldots, f_N(w_N))^\top$

$$J_{\boldsymbol{w}} = \begin{pmatrix} \left(\frac{\partial f_1(w_1)}{\partial w}\right)^{-1} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \left(\frac{\partial f_N(w_N)}{\partial w}\right)^{-1} \end{pmatrix}$$

Hence:

If $\phi_i(w_i) = \sqrt{\frac{\partial f_i(w_i)}{\partial w}}$, then $I_{\boldsymbol{w}}^{-1} = J_{\boldsymbol{w}}$

Eucl. Gradients vs. Exponentiated Gradients

For standard gradient descent $\Delta_F(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_2$, we have $f(\boldsymbol{w}) = \boldsymbol{w}$
$\Rightarrow \phi_i(w_i) = 1$ (improper uniform prior)

For exponentiated gradient descent $\Delta_F(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \sum_j w_j \log \frac{w_j}{\tilde{w}_j}$, we have $f_i(w_i) = \log w_i$
$\Rightarrow \phi_i(w_i) = 1/\sqrt{w_i}$ (improper prior inducing sparseness)

Matches our experimental observations!

Application: Adaptive Channel Equalization

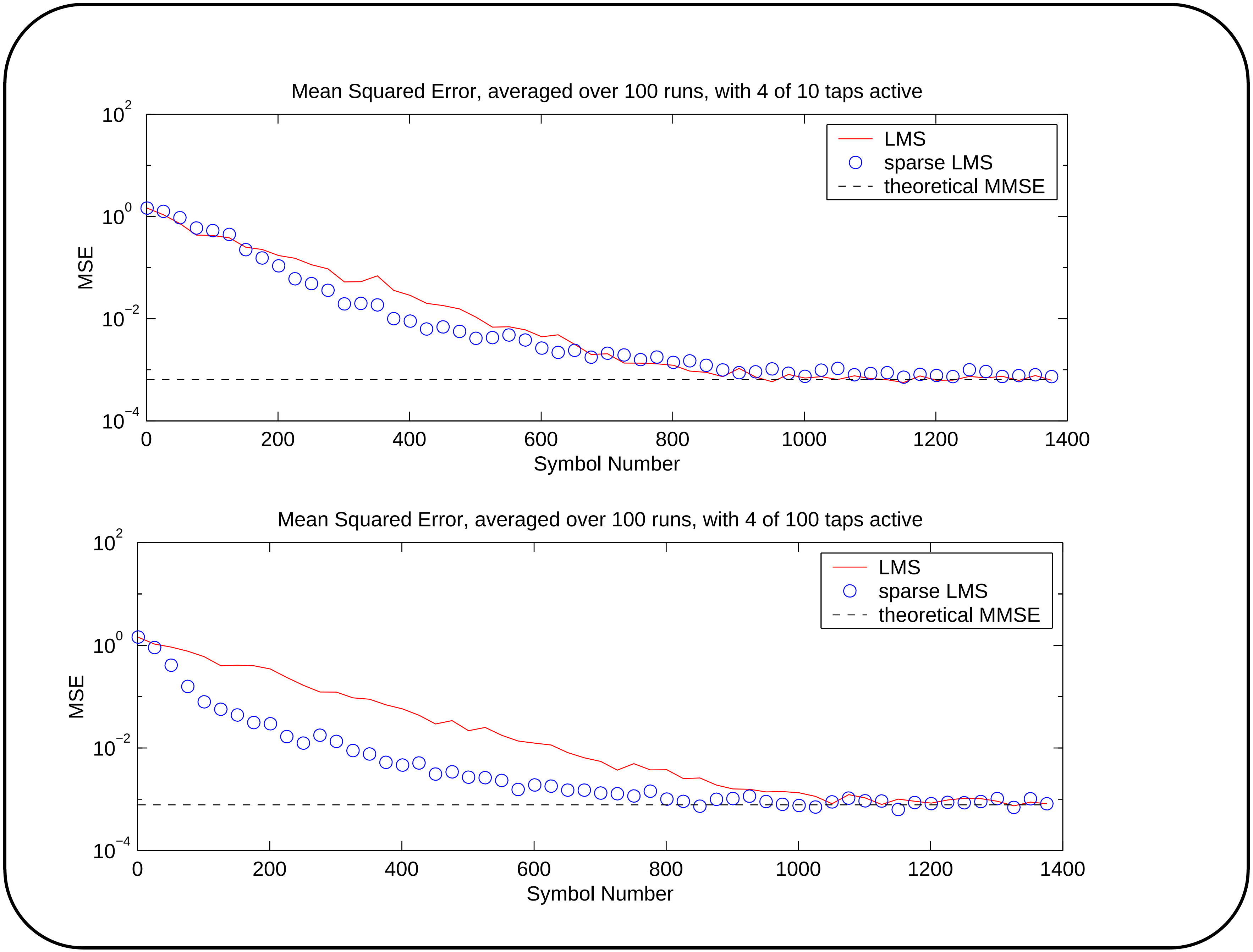
- Online Linear Regression Problem:
  $\Rightarrow$ Find $\boldsymbol{w}$ such that $(y - \boldsymbol{w} \cdot \boldsymbol{x})^2$ is minimized

- Common approach:

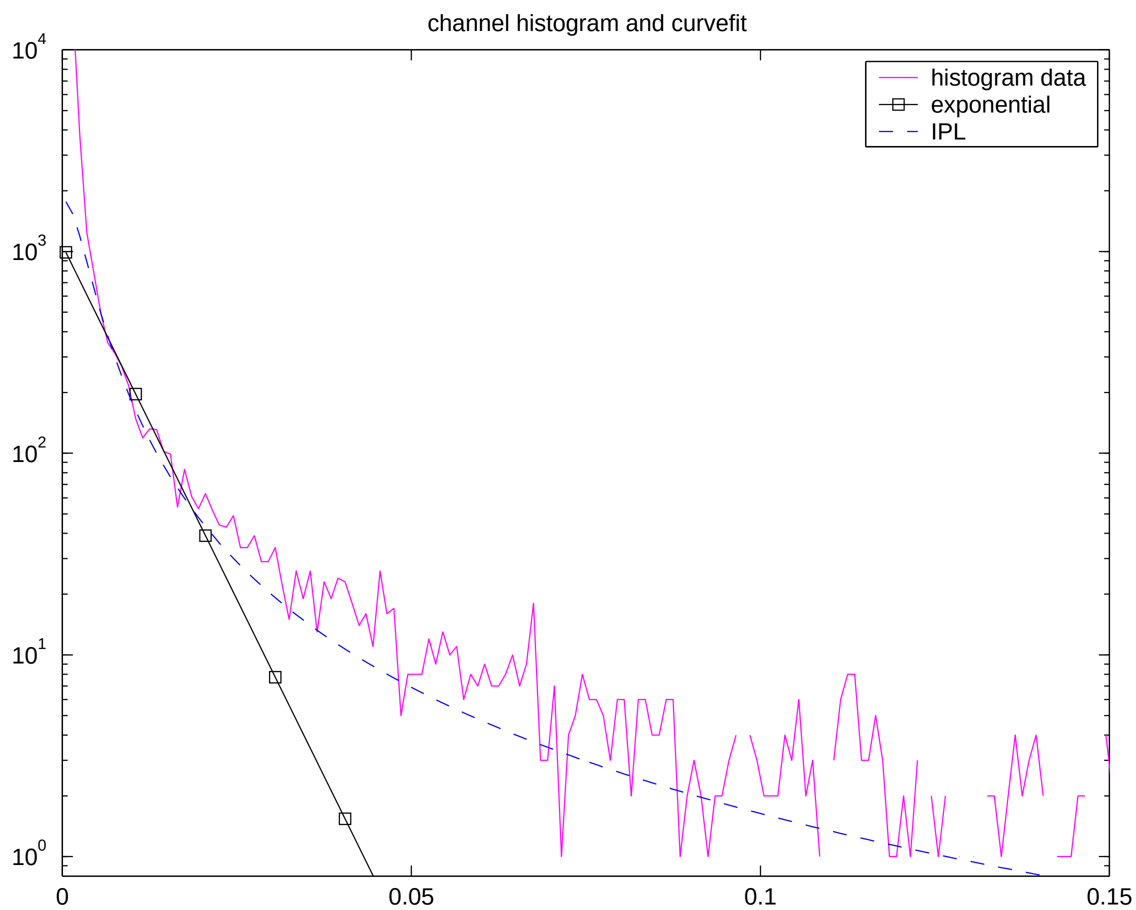
$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta(y - \boldsymbol{w}_t \cdot \boldsymbol{x}_t)\boldsymbol{x}_t$$

- But: Many coefficients are zero, or close to zero [MSWJ]
  $\Rightarrow$ Use exponentiated gradient descent or approximate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta w_t(y - \boldsymbol{w}_t \cdot \boldsymbol{x}_t)\boldsymbol{x}_t$$

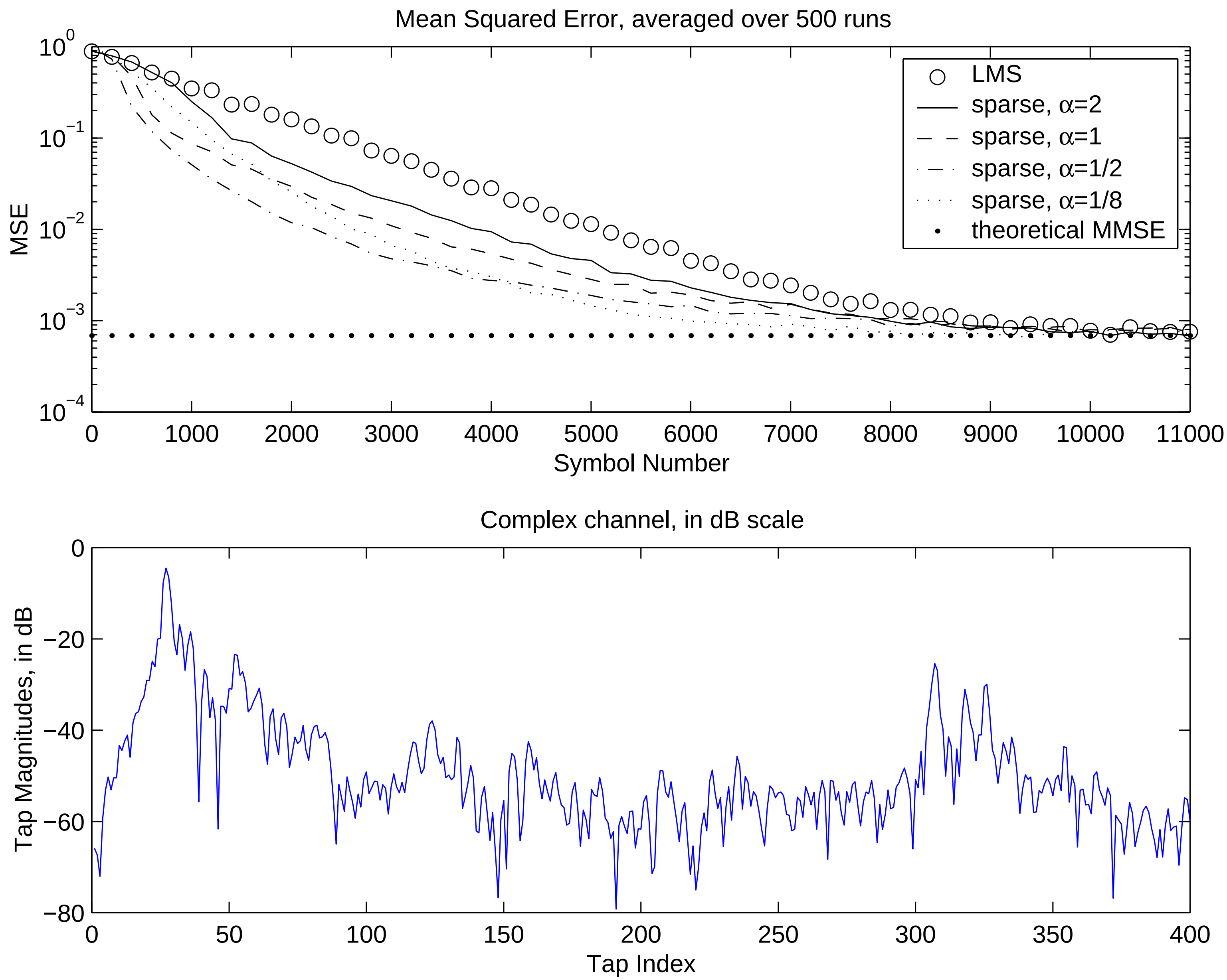

Mean Squared Error, averaged over 100 runs, with 4 of 10 taps active
MSE
Symbol Number
LMS
sparse LMS
theoretical MMSE
Mean Squared Error, averaged over 100 runs, with 4 of 100 taps active
MSE
Symbol Number
LMS
sparse LMS
theoretical MMSE

"Estimating the prior" from Histograms



$$\phi(w) = c\exp(-\alpha w)$$

$$\phi(w) = \frac{c}{|w|^{\alpha} + \epsilon}$$


Mean Squared Error, averaged over 500 runs
MSE
Symbol Number
LMS
sparse, α=2
sparse, α=1
sparse, α=1/2
sparse, α=1/8
theoretical MMSE
Complex channel, in dB scale
Tap Magnitudes, in dB
Tap Index

Application: Disk Spin Down [HLSS]

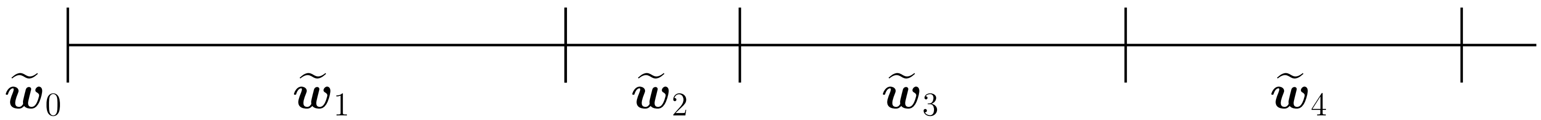
Problem of adapt. spinning down hard disks in mobile computers

Common approach: fixed time-out (e.g. 2 min)
$\Rightarrow$ suboptimal, changing usage patterns, etc.

Idea:
- Use many experts with different time-outs
- predict as good as the best time-out, if nothing changes
- switch fast to another time-out, if necessary

Needs very efficient algorithm!

Comparator shifts with time



On-line examples and on-line comparator

$$\sum_{t=1}^{T} L_t(\boldsymbol{w}_t) \quad - \quad \inf_{\widetilde{\boldsymbol{w}}_t} \sum_{t=1}^{T} \left(L_t(\widetilde{\boldsymbol{w}}_t) + \Delta(\widetilde{\boldsymbol{w}}_{t-1}, \widetilde{\boldsymbol{w}}_t)\right)$$

total loss of on-line algorithm		total loss of shifting off-line comparator

Modifications to the Expert Algorithm [HW]

Predict $\hat{y}_t = \boldsymbol{v}_t \cdot \boldsymbol{x}_t$,

where $v_{t,i} = \frac{w_{t,i}}{\sum_{i=1}^{n} w_{t,i}}$

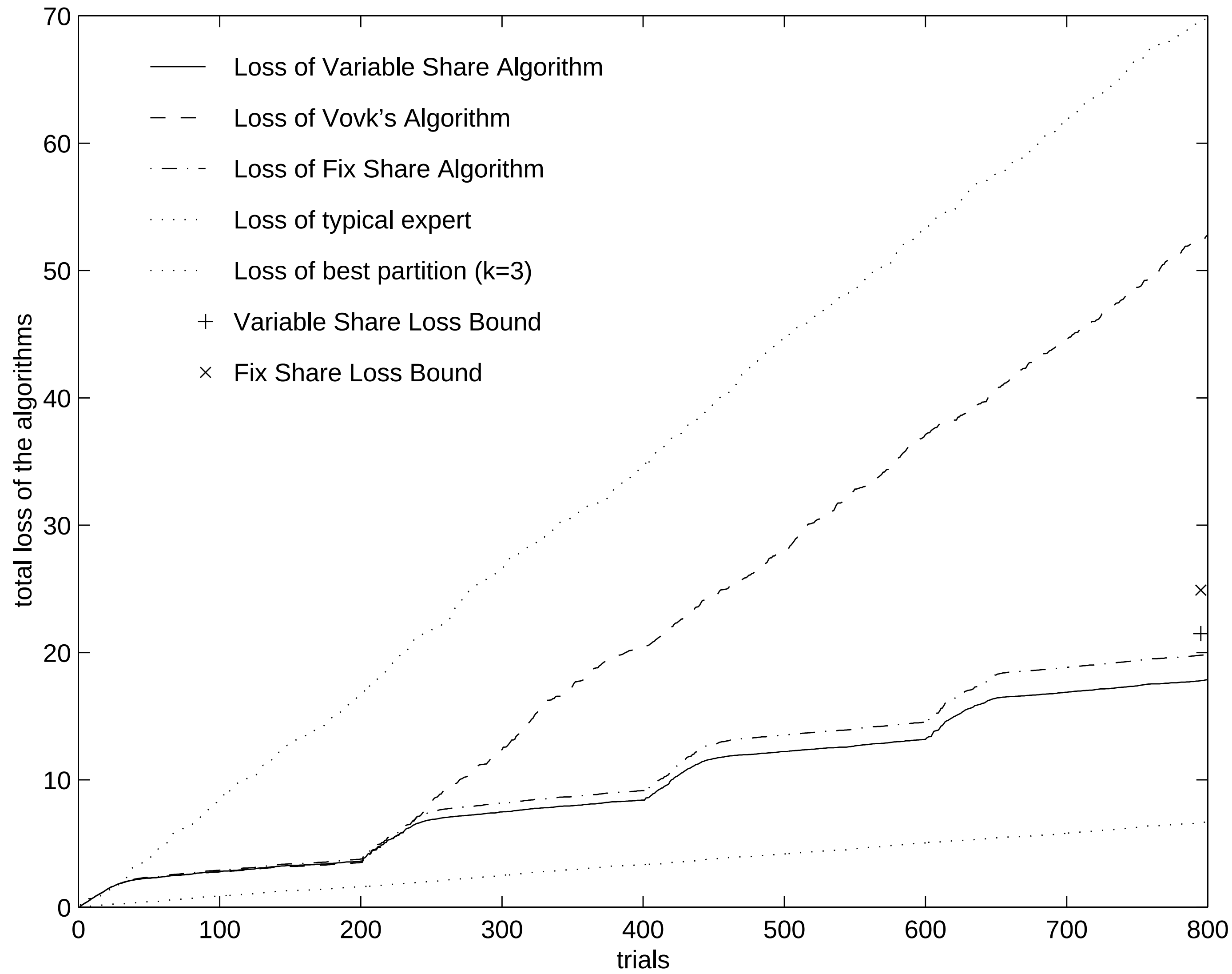
Loss Update $w_{t,i} := w_{t,i} e^{-\eta L_{y_t, x_{t,i}}}$

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

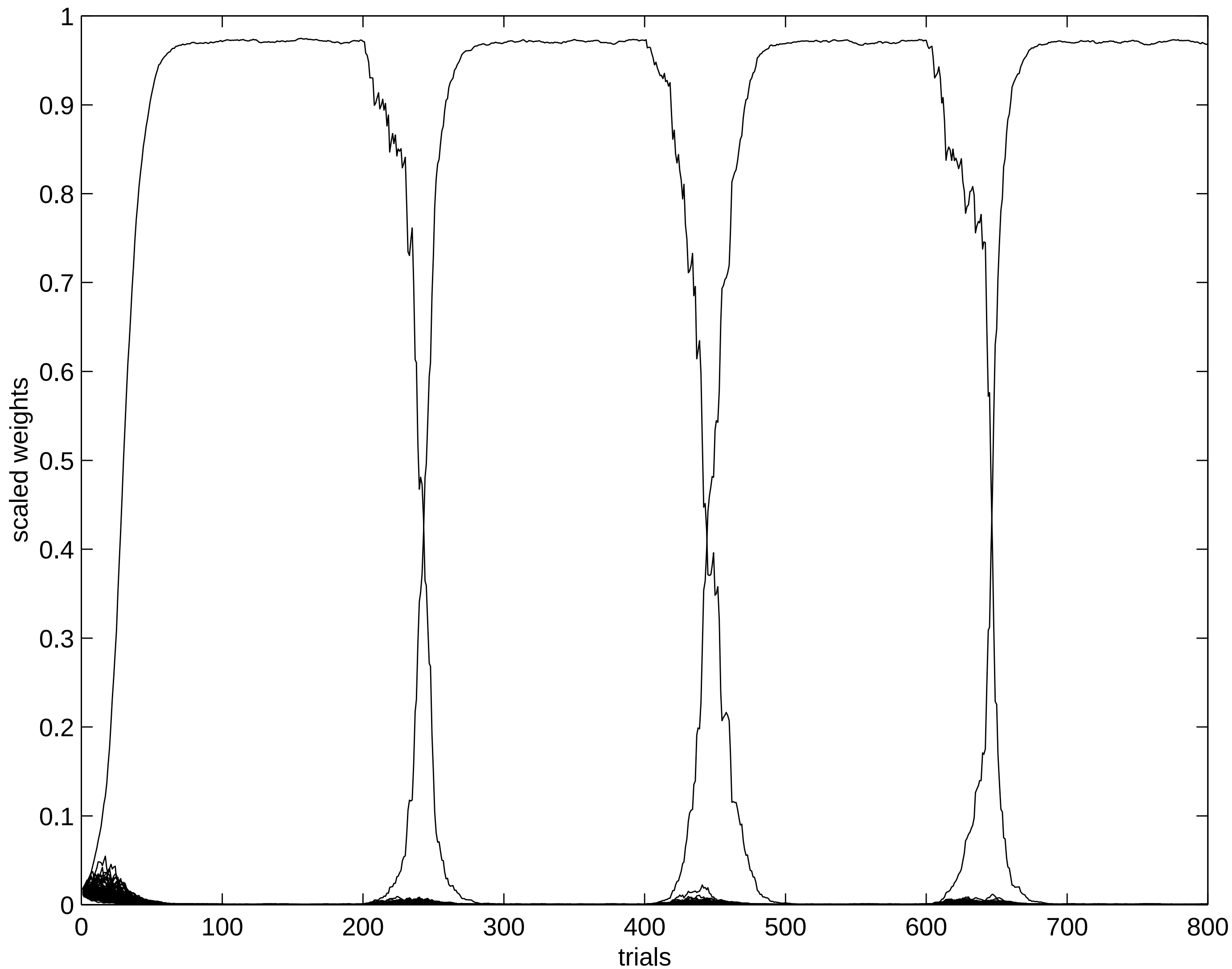
- Static-expert: Blank
- Fixed-share:
  Each expert sends $\frac{\alpha}{n-1}$ of its weight to the other $n-1$ experts
- Variable-share: Replace $\frac{\alpha}{n-1}$ by

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

Loss of share algorithms versus Static Expert Algorithm



Relative weigths of the Fixed Share Algorithm



Shifting bounds

- The Static Expert bounds

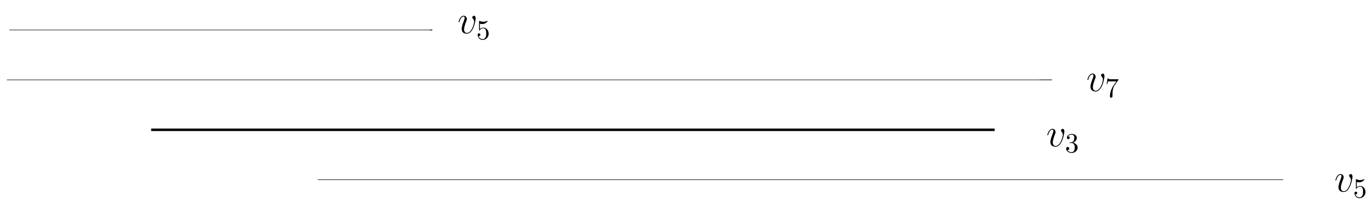
$$L_{\text{Alg}}(S) \leq \min_i L_i(S) + O(\log n))$$

become [HW]

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

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

- For shifting disjunctions [AW]



Schedule τ

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

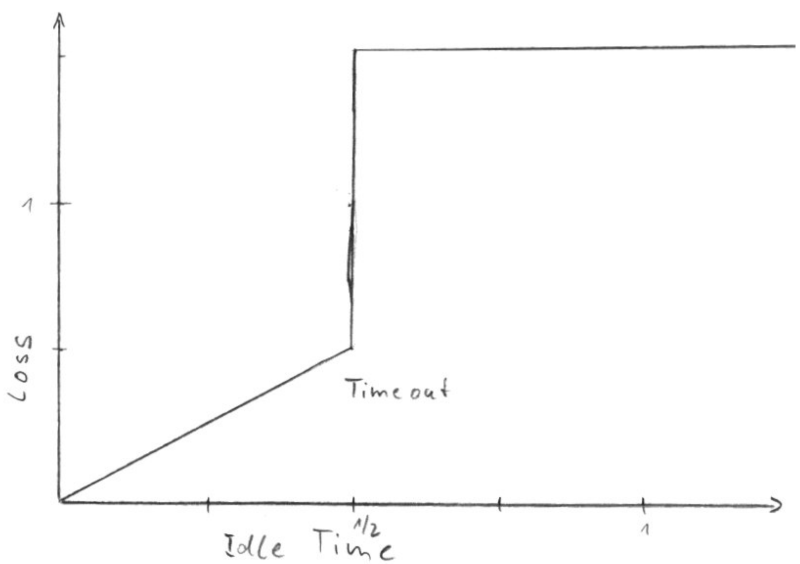
where size(τ) is # of literals in τ
and $A_{\boldsymbol{\tau}}(S)$ is # of attrib. errors w.r.t. τ

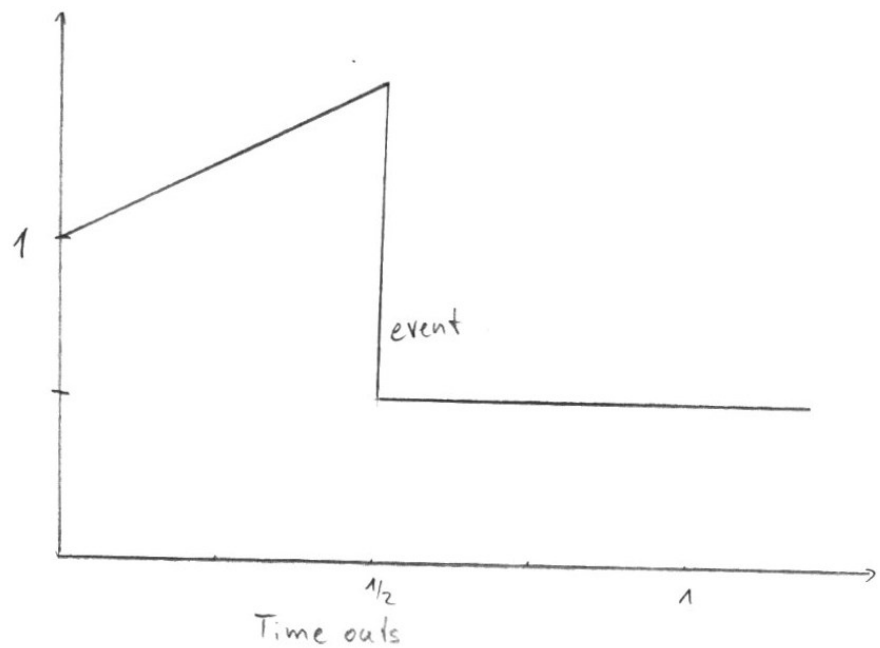
Back to the Disk Spin down problem

Loss function: Costs for spinning up/down, idle, etc.

L("idle-time", "time-out")

Measure time and loss in multiples of the "Spin-down cost"





Weight Updates:

$$w'_{t,i} := w_{t,i} e^{-\eta L_{y_t, x_{t,i}}}$$

Share Updates:

$$w_{t,i} = w'_{t,i} + \frac{\alpha}{n-1} \sum_{j \neq i} w'_{t,j}$$

Weighted average of experts determines time-out

$$\text{timeout}_t = \sum_i w_{t,i} \text{timeout}_i / \sum_i w_{t,i}$$

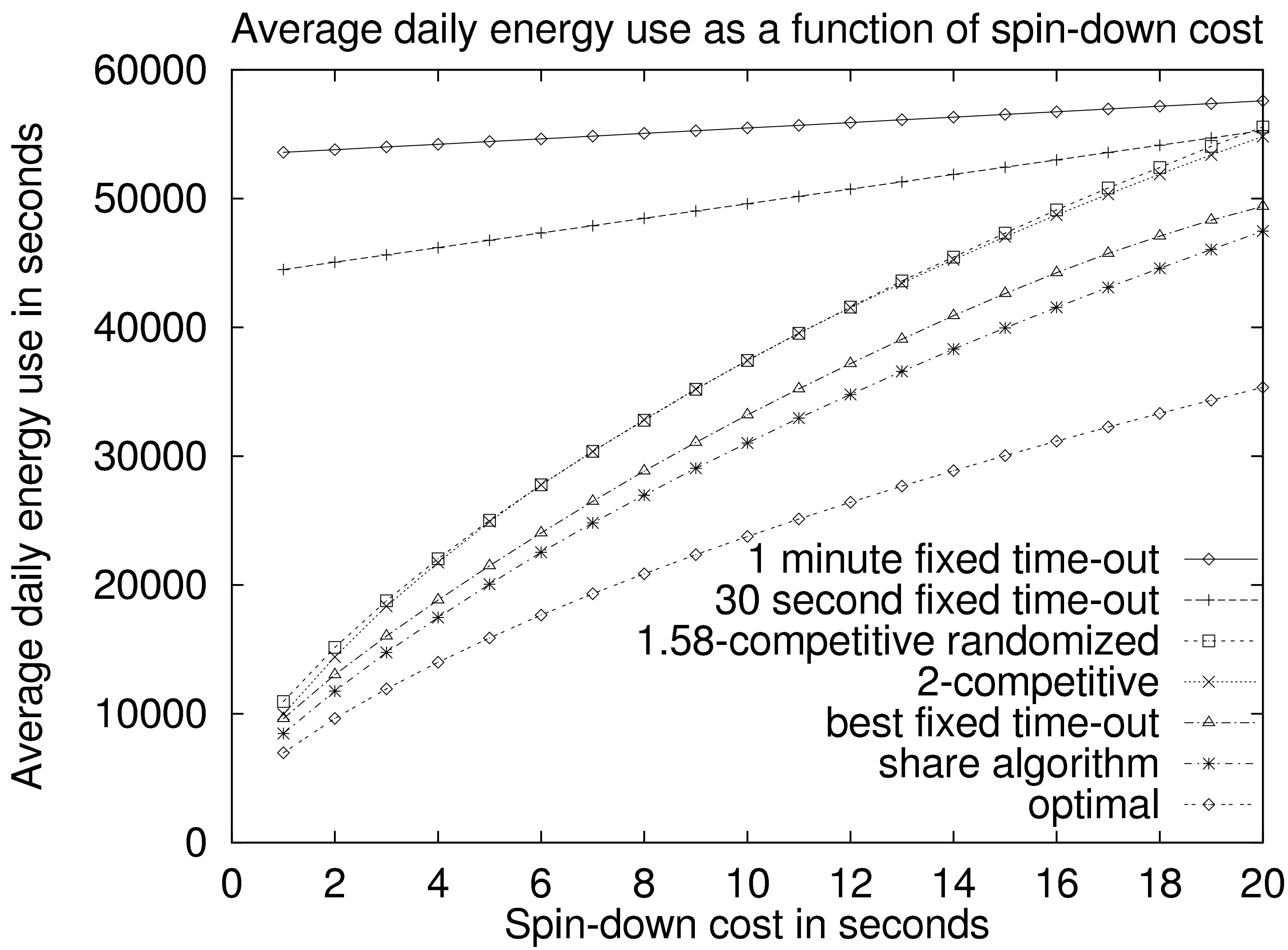
Problem: Non-convex loss function

Randomized prediction of experts determines time-out

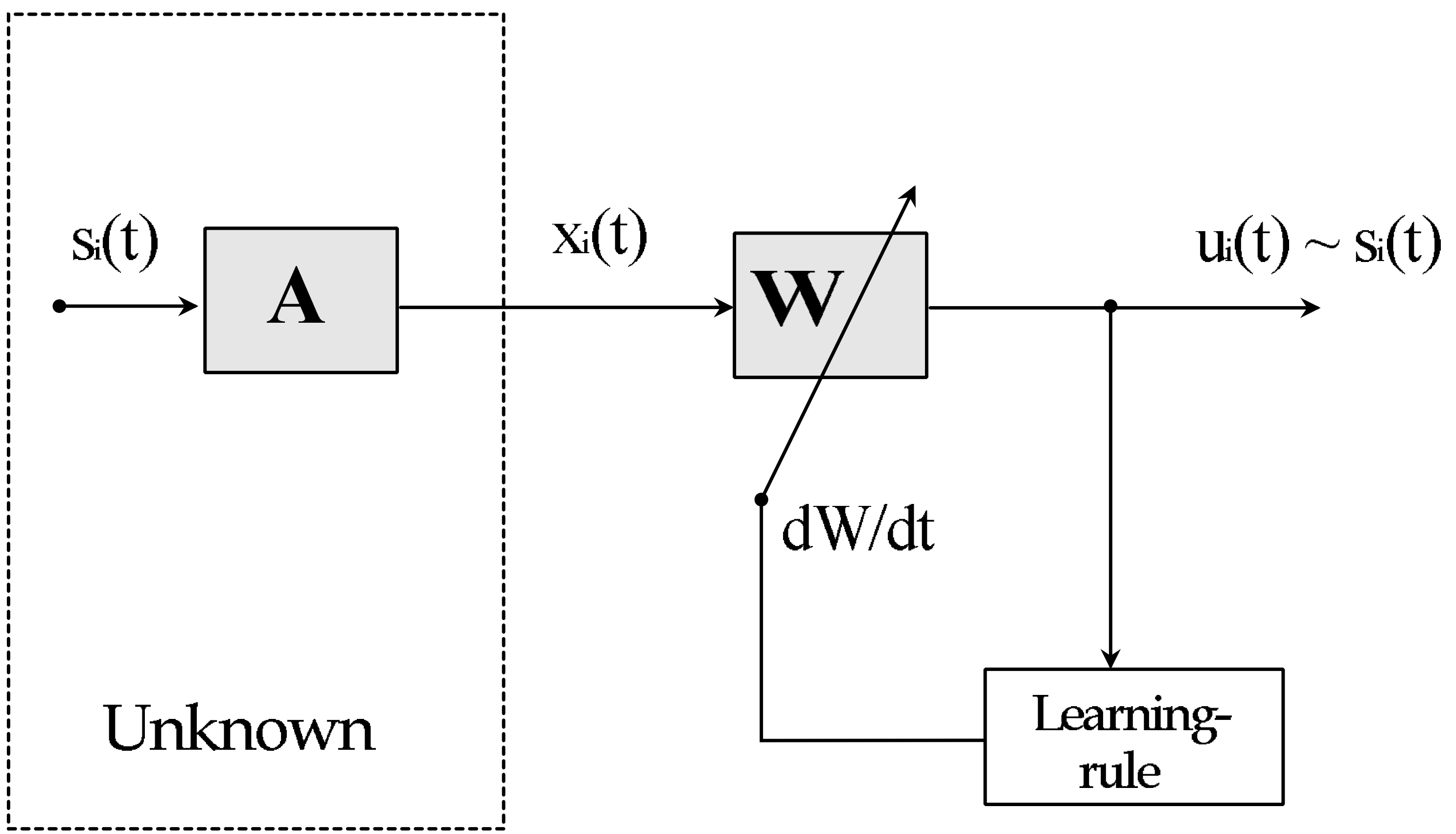
$$\text{timeout}_t = \text{timeout}_i$$

where i is chosen with $w_{t,i} / \sum_j w_{t,j}$

Disk Spin down



Application: Adaptive Source Separation [MM+]



Strategies for Online Learning

far from solution: large steps with constant η (phase 1)

close to solution: small steps $\eta \sim 1/t$ (phase 2)

But: When to go from phase 1 to phase 2 and **when back** automatically?

What if rule changes ?

- η large and constant: ok, but large remaining error
- η small and constant: bad
- $\eta = 1/t$: very bad

Goal: notice when rule changes and choose best strategy

The spirit of Sompolinsky et al.'s algorithm [BSS]

$$\hat{\boldsymbol{w}}_{t+1} = \hat{\boldsymbol{w}}_t - \eta_t H^{-1}(\hat{\boldsymbol{w}}_t) \frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}; \hat{\boldsymbol{w}}_t), \quad (4)$$

$$\eta_{t+1} = \eta_t + \alpha \eta_t \left(\beta \left(L(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}; \hat{\boldsymbol{w}}_t) - \hat{R} \right) - \eta_t \right), \quad (5)$$

H is Hessian, $\hat{R}$ is estimator of the optimum, i.e.

$$\hat{R}_{t+1} = (1 - \gamma)\, \hat{R}_t + \gamma L(\boldsymbol{x}_{t+1}; \boldsymbol{y}_{t+1}; \hat{\boldsymbol{w}}_t). \quad (6)$$

Intuition:

- far from minimum: **accelerate!** $\longrightarrow$ large η
- close to minimum: **annealing!** $\longrightarrow$ small $\eta = 1/t$

continuous version:

$$\frac{d}{dt}\boldsymbol{w}(t) = -\eta(t)H_*(\boldsymbol{w}(t) - \boldsymbol{w}_*), \tag{7}$$

$$\frac{d}{dt}\xi(t) = -\lambda\eta(t)\xi(t), \tag{8}$$

$$\frac{d}{dt}\eta(t) = \alpha\eta(t)\left(\beta|\xi(t)| - \eta(t)\right), \tag{9}$$

where $\xi(t) = \boldsymbol{\nu}^T H_*(\boldsymbol{w}(t) - \boldsymbol{w}_*)$.

solutions:

$$\begin{cases} \xi(t) = \frac{1}{\beta}\left(\frac{1}{\lambda} - \frac{1}{\alpha}\right) \cdot \frac{1}{t}, \\ \eta(t) = \frac{1}{\lambda} \cdot \frac{1}{t}. \end{cases} \tag{10}$$

Demonstration

We use two audio files (sampling rate 8kHz; sun audio file):

$$\vec{s_t} = \begin{cases} s_t^1 & : \text{“speech”} \\ s_t^2 & : \text{“music”} \end{cases} \tag{11}$$

Sources are mixed:

$$\begin{cases} \vec{x_t} = (I + A)\vec{s_t} & \text{if } \; 0s < t < 2.5s \text{ and } 6.5s \le t \le 10s, \\ \vec{x_t} = (I + B)\vec{s_t} & \text{if } \; 2.5s \le t < 6.5s, \end{cases} \tag{12}$$

where mixing matrices are

$$A = \begin{pmatrix} 0 & 0.9 \\ 0.6 & 0 \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & 0.8 \\ 0.4 & 0 \end{pmatrix}.$$

Other Applications

- Calendar managing

  Many features (sleeping experts) [Bl,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]