On-Line Learning of Non-Stationary Data

Manfred K. Warmuth
UC, Santa Cruz

Joint work with:

Olivier Bousquet

Mark Herbster

Outline

- Motivate on-line learning
- Motivate relative loss bounds
- Halving Algorithm as example
- Loss Update
- Flavor of proof techniques
- Comparator on-line as well
- How to adapt the algs
- Future work

On-Line Learning

Loop

Get next instance

Predict

Get label

Incur loss

- Choose comparison class of predictors
  e.g. linear
- Goal:
  Do well compared to best off-line comparator
- No statistical assumptions on the data

Experts as Comparators

	experts						
	E_1	E_2	E_3	E_n	prediction	*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	$x_{t,1}$	$x_{t,2}$	$x_{t,3}$	$x_{t,n}$	$\hat{y}_t$	y_t	$\|y_t - \hat{y}_t\|$

Protocol of the Master Algorithm

For $t = 1$ To T Do

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

Minimax Algorithm for T Trials

- Learner against adversary

$$\sup_{\boldsymbol{x}_1} \inf_{\hat{y}_1} \sup_{y_1} \quad \sup_{\boldsymbol{x}_2} \inf_{\hat{y}_2} \sup_{y_2} \quad \ldots \quad \sup_{\boldsymbol{x}_T} \inf_{\hat{y}_T} \sup_{y_T}$$

$$\sum_{t=1}^{T} L(y_t, \hat{y}_t) \quad - \quad \inf_{\mathbf{f} \in \mathbf{C}} \left(\sum_{t=1}^{T} L(f(\boldsymbol{x}_t), y_t) \right)$$

total loss of on-line algorithm

total loss of off-line algorithm

- C **is comparison class**
- Minimax algorithm usually intractable

What kind of performance can we expect ?

- $L_{1..T,A}$ be the total loss of algorithm A
- $L_{1..T,i}$ be the total loss of i-th expert E_i

- Form of bounds

$$\forall S: \quad L_{1..T,A} \;\leq\; \min_i \left(L_{1..T,i} + c\ \log n\right)$$

  where c is constant

- Bounds the loss of the algorithm relative to the loss of best expert

General Expert Algorithm

- Master algorithm predicts with weighted average

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

- The weights are updated according to the Loss Update

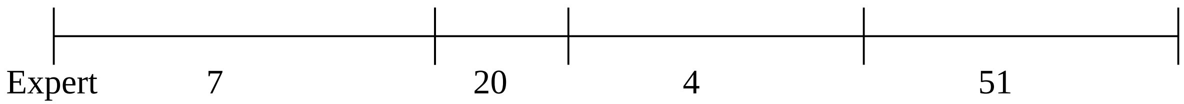
$$v_{t+1,i} := \frac{v_{t,i} \;\; e^{-\eta \, L_{t,i}}}{\text{normaliz.}}$$

where $L_{t,i}$ is loss of expert i in trial t

$\longrightarrow$ Weighted Majority Algorithm **[LW89]**

$\longrightarrow$ Generalized by Vovk **[Vovk90]**

Comparator Changes with Time



- Off-line alg. partitions sequence into sections
  and chooses best expert in each section
- Goal:
  Do well compared to best off-line partition
- Problem:
  Loss Update learns too well
  and does not recover fast enough

Modifications to the Expert Alg. [HW98]

- Predict $\hat{y}_t = \boldsymbol{v}_t \cdot \boldsymbol{x}_t$
- Loss Update

$$v_{t,i}^m := \frac{v_{t,i} e^{-\eta L_{t,i}}}{\text{normaliz.}}$$

- Share Update
  - Static Expert

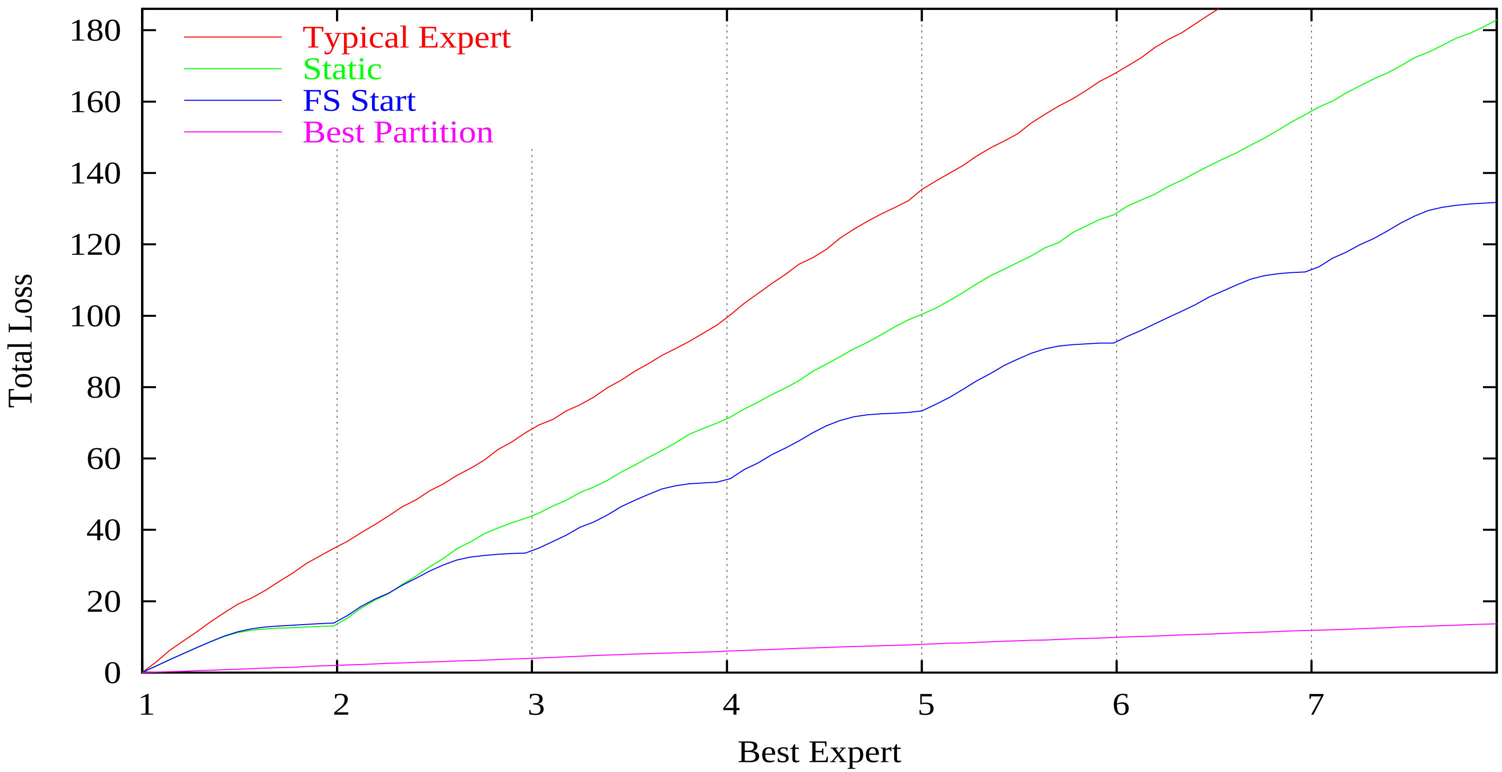
$$\boldsymbol{v}_{t+1} = \boldsymbol{v}_t^m$$

  - Fixed Share to Start Vector ($\alpha \in [0, 1)$)

$$\boldsymbol{v}_{t+1} = (1 - \alpha)\boldsymbol{v}_t^m + \alpha \boldsymbol{v}_0$$

where $\boldsymbol{v}_0 = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$

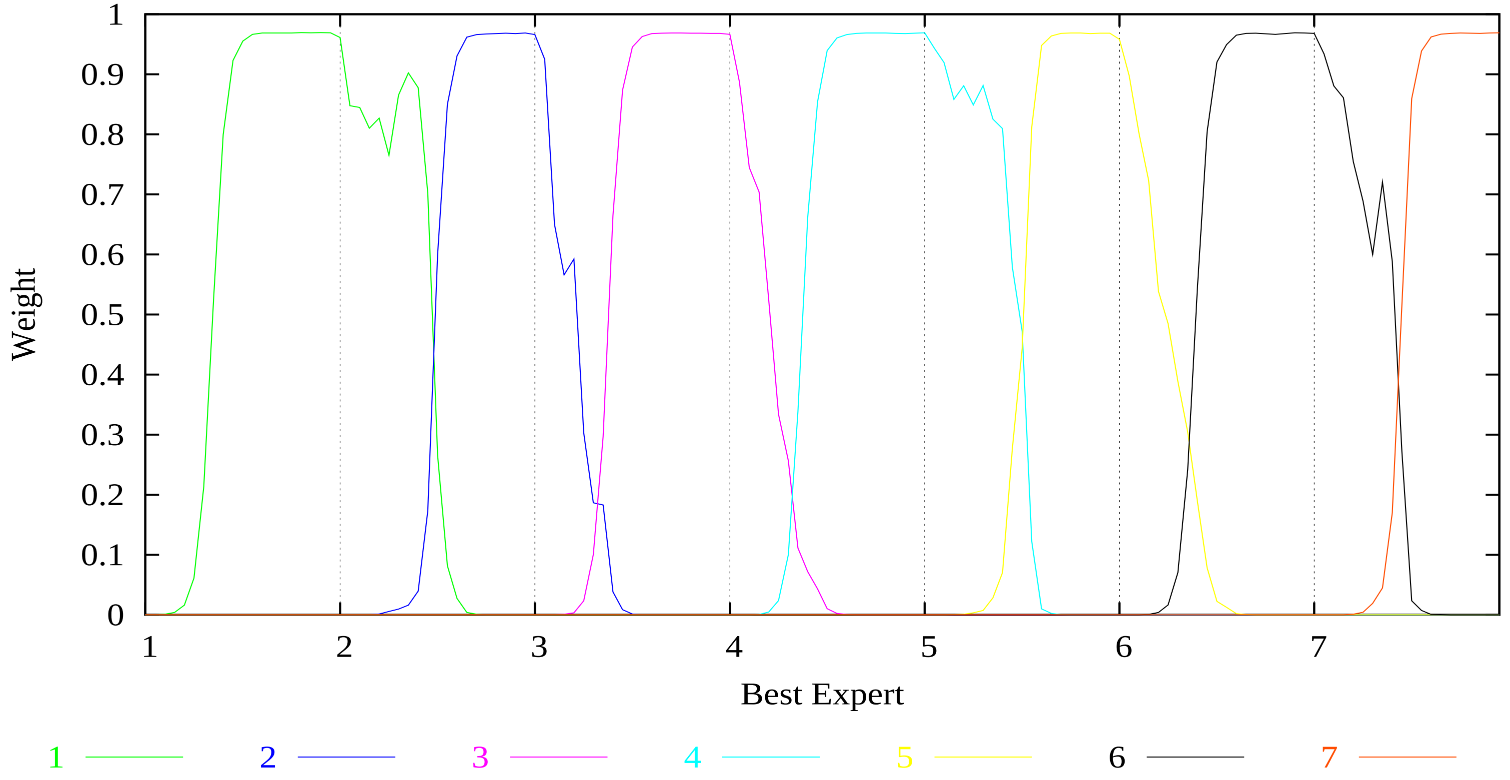
Static Expert Alg. Versus Share Algs.



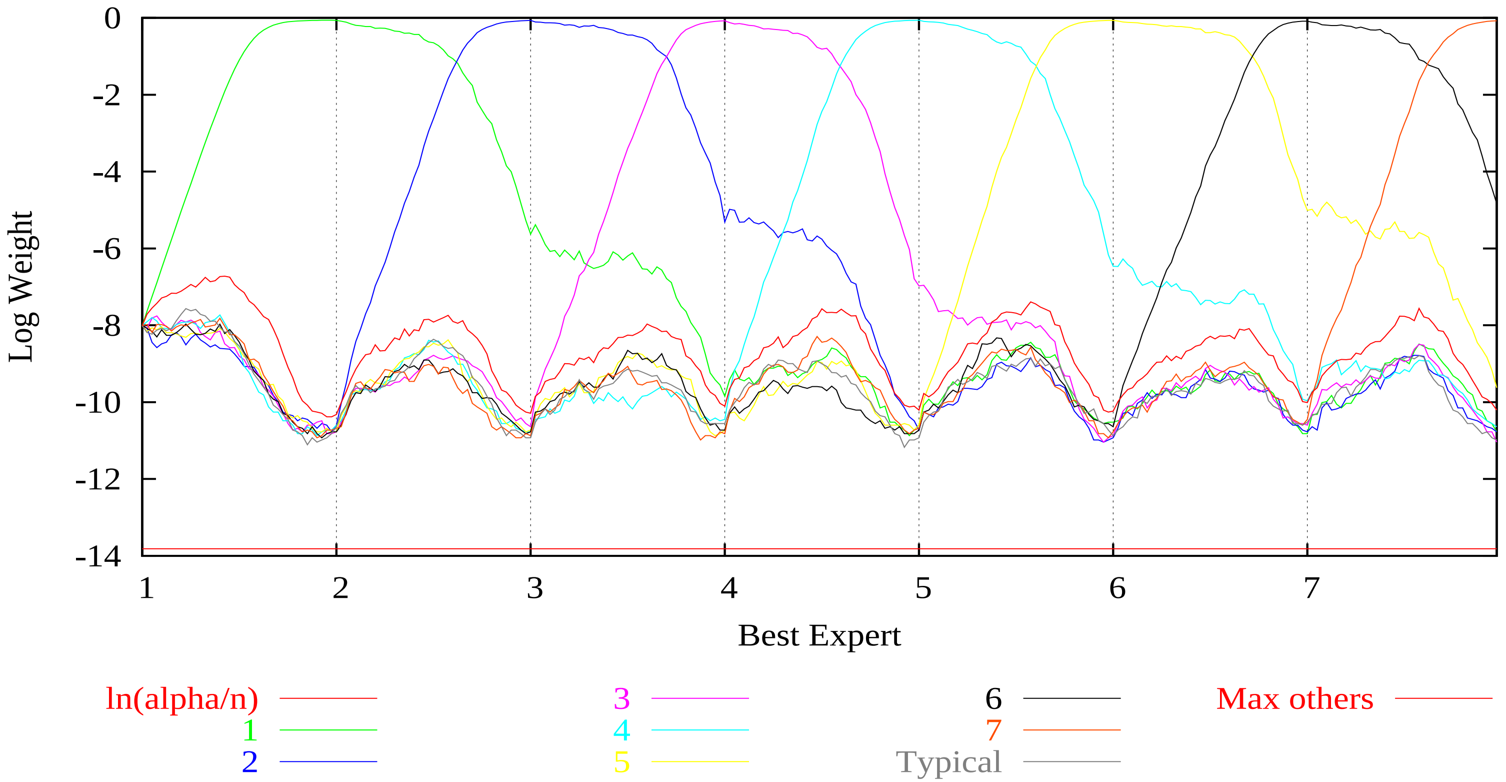
- Square loss, target outcome always 0, experts have predictions between 0 and $1/2$ uniform for typical experts and restricted to $[0, 0.12]$ for current best expert
- $T = 1400$ trials, $n = 20000$ experts, $k = 6$ shifts

Weights of Fixed Share Alg.

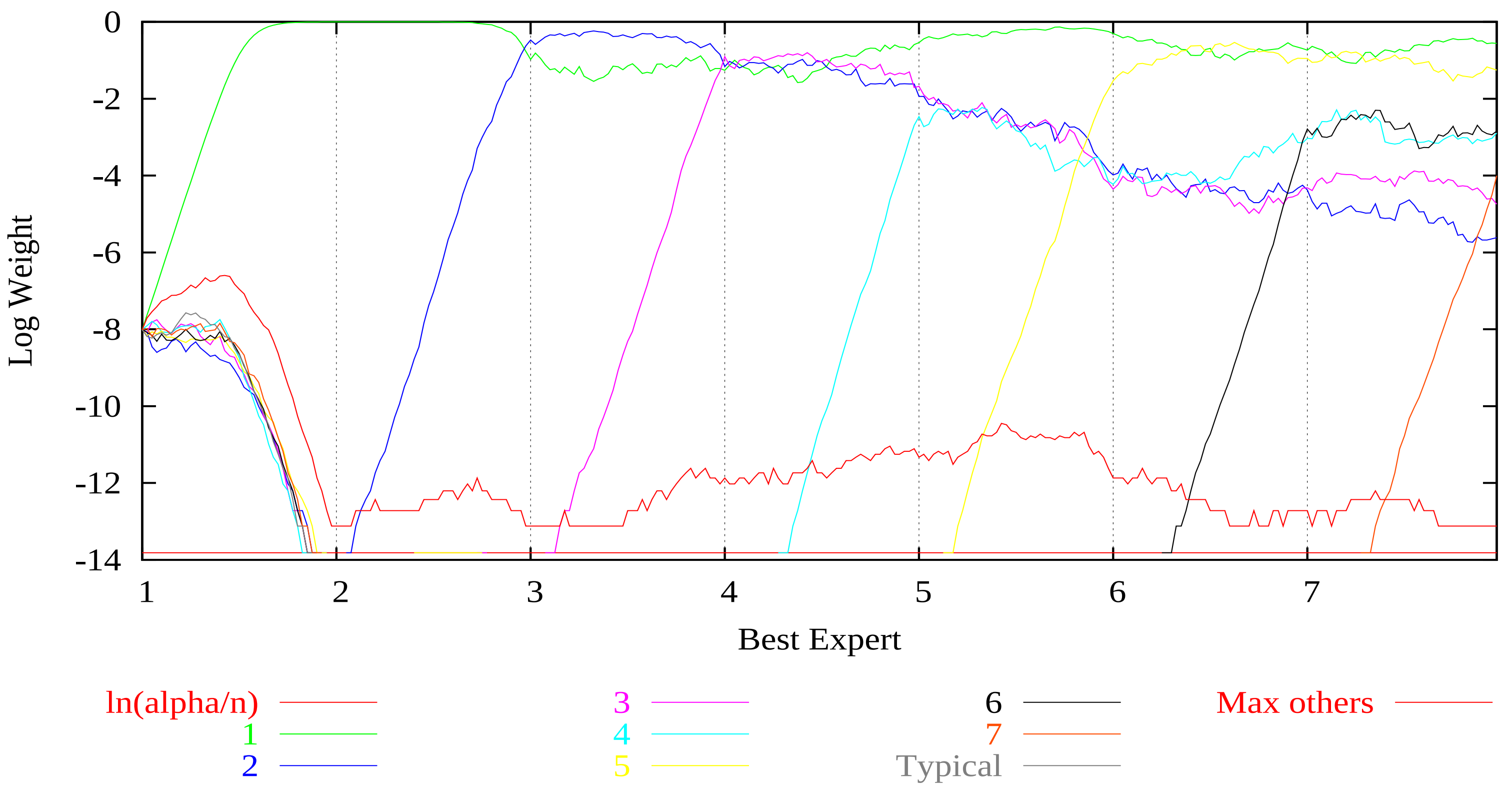
- Tracks the best expert



Weights of Fixed Share Alg.



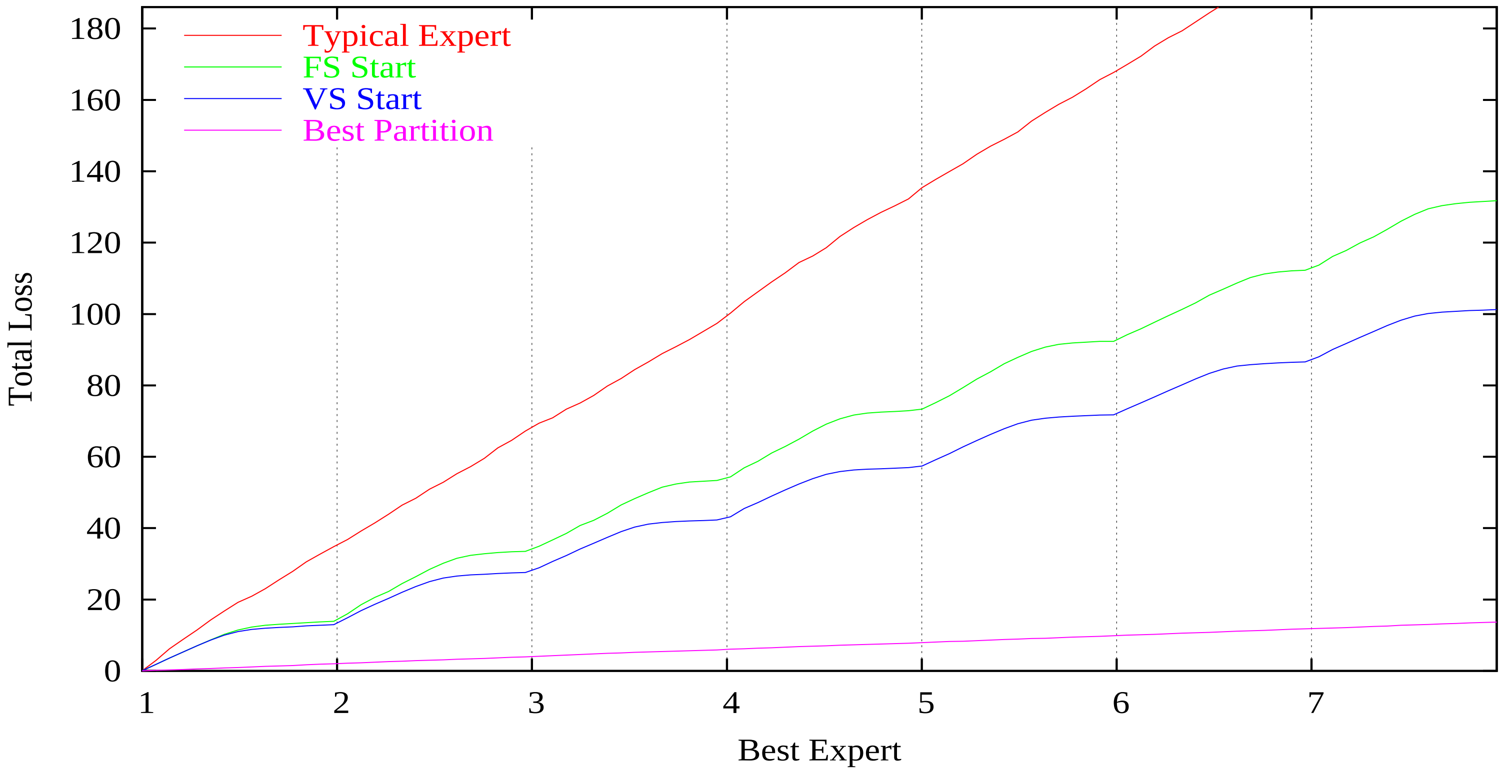
Weights of Static Expert Alg.



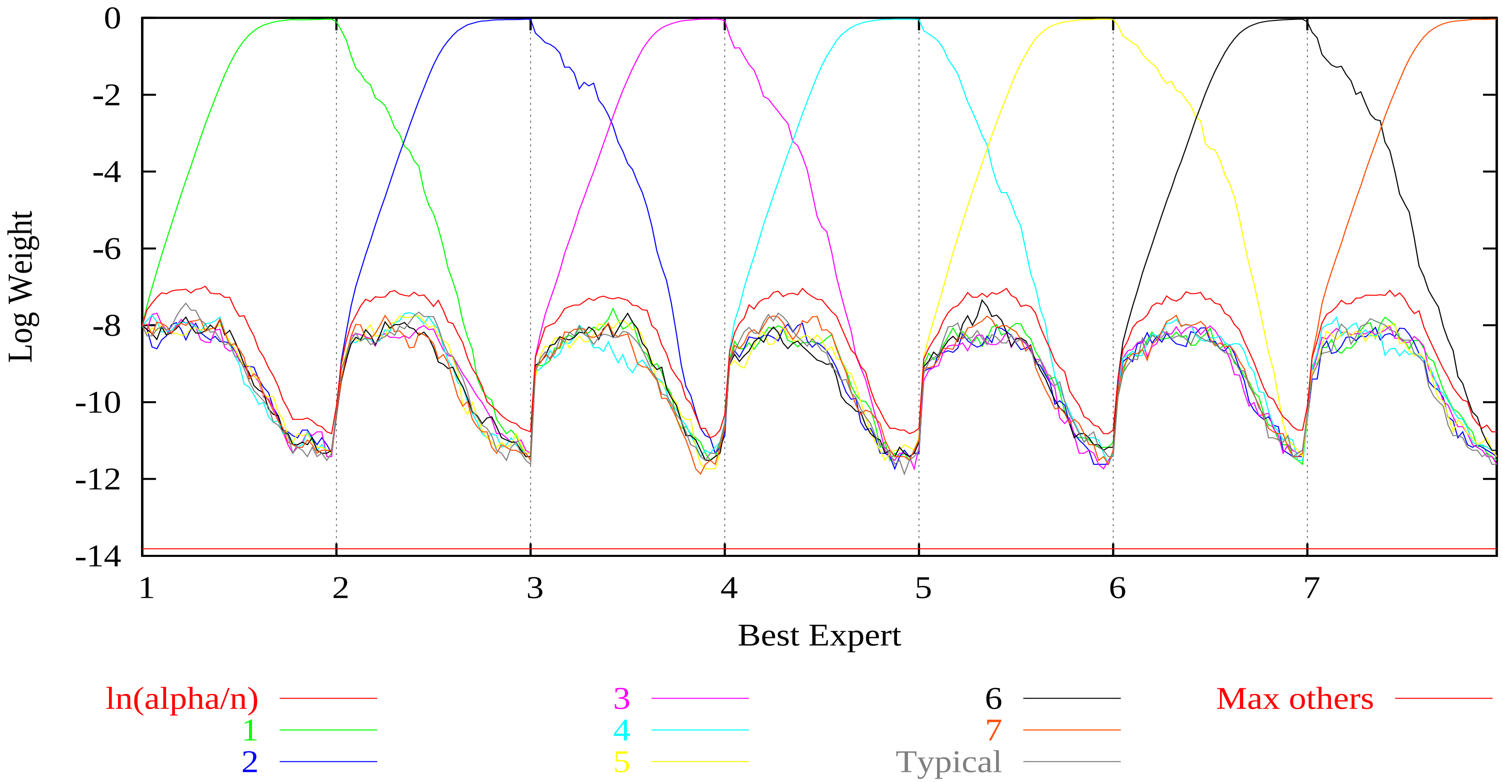
Variable Share [HW]

- Variable Share to Start Vector
  - Replace
$$\boldsymbol{v}_{t+1,i} = (1-\alpha)\,\boldsymbol{v}^m_{t,i} + \alpha\,\frac{1}{n}$$
  - by
$$\boldsymbol{v}_{t+1,i} = (1-\alpha)^{L_{t,i}}\,\boldsymbol{v}^m_{t,i} + \left(1 - \sum_{i=1}^{n}(1-\alpha)^{L_{t,i}}\boldsymbol{v}^m_{t,i}\right)\frac{1}{n}\text{where } L_{t,i} \in [0,1]$$

Fixed Share vs. Variable Share



Weights of Variable Share Alg.



Shifting Bounds

- Recall Static Expert bound

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

  - Comparison class: set of experts

- Bounds for Share Algs.

$$L_{\text{Alg}}(S) \le \min_P \left(L_P(S) + O(\#\text{ of bits for } P)\right)$$

  - Comparison class: set of partitions
  - # of bits for partitions with k shifts:

$$k \log n + \log \binom{T}{k}$$

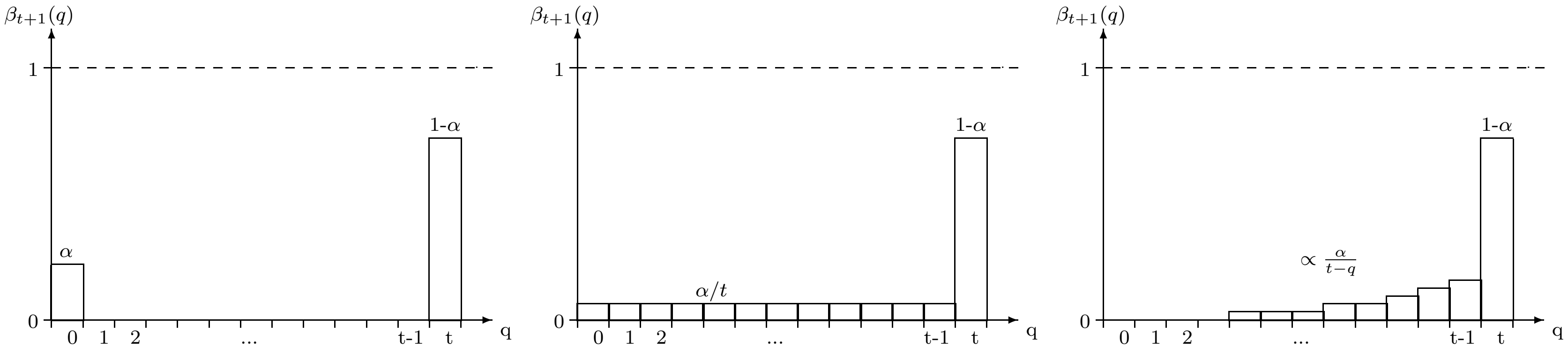
Yoav's Open Problem

- Number of possible experts n is large $n \approx 10^6$
- Experts in partition from small subset of size m $m \approx 10$
- # of bits for partitions with k shifts:

$$\log \binom{n}{m} + k \log m + \log \binom{T}{k}$$

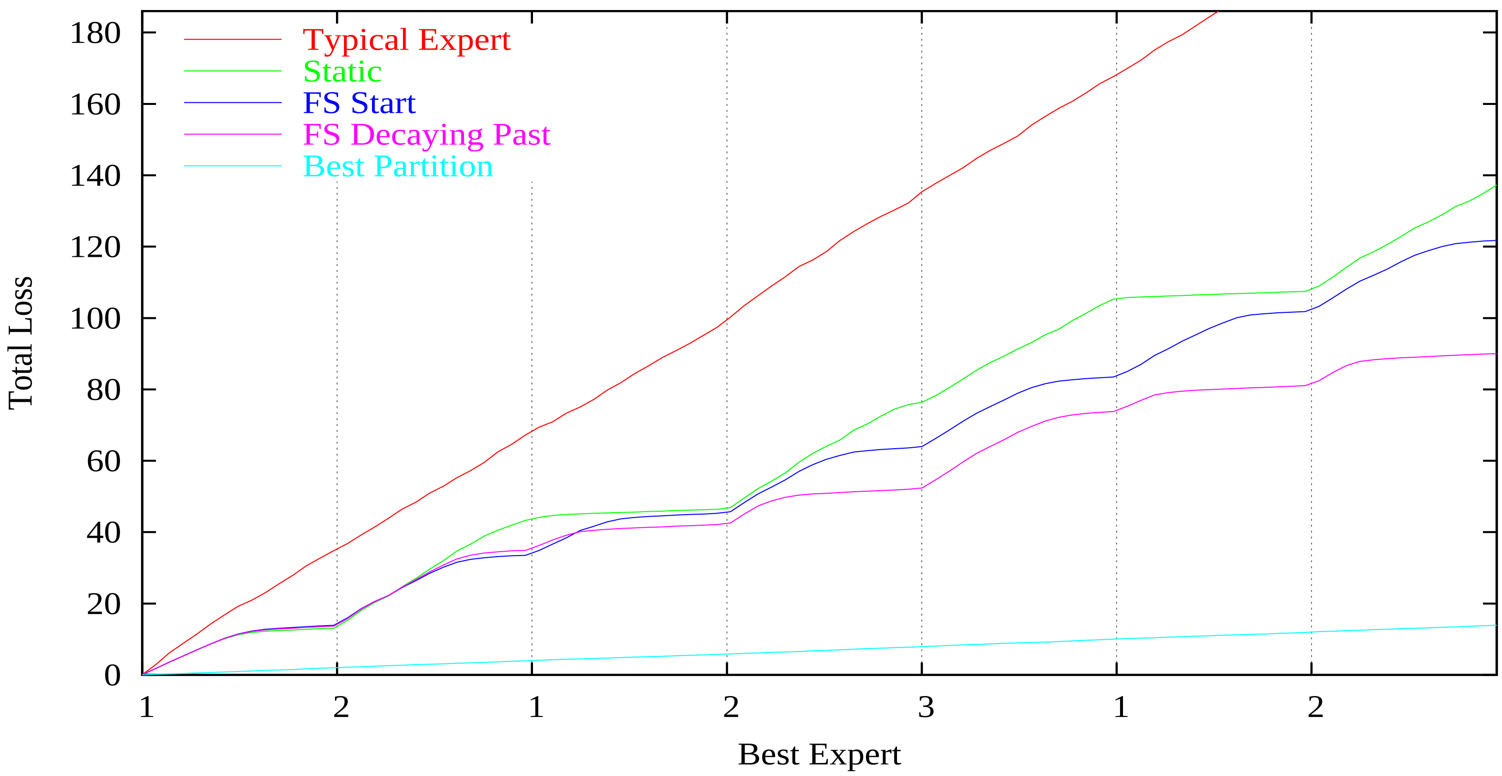
Mixing Update

- Predict $\hat{y}_t = \boldsymbol{v}_t \cdot \boldsymbol{x}_t$
- Loss Update $v^m_{t,i} := \frac{v_{t,i} e^{-\eta L_{t,i}}}{\text{normaliz.}}$
- Mixing Update: $\boldsymbol{v}_{t+1} = \sum_{q=0}^{t} \beta_{t+1,q} \boldsymbol{v}^m_q$
- Mixing scheme



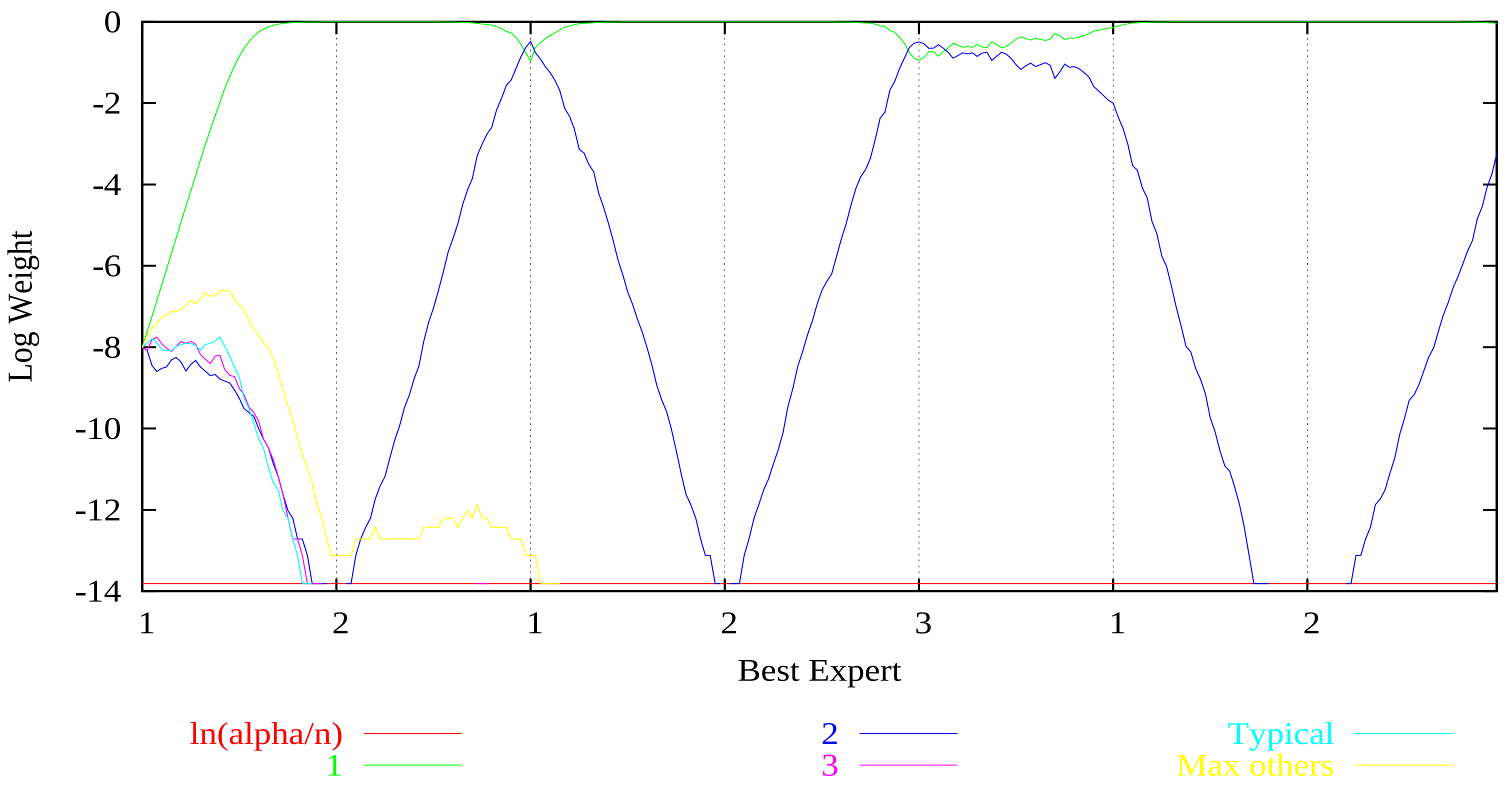
FS to Start Vector FS to Uniform Past FS to Decaying Past

Total Loss Plots

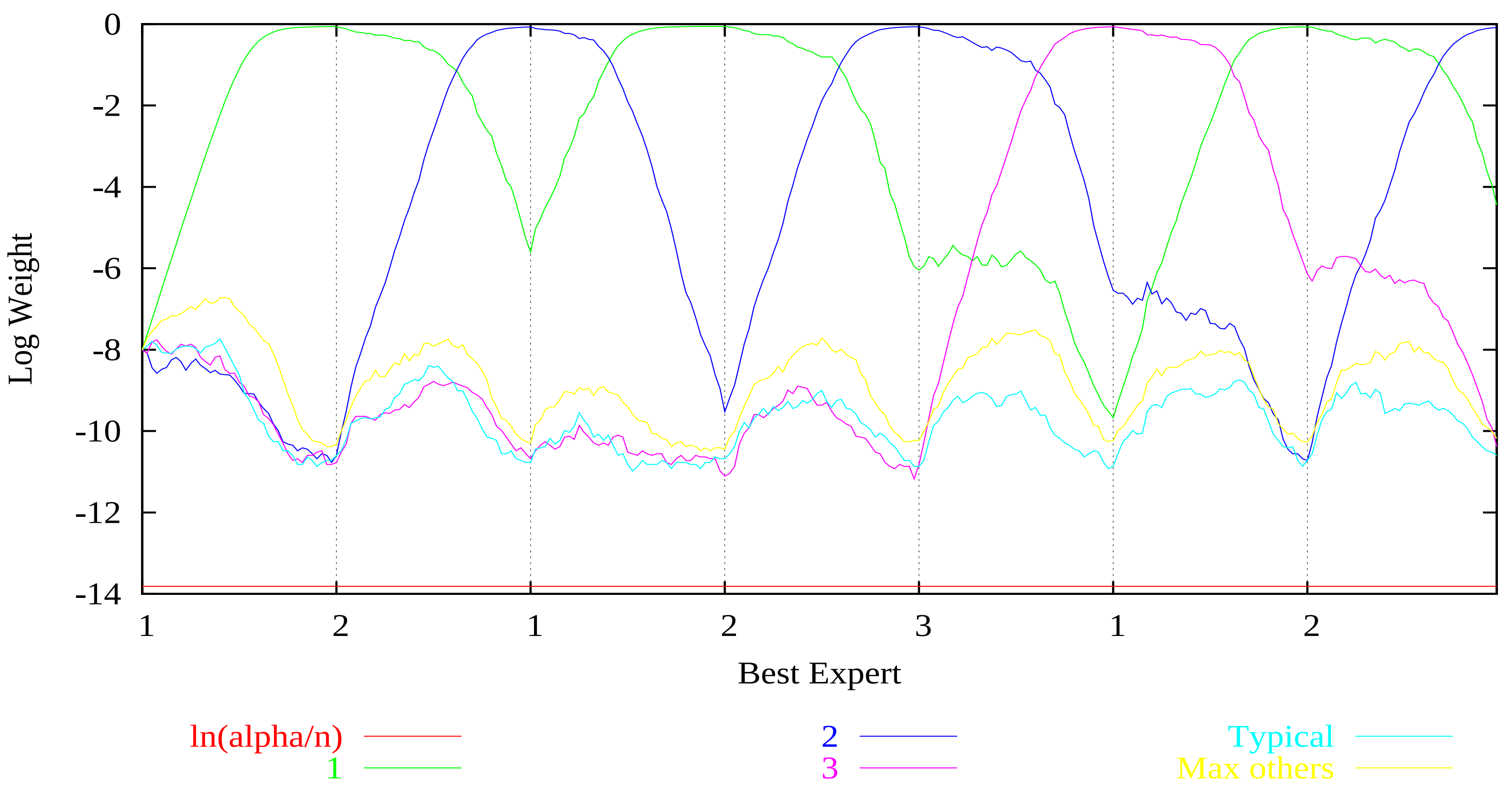


- Square loss, target outcome always 0, experts have predictions between 0 and $1/2$ uniform for typical experts and restricted to $[0, 0.12]$ for current best expert
- $T = 1400$ trials, $n = 20000$ experts, $k = 6$ shifts, $m = 3$ experts in the small subset

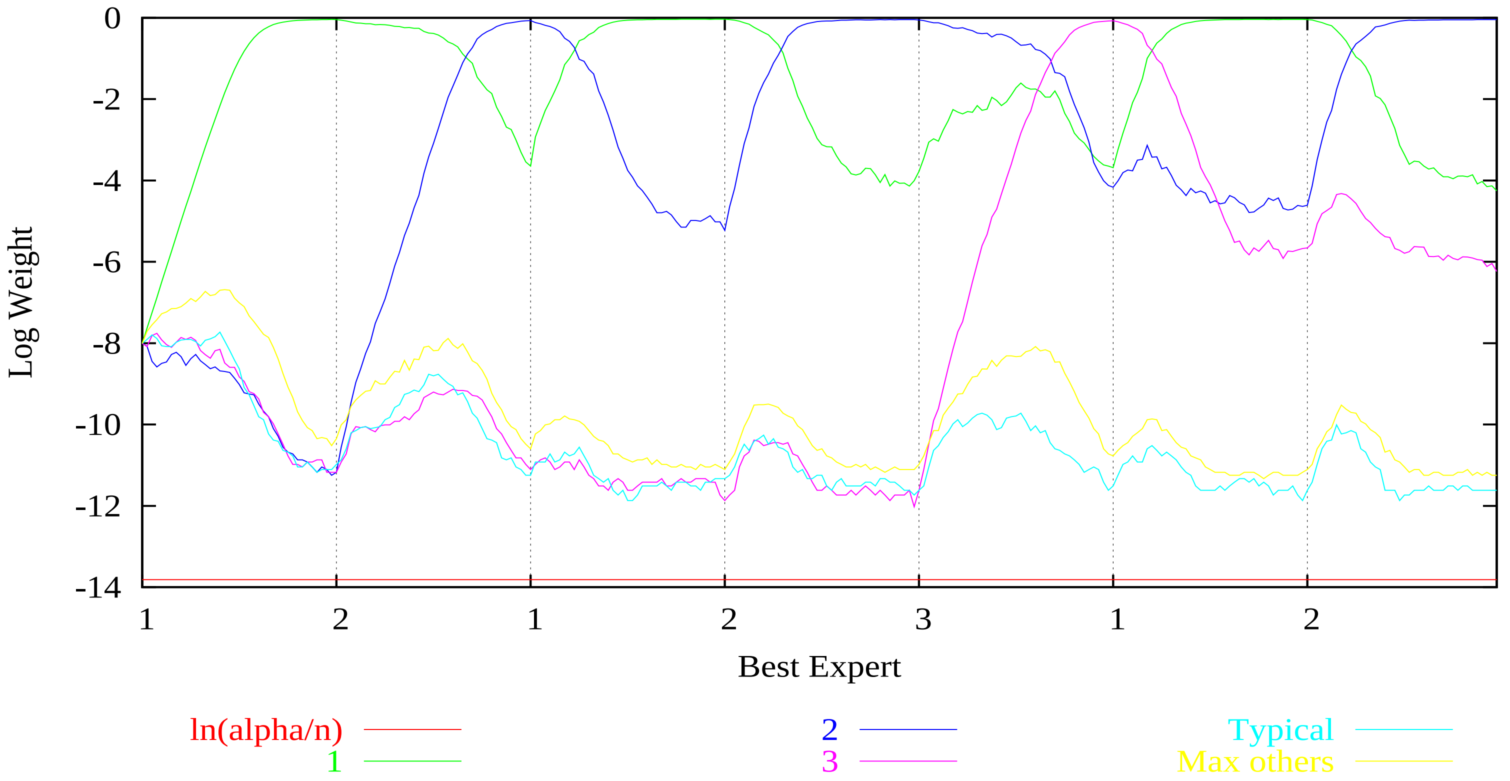
Static Experts



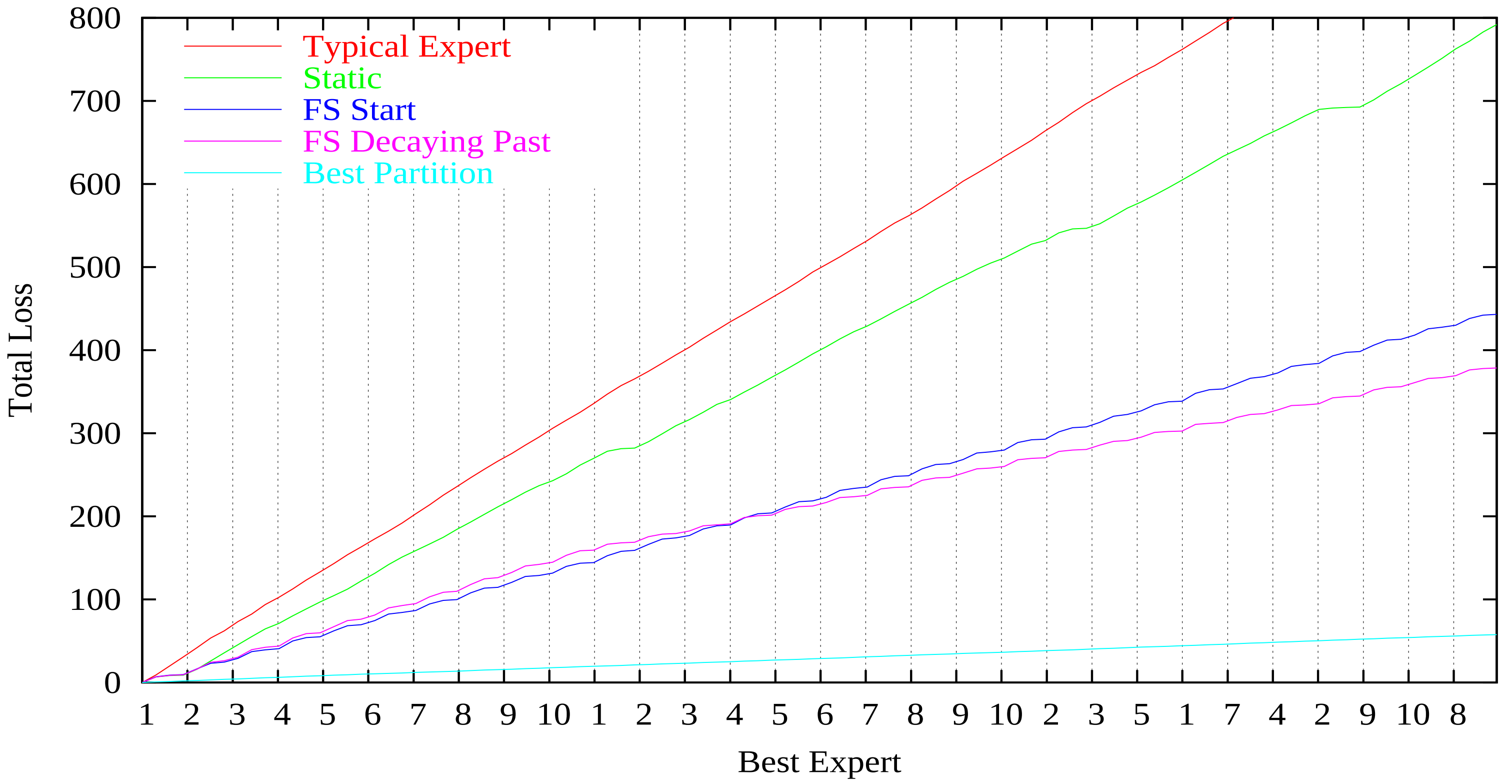
Fixed Share to Start Vector



Fixed Share to Decaying Past - Log Weights

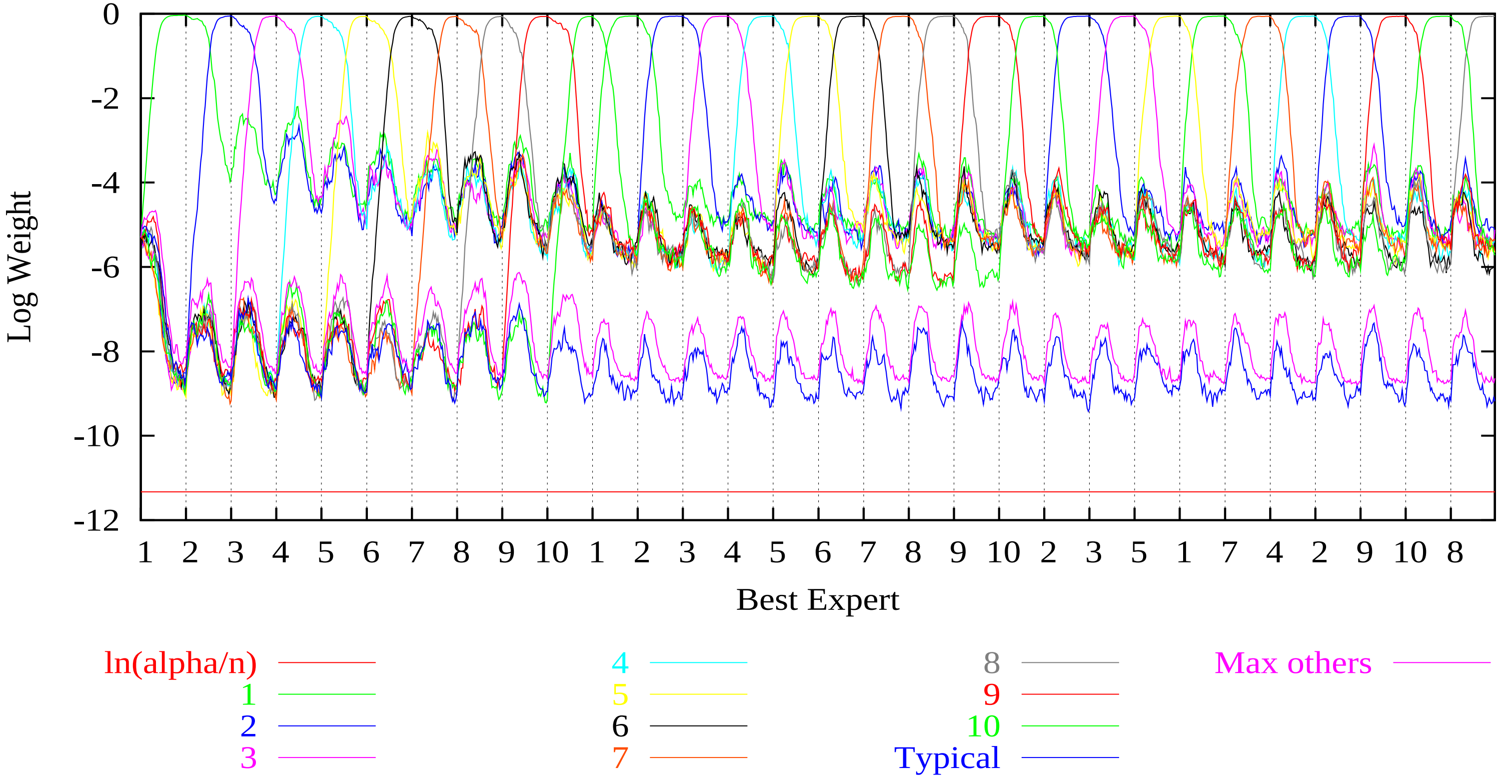


More Experts Remembered



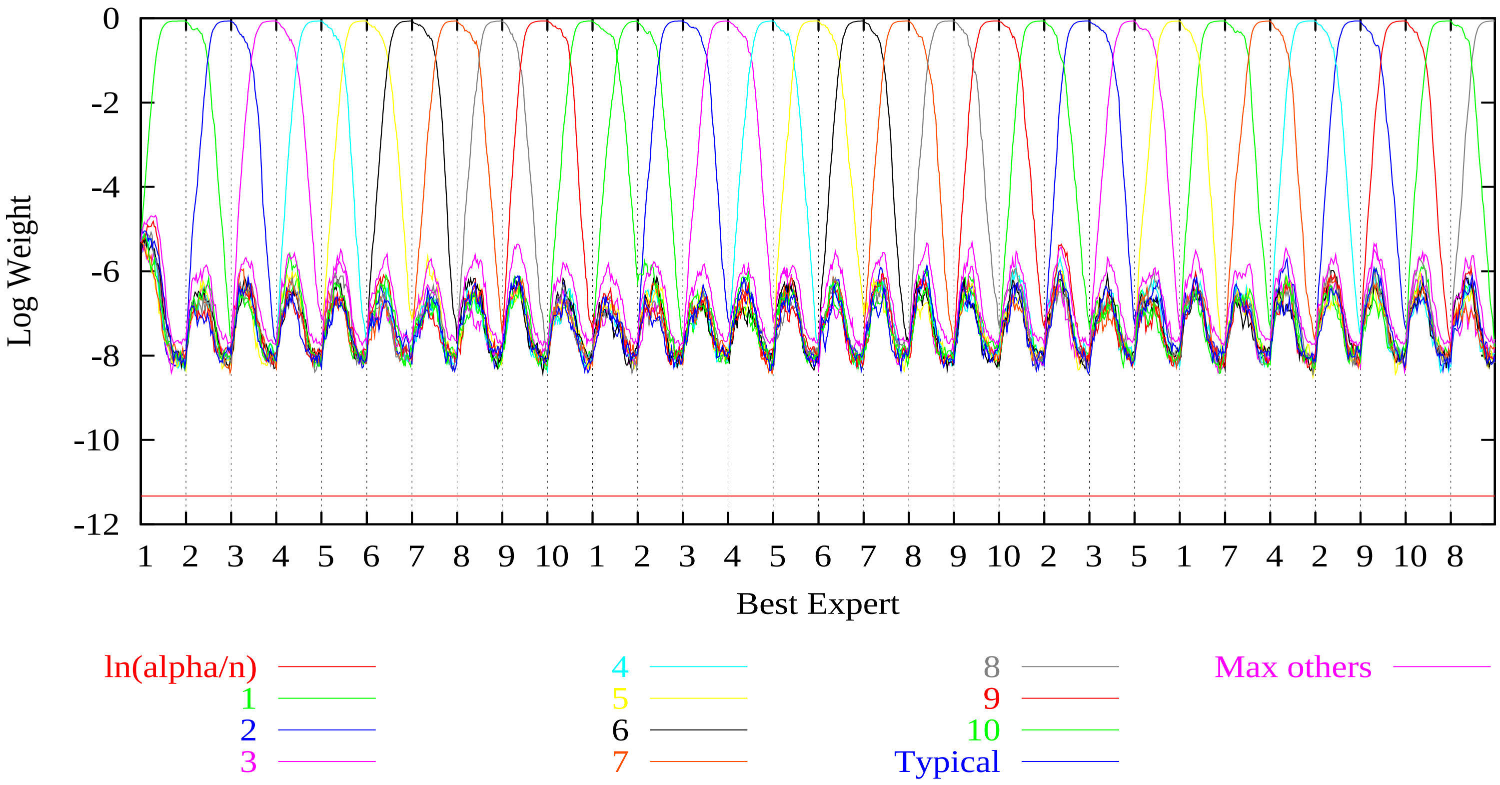
Fixed Share to Decaying Past - Log Weights

- Weights past good expert remain at higher level

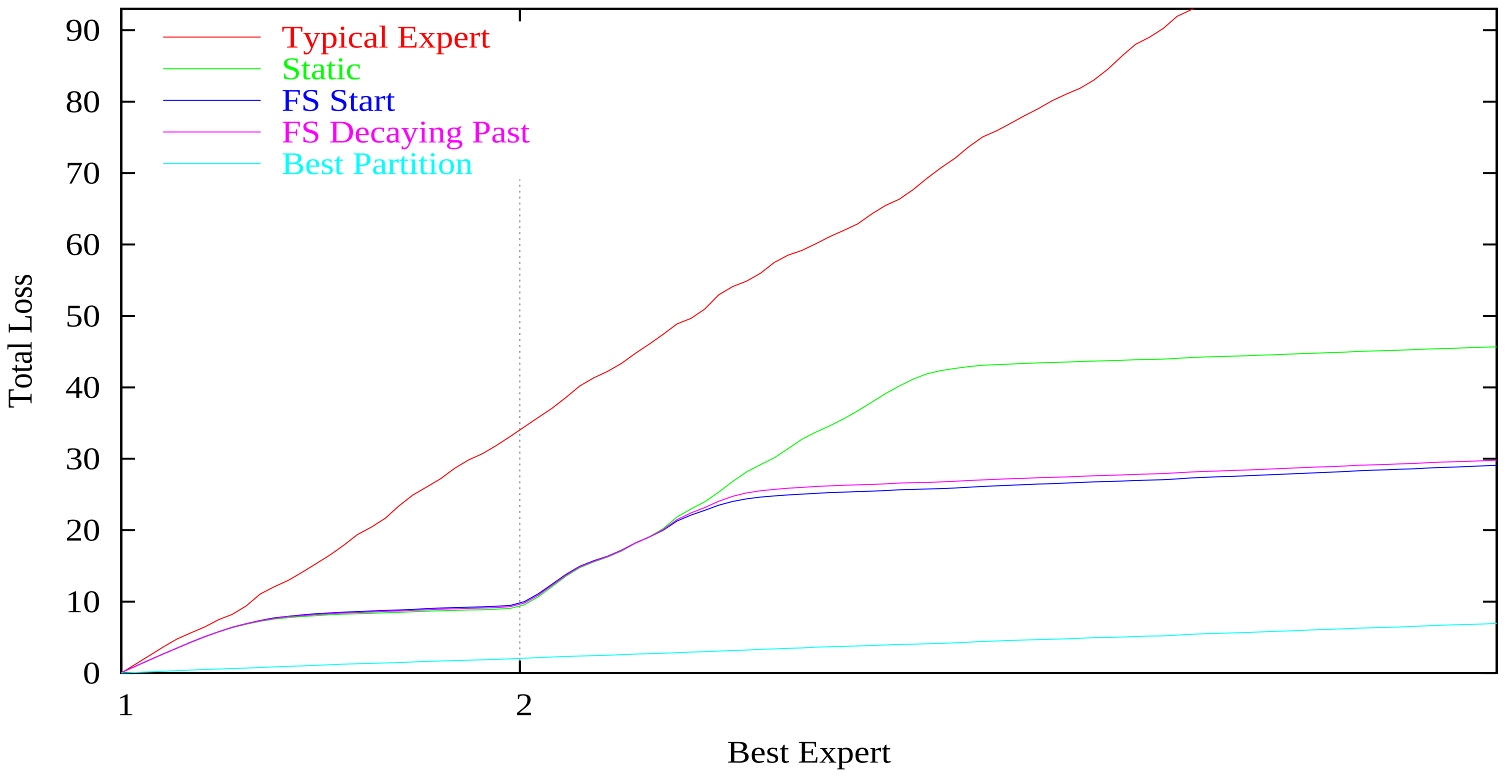


Fixed Share to Start Vector - Log Weights

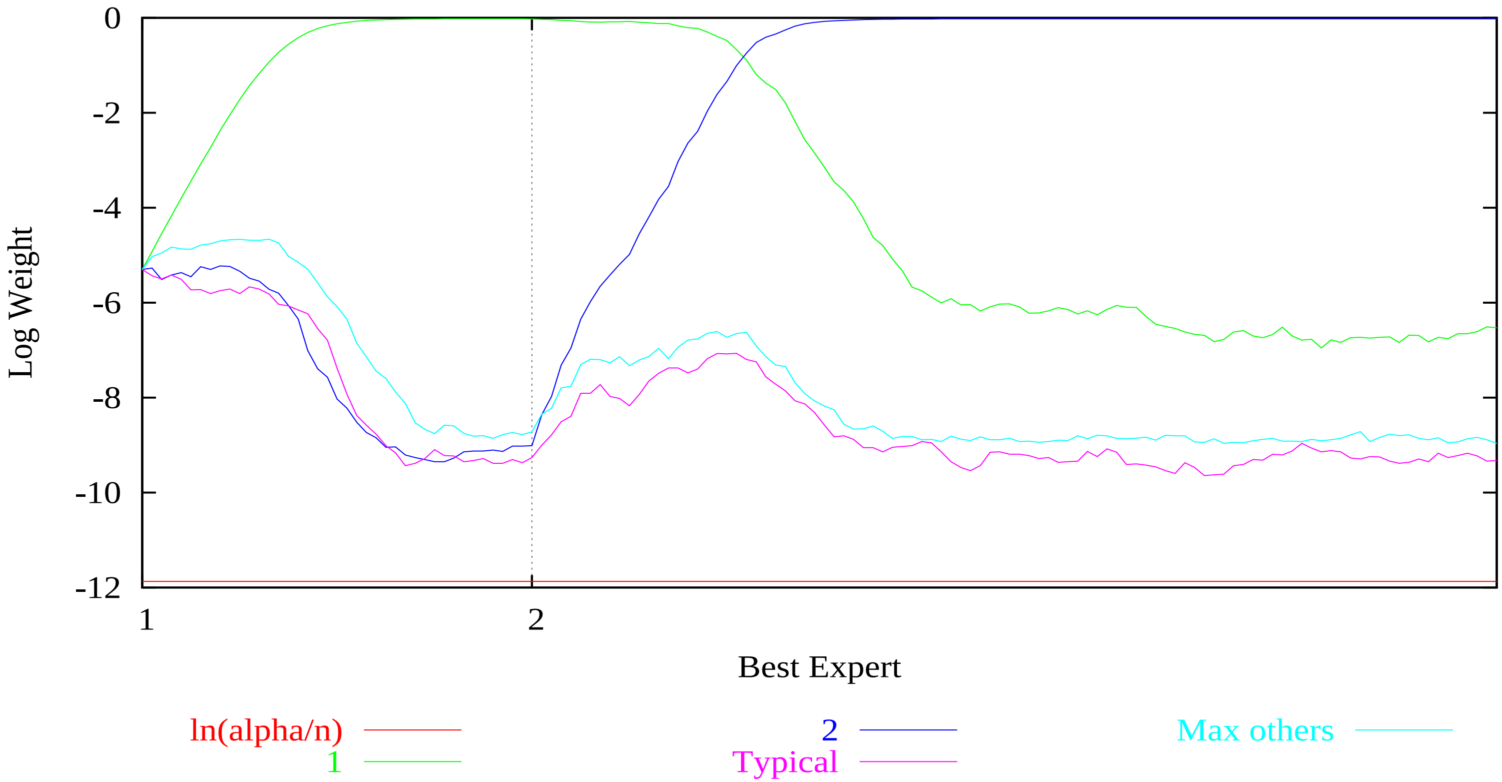
- No memory



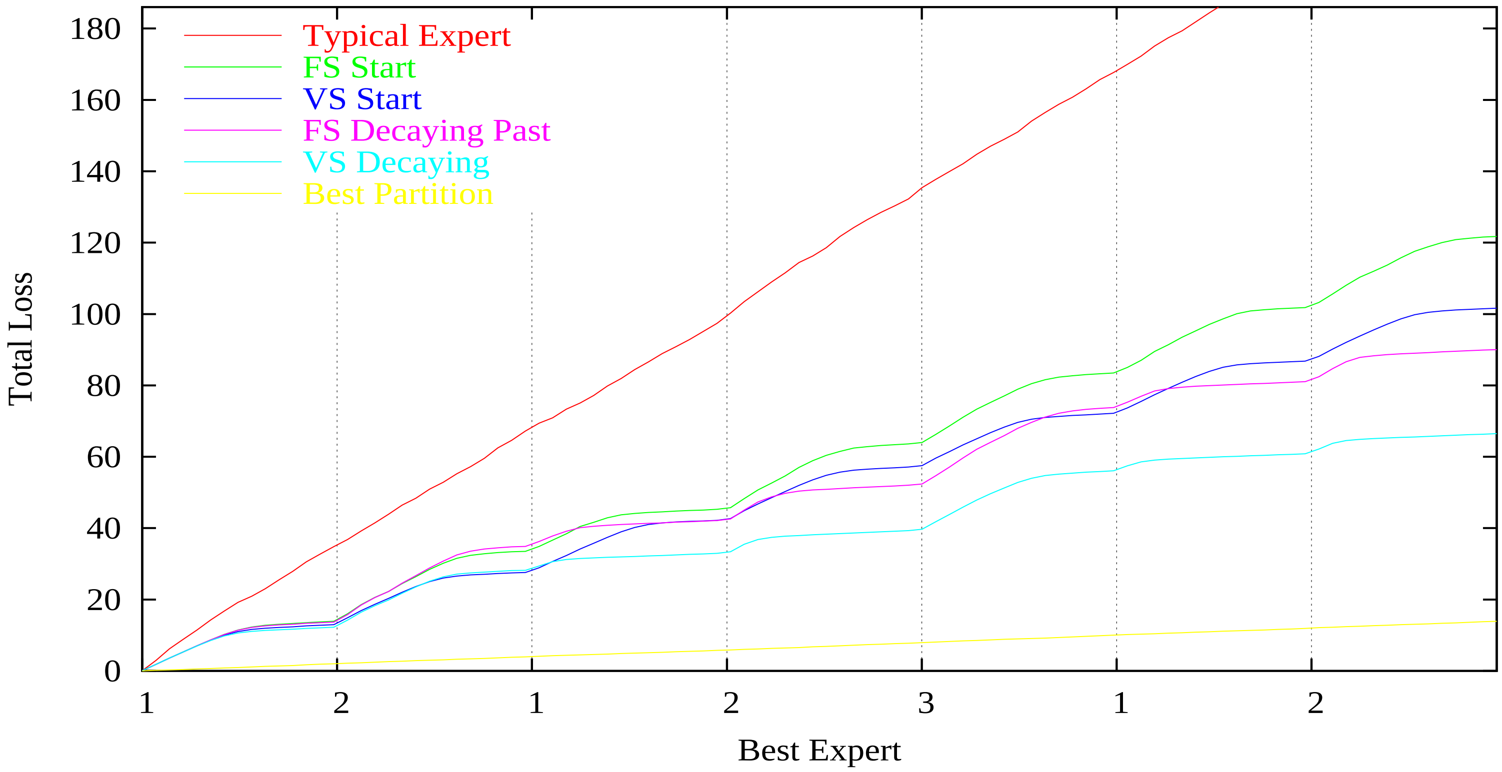
Long-term Versus Short-term Memory



Fixed Share to Decaying Past



Variable Share Beats Fixed Share



Bounds Again

- Bounds still have the form

$$L_{1..T,A} \leq \min_{P} \left(L_{1..T,P} + \ O(\#\text{ of bits for } P)\right)$$

- Excess loss for naive alg.

$$O(\log \binom{n}{m} + k \log m + \log \binom{T}{k})$$

- Excess loss for Fixed Share to Decaying Past

$$O\left(m \log n + k \log m + 2 \log \binom{T}{k}\right)$$

$\longrightarrow$ Boundaries are encoded twice

$\longrightarrow$ Off-line problem NP-complete

Loss Bounds Versus Storage Complexity

- Naive alg. has optimal bound - exponential storage

- Fixed Share to Uniform Past - $O(n)$ weights

- Fixed Share to Decaying Past - $O(nT)$ weights and better bound

$\longrightarrow$ With tricks $O(n \ln T)$ weights and essentially same bound

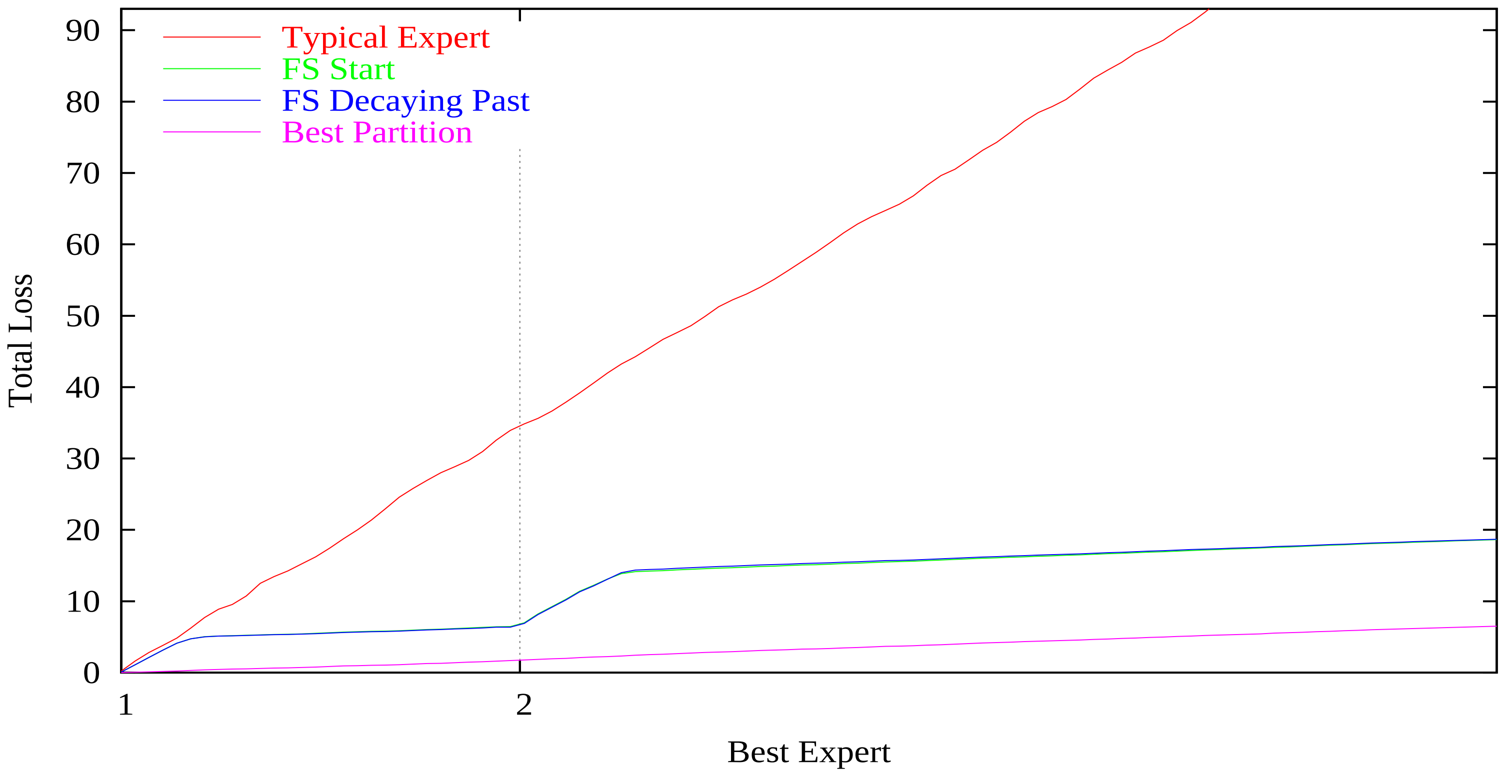
Alternates to Mixing

- What we need for bounds

$$\boldsymbol{v}_{t+1} = \beta_{t+1,q}\boldsymbol{v}_q^m, \quad \text{for } 0 \leq q \leq t \qquad (*)$$

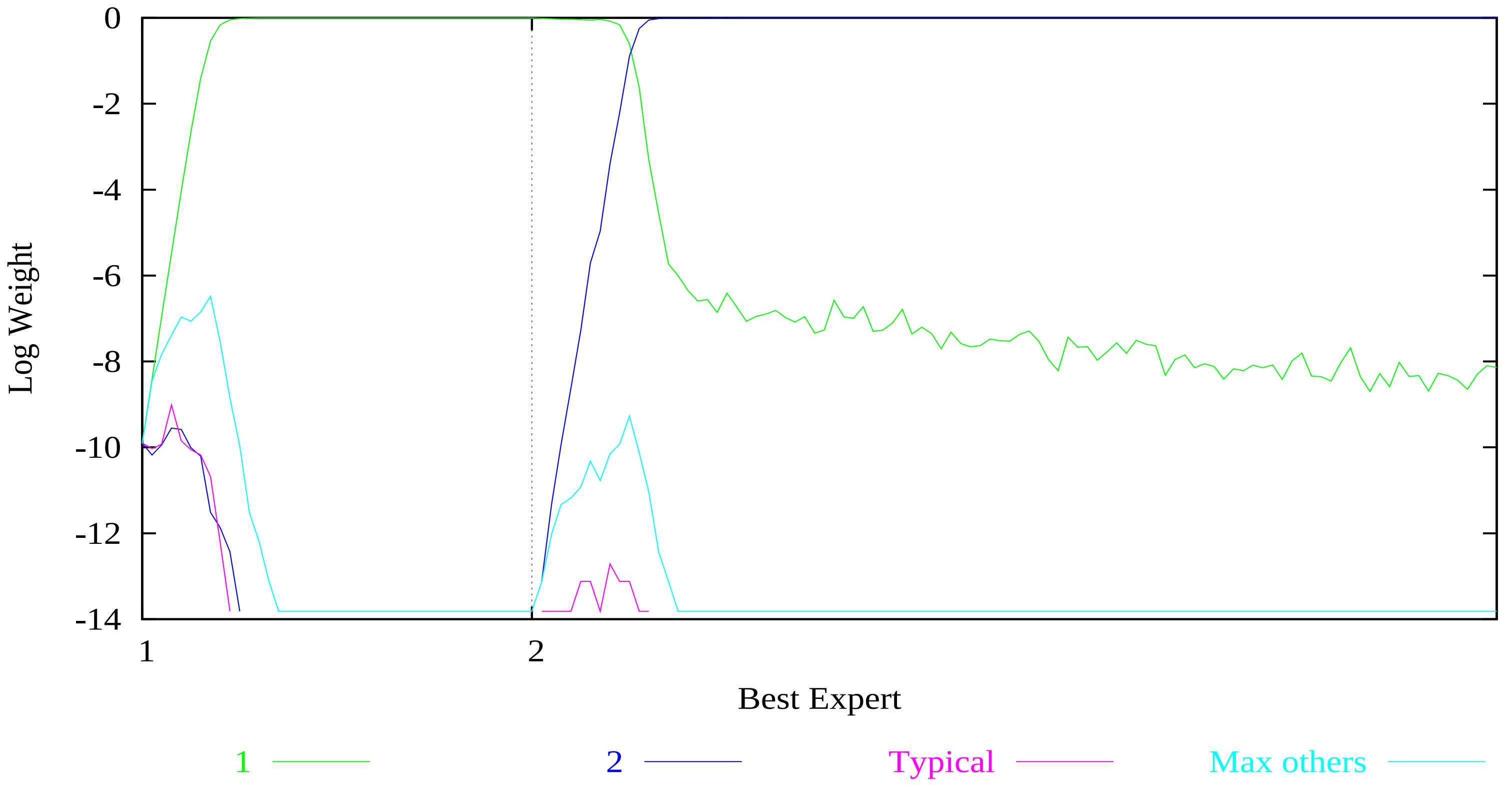
Mixing Update	$\boldsymbol{v}_{t+1} = \sum_{q=0}^{t} \beta_{t+1,q}\boldsymbol{v}_q^m$
Max Update	$\boldsymbol{v}_{t+1} = \frac{1}{\text{normaliz.}} \max_{q=0,\ldots,t} \beta_{t+1,q}\boldsymbol{v}_q^m$
Projection Update	$\boldsymbol{v}_{t+1} = \arg\min_{\boldsymbol{v}\in(*)} \Delta(\boldsymbol{v}, \boldsymbol{v}_t^m)$

Long-term Versus Short-term Memory



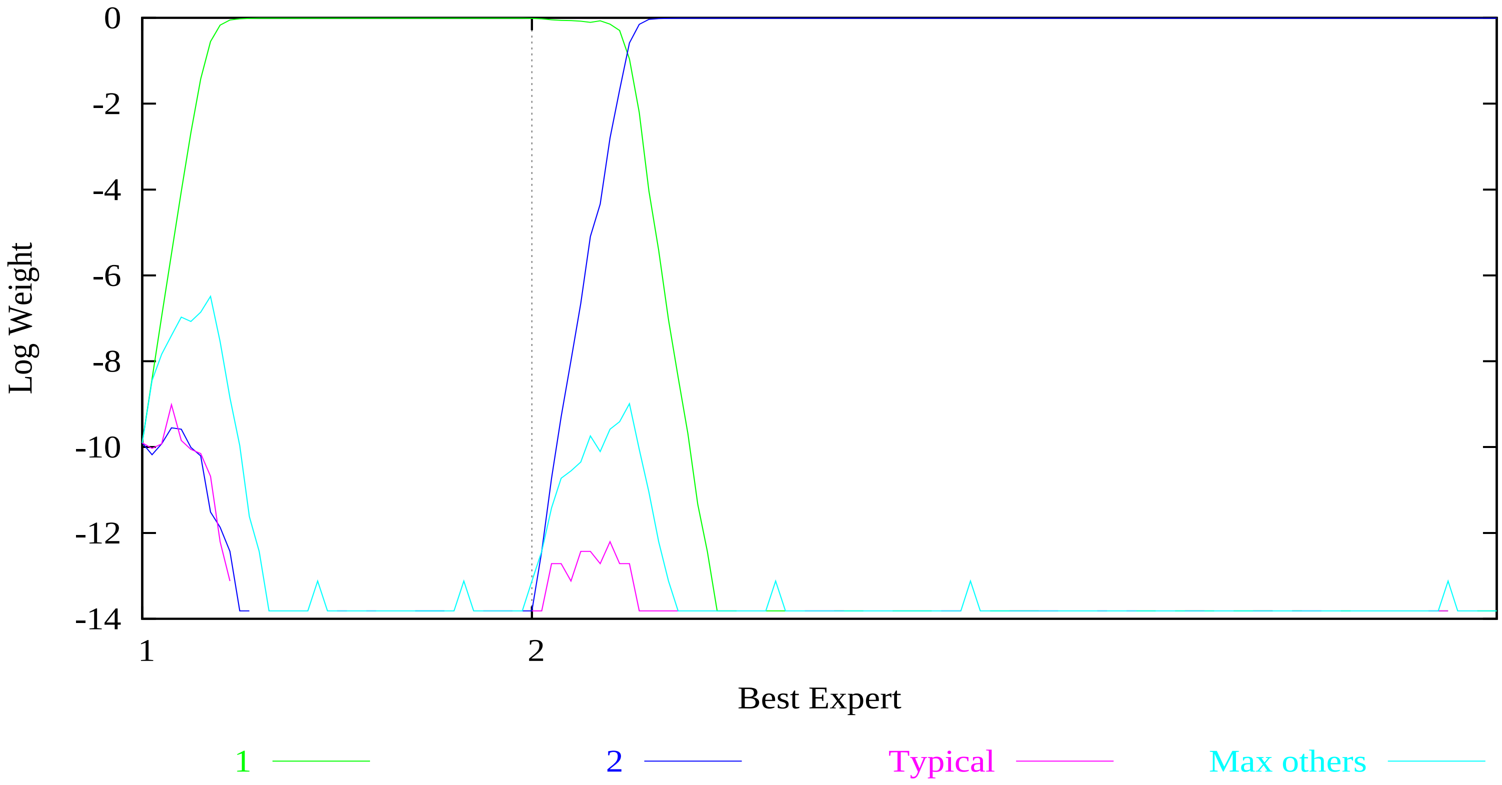
- $T = 1400$ trials, $n = 20000$ experts
- $k = 1$ shift (at trial 400), $m = 2$ experts in the small subset

Fixed Share to Decaying Past - Log Weights

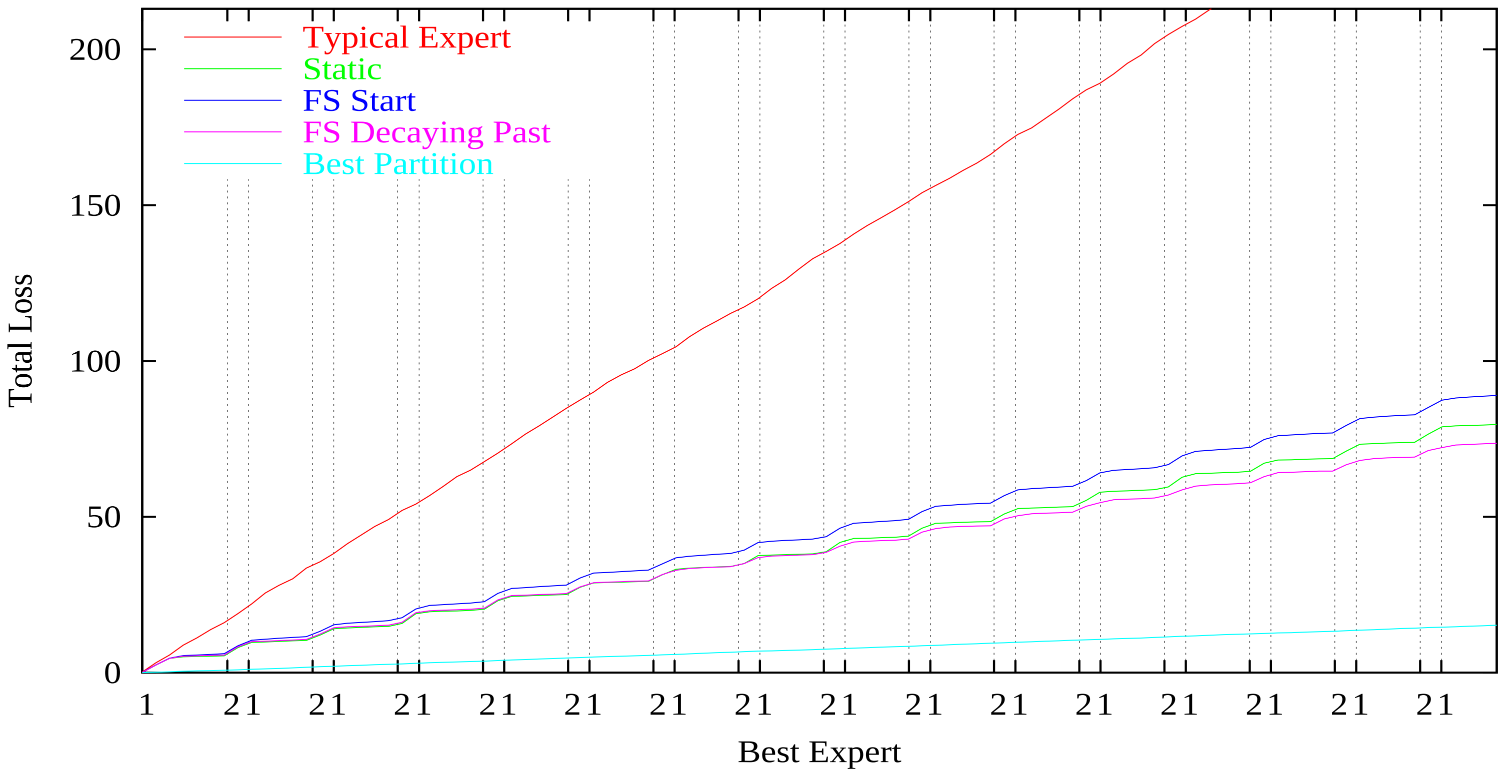


- Larger alpha gives better long-term memory

Fixed Share to Start Vector - Log Weights



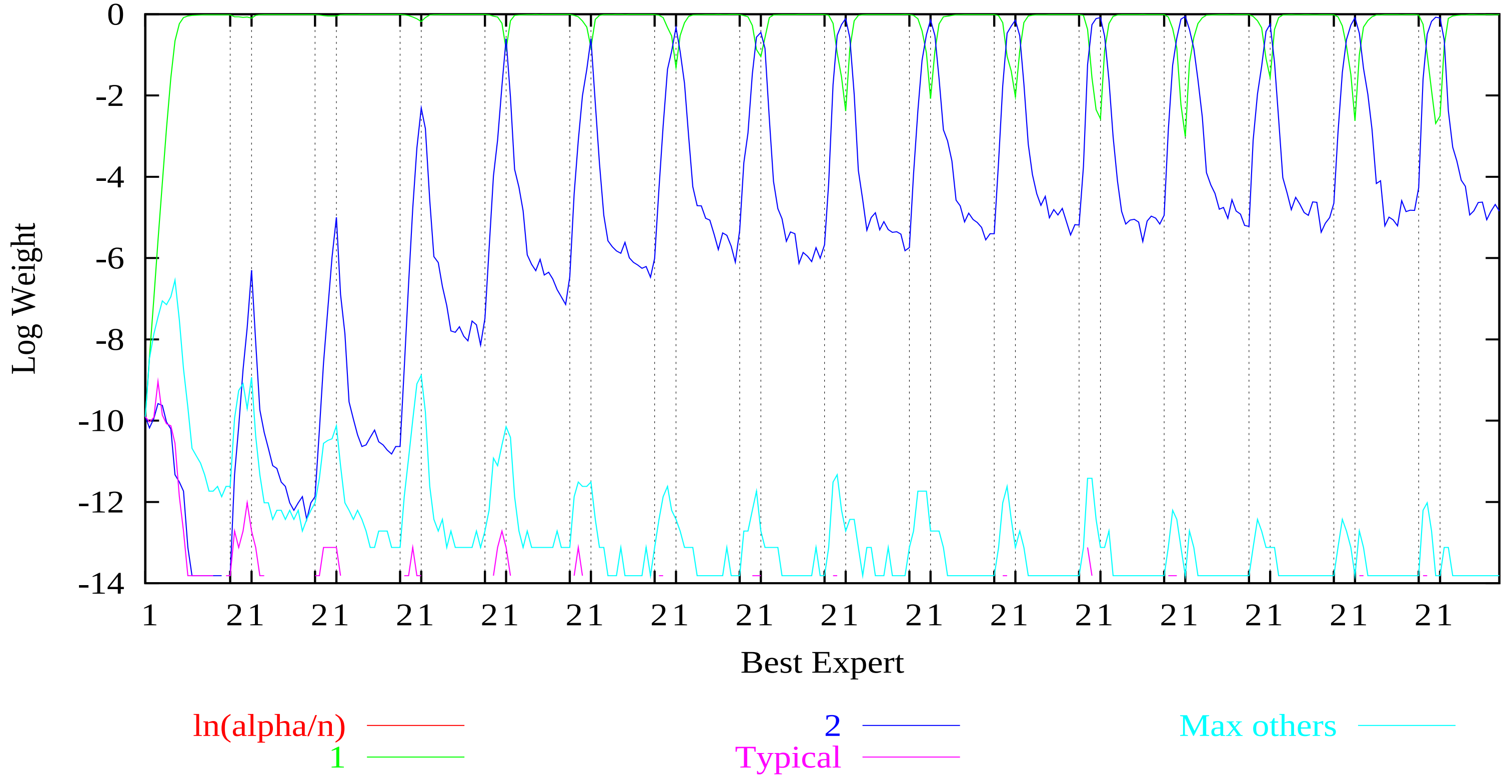
Many Short Sections of the Same Experts



- $T = 3200$ trials, $n = 20000$ experts
- $k = 30$ shifts (every 200 and 50 trials), $m = 2$ experts in the small subset

Fixed Share to Decaying Past - Log Weights

- The memory from many short sections accumulates



Future

- Bayesian interpretation
- Variable share
- Lower bounds
- Automatic tuning
- Mixing Update works for EG family
- Connections to Universal Coding
- Applications
  - Load balancing
  - Switching between a few users
  - Segmentation