Tracking a Small Set of Experts by Mixing Past Posteriors

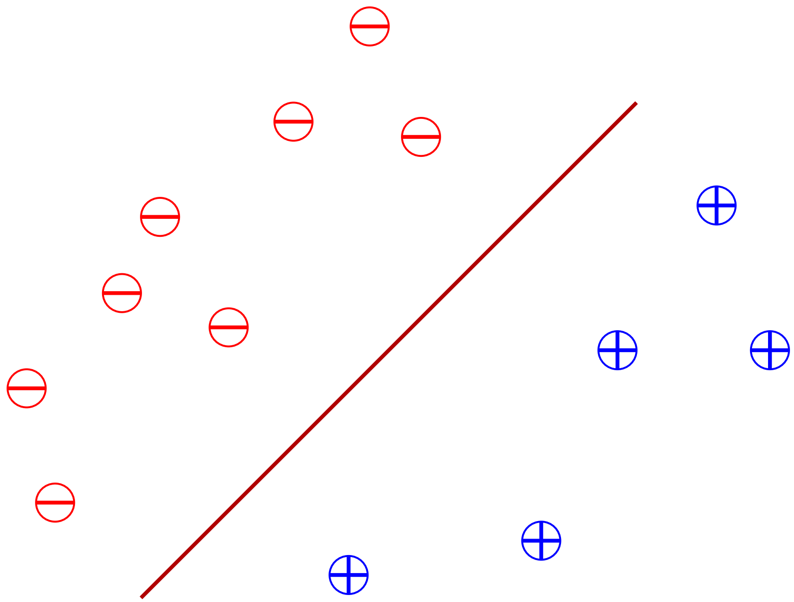
Olivier Bousquet
Ecole Polytechnique, France
and
BIOwulf Technologies, New-York

Manfred K. Warmuth
UC Santa Cruz

Outline

- Motivate on-line learning, relative loss bounds
- Comparator on-line as well
- Shifting back
- Mixing Update
- Experimental Results
- Future work

Common Batch Learning Paradigm



- Given batch of random examples
- Goal: Find discriminator that predicts well on random unseen examples

On-Line Learning

Loop for each trial $t = 1, .., T$

Get next instance	$\boldsymbol{x}_t$
Make prediction	$\hat{y}_t$
Get label	y_t ("true outcome")
Incur loss	$L(\hat{y}_t, y_t)$

- No statistical assumptions on the data
- Choose comparison class of predictors (experts)
  $\boldsymbol{x}_t = (x_{t,1}, x_{t,2}, x_{t,3}, \ldots, x_{t,n})$ vector of expert's predictions

Goal

- Do well compared to the best off-line comparator

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: for all sequence $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)$

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

  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

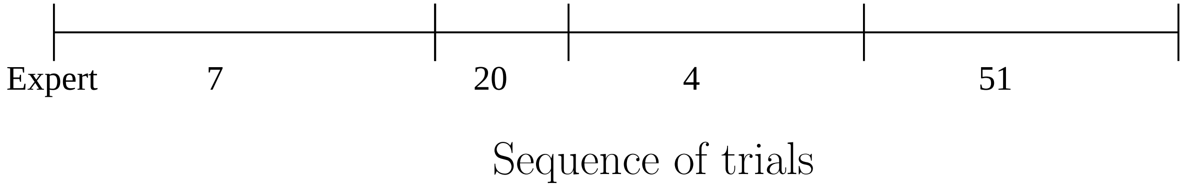
$$v_{t+1,i} := \frac{v_{t,i} \quad e^{-\eta \, L_{t,i}}}{\text{normaliz.}}$$

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

→ Weighted Majority Algorithm **[LW89]**

→ Generalized by Vovk **[Vovk90]**

Comparator Changes with Time



- Off-line algorithm partitions sequence into sections and chooses best expert in each section
- Goal:
  Do well compared to the 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

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

- Share Update
  - Fixed Share to Start Vector (small α)

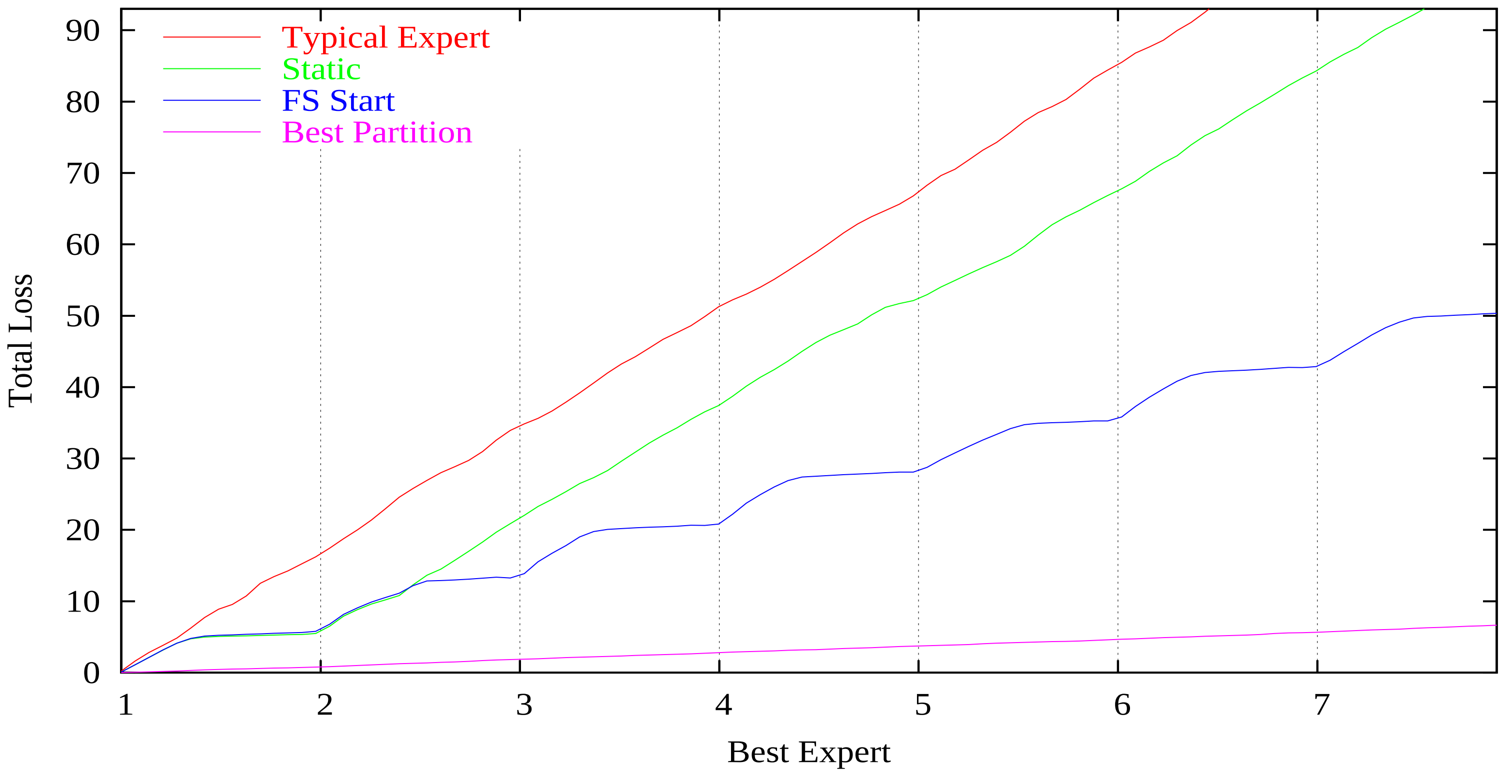
$$\boldsymbol{v}_{t+1} = (1-\alpha)\boldsymbol{v}^m_t + \alpha \boldsymbol{v}_0$$

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

  - Static Expert: corresponds to $\alpha = 0$

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

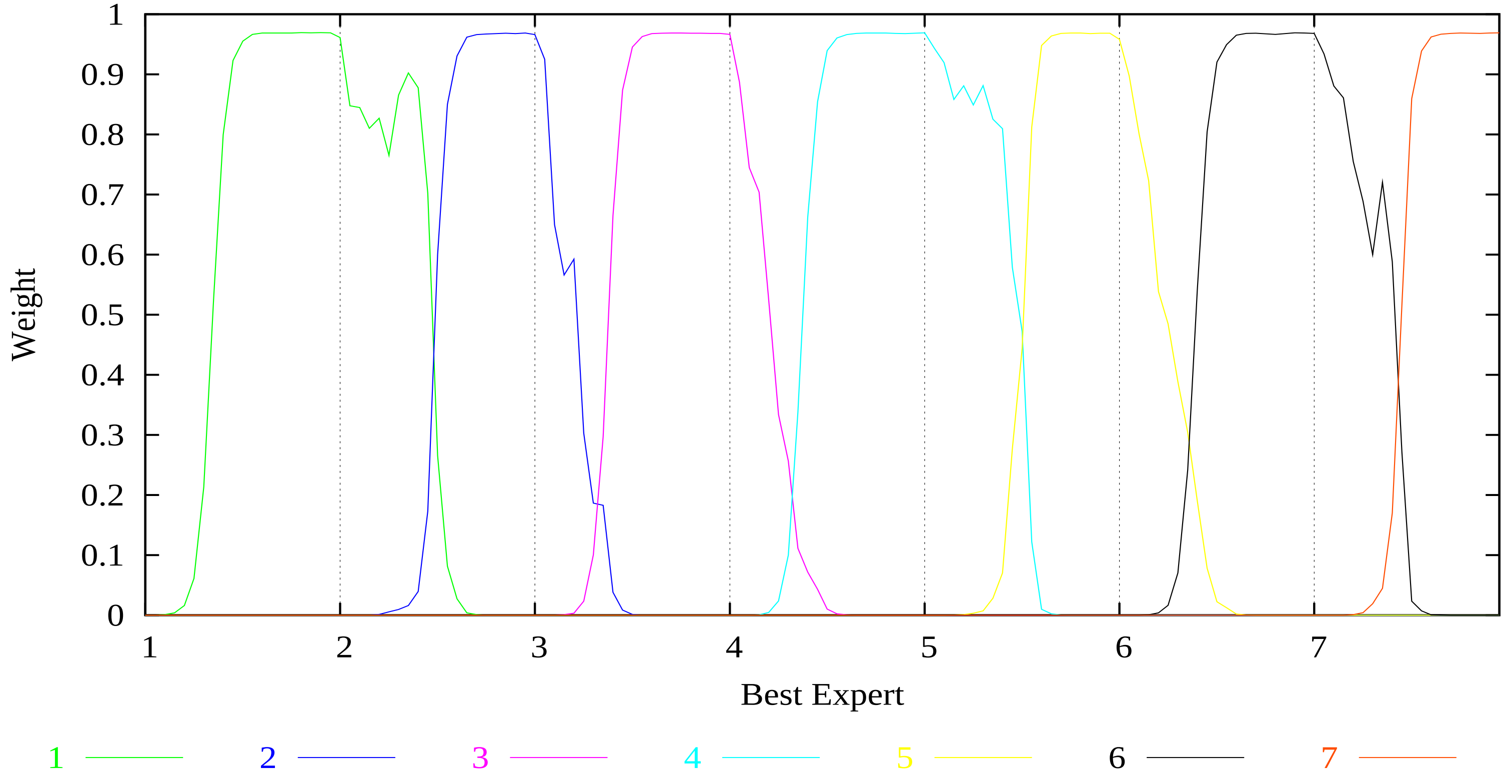
Static Expert Alg. Versus Share Alg.



- Square loss, Vovk's prediction, labels always 0, typical experts predict uniform in $[0, .5]$, current best expert predicts uniform in $[0, .12]$
- $T = 1400$ trials, $n = 20000$ experts, $k = 6$ shifts

Weights of Fixed Share Alg.

- Tracks the best expert



Shifting Bounds

- Recall Static Expert bound

$$L_{1..T,A} \leq \min_i \left(L_{1..T,i} + O(\log n)\right)$$

  - Comparison class: set of experts

- Bounds for Share Algorithms **[HW98]**

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

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

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

Freund's Problem

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

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

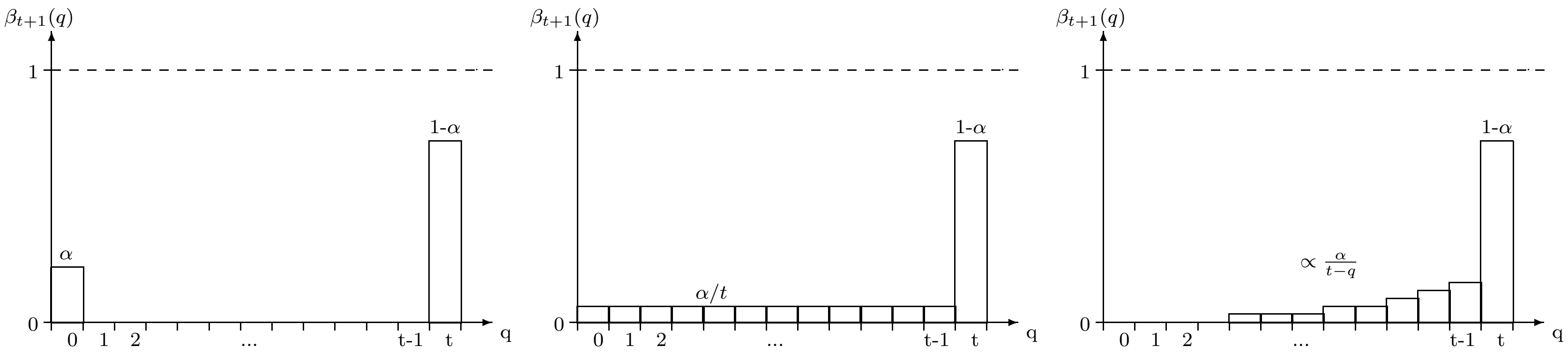
- Naive algorithm runs Fixed Share to Start Vector alg. for every subset of m out of n experts

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

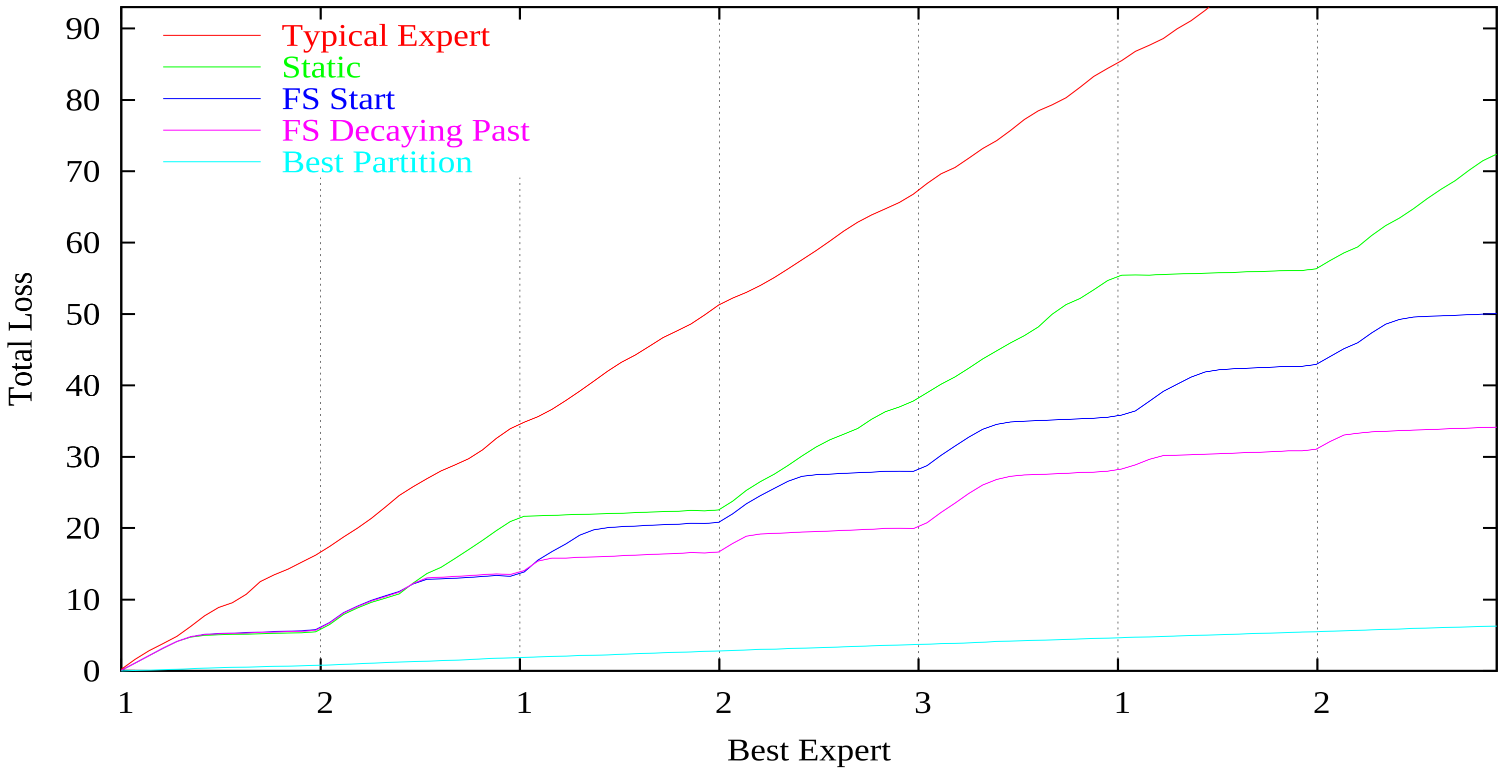
$$\boldsymbol{v}_{t+1} = \sum_{q=0}^{t} \beta_{t+1,q} \boldsymbol{v}^m_q, \quad \text{where} \sum_{q=0}^{t} \beta_{t+1} = 1$$

- Mixing schemes



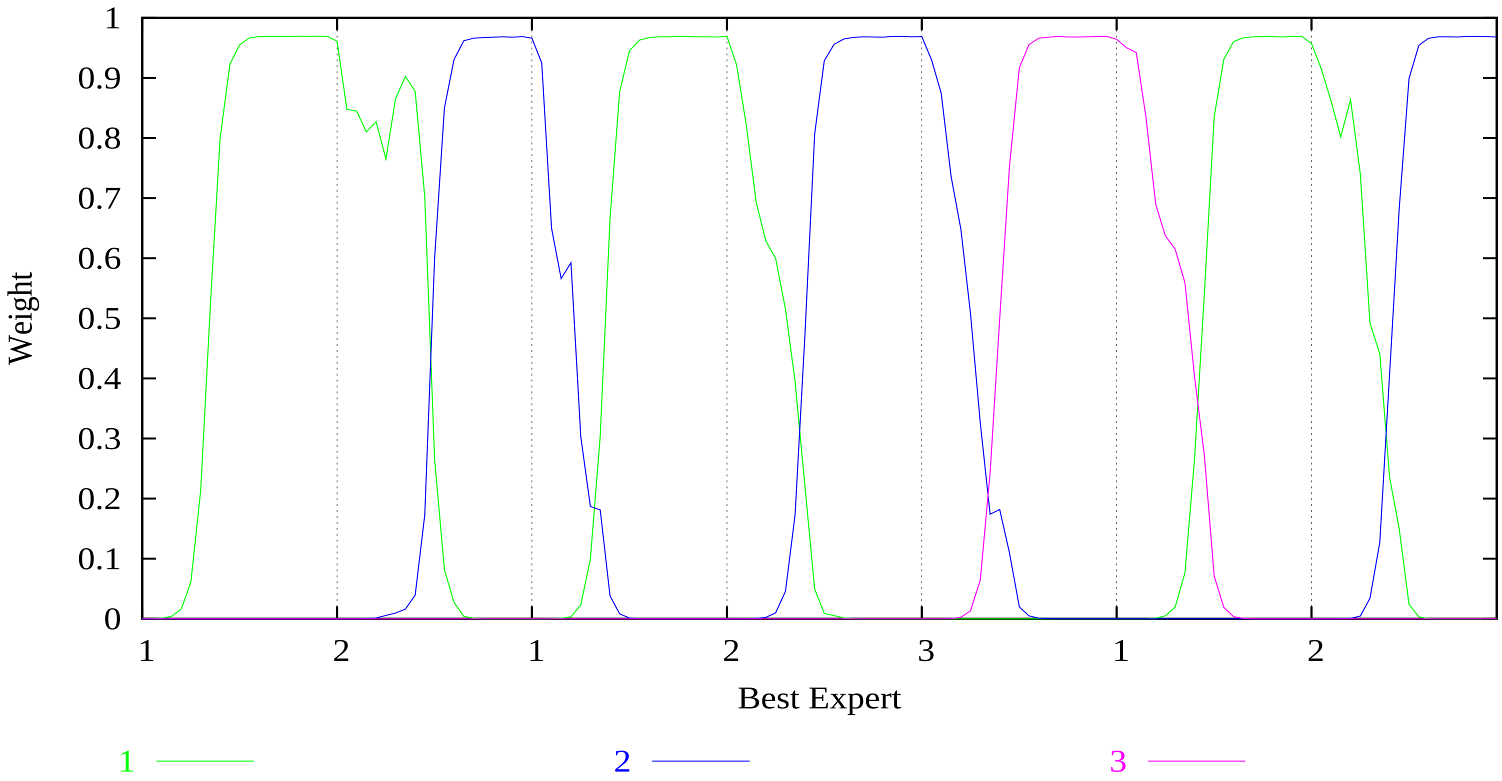
FS to Start Vector　　FS to Uniform Past　　FS to Decaying Past

Total Loss Plots



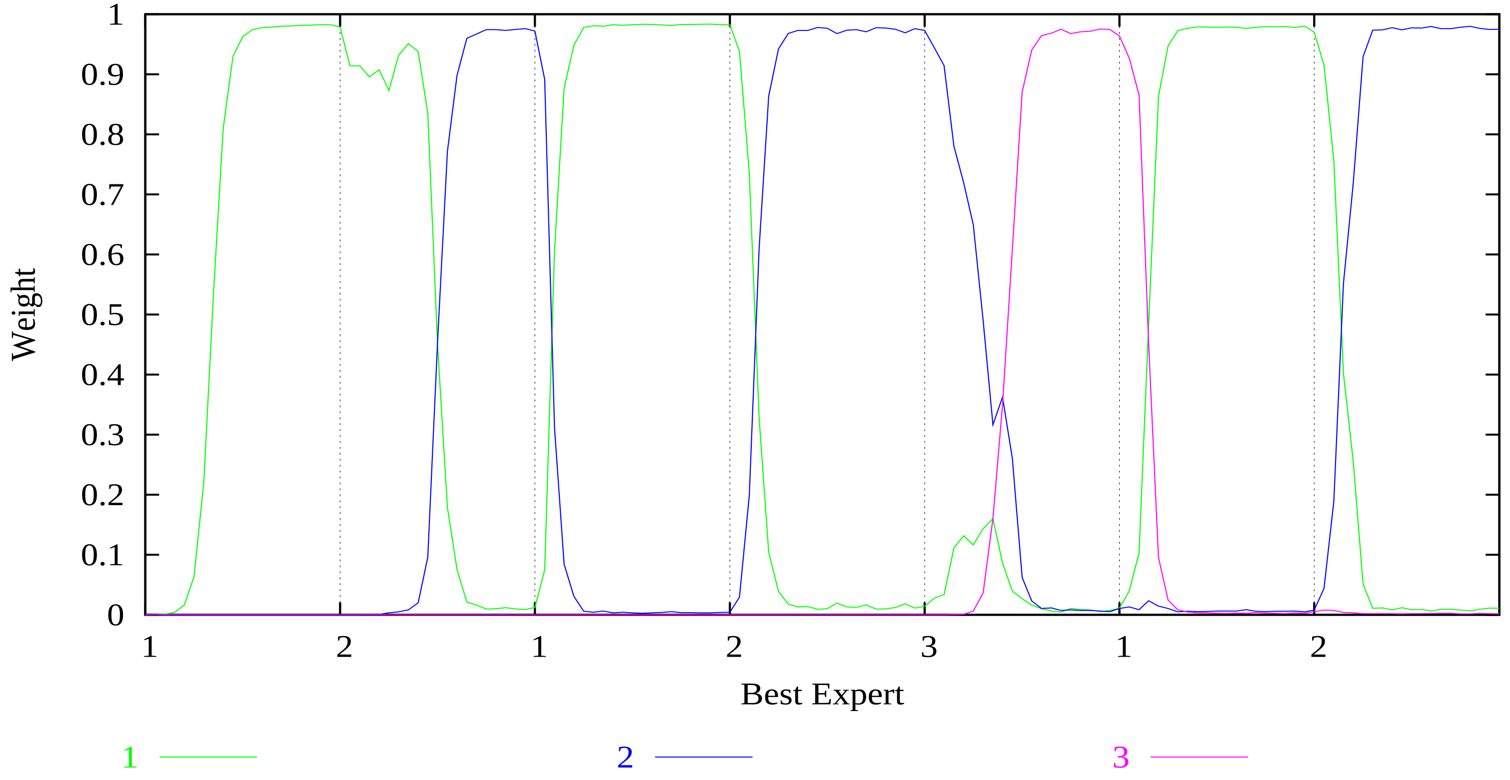
- $T = 1400$ trials, $n = 20000$ experts
- $k = 6$ shifts (every 200 trials), $m = 3$ experts in the small subset

Weights of Fixed Share to Start Vector Alg.



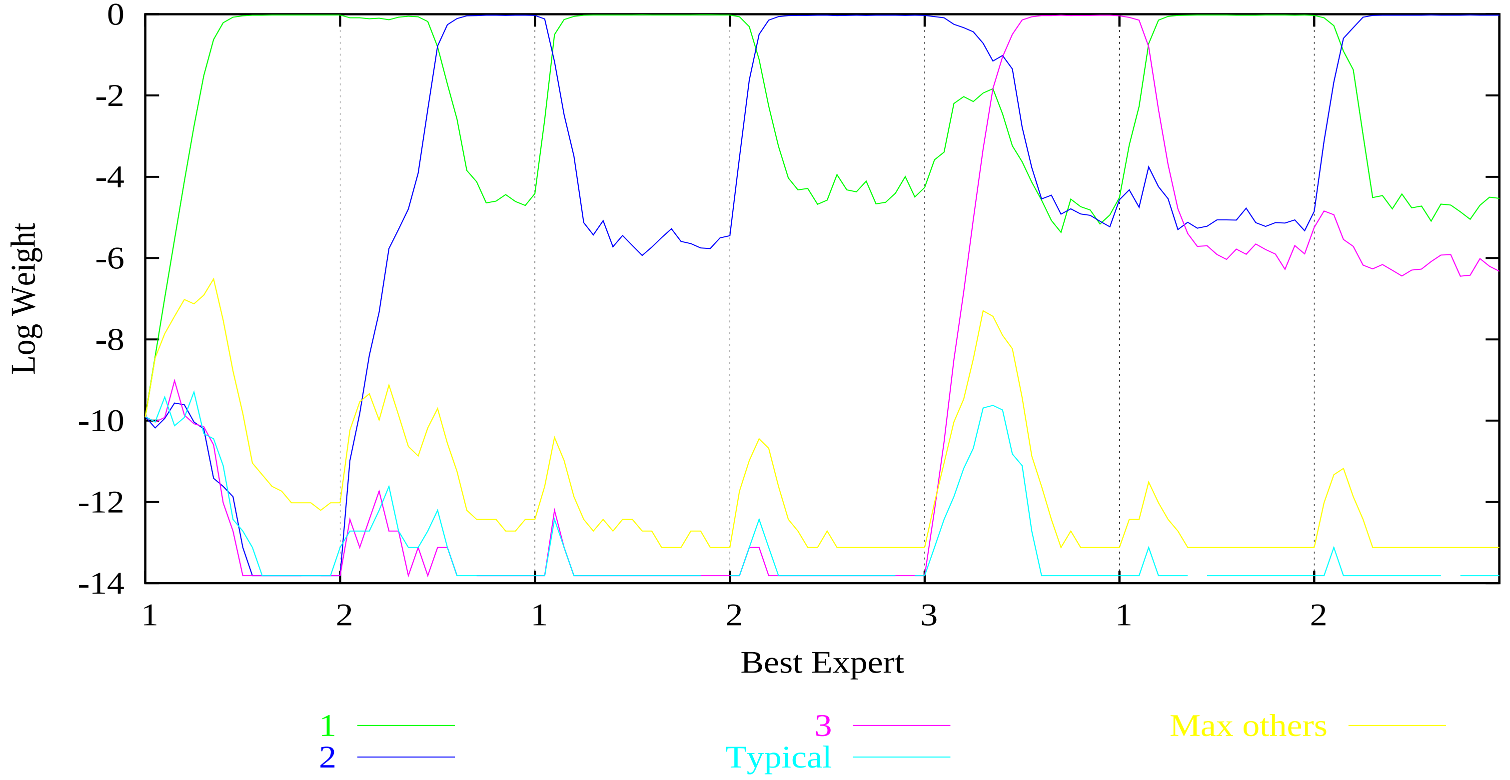
Weights of Fixed Share to Decaying Past Alg.

- Improved recovery when expert used before

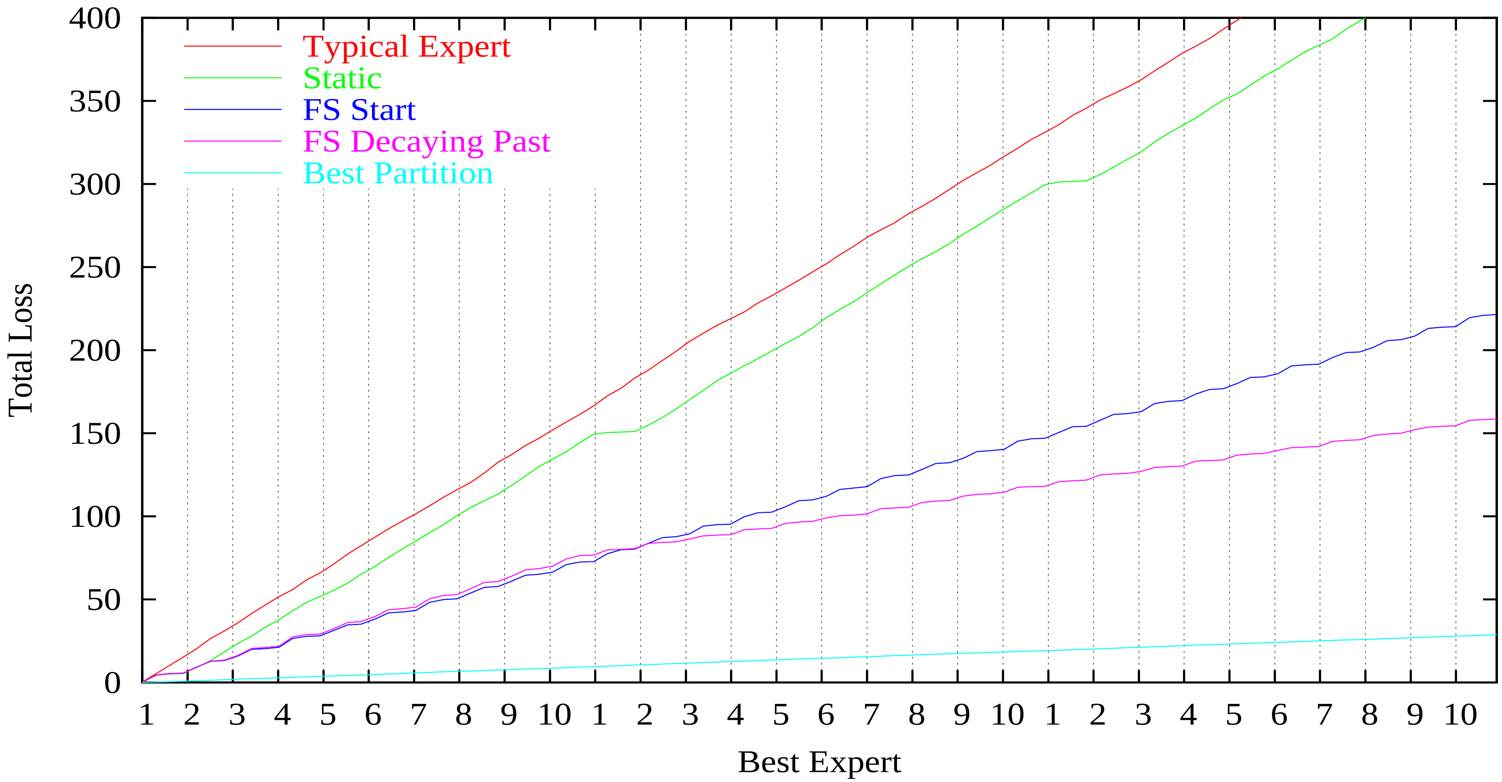


Fixed Share to Decaying Past - Log Weights

- Past good experts remain at higher level



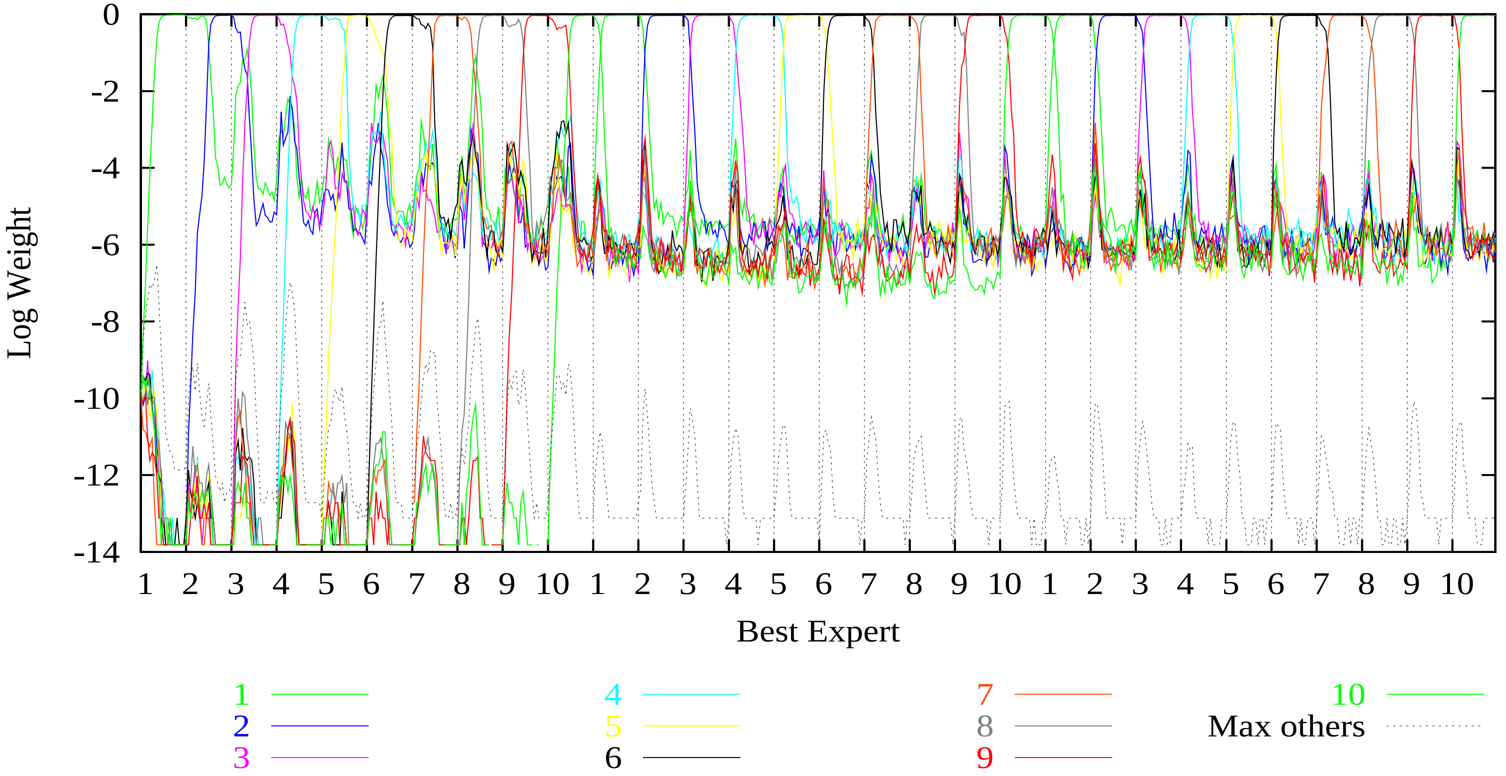
More Experts Remembered



- $T = 6000$ trials, $n = 20000$ experts
- $k = 29$ shifts (every 200 trials), $m = 10$ experts in the small subset

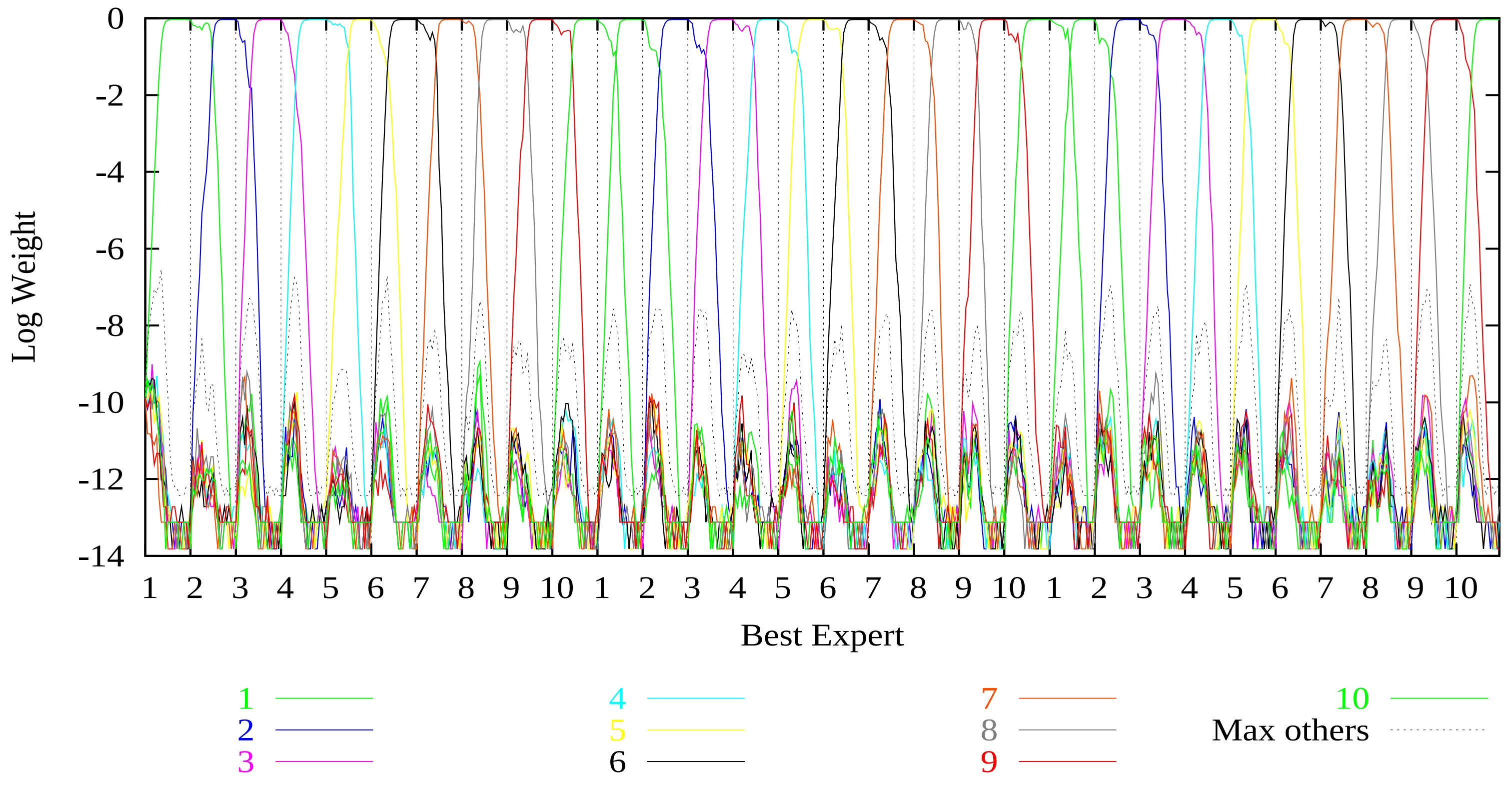
Fixed Share to Decaying Past - Log Weights

- Past good expert are cached



Fixed Share to Start Vector - Log Weights

- No memory



Bounds Again

- Bounds still have the form

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

$\longrightarrow$ Boundaries are encoded twice

$\longrightarrow$ Off-line problem NP-complete

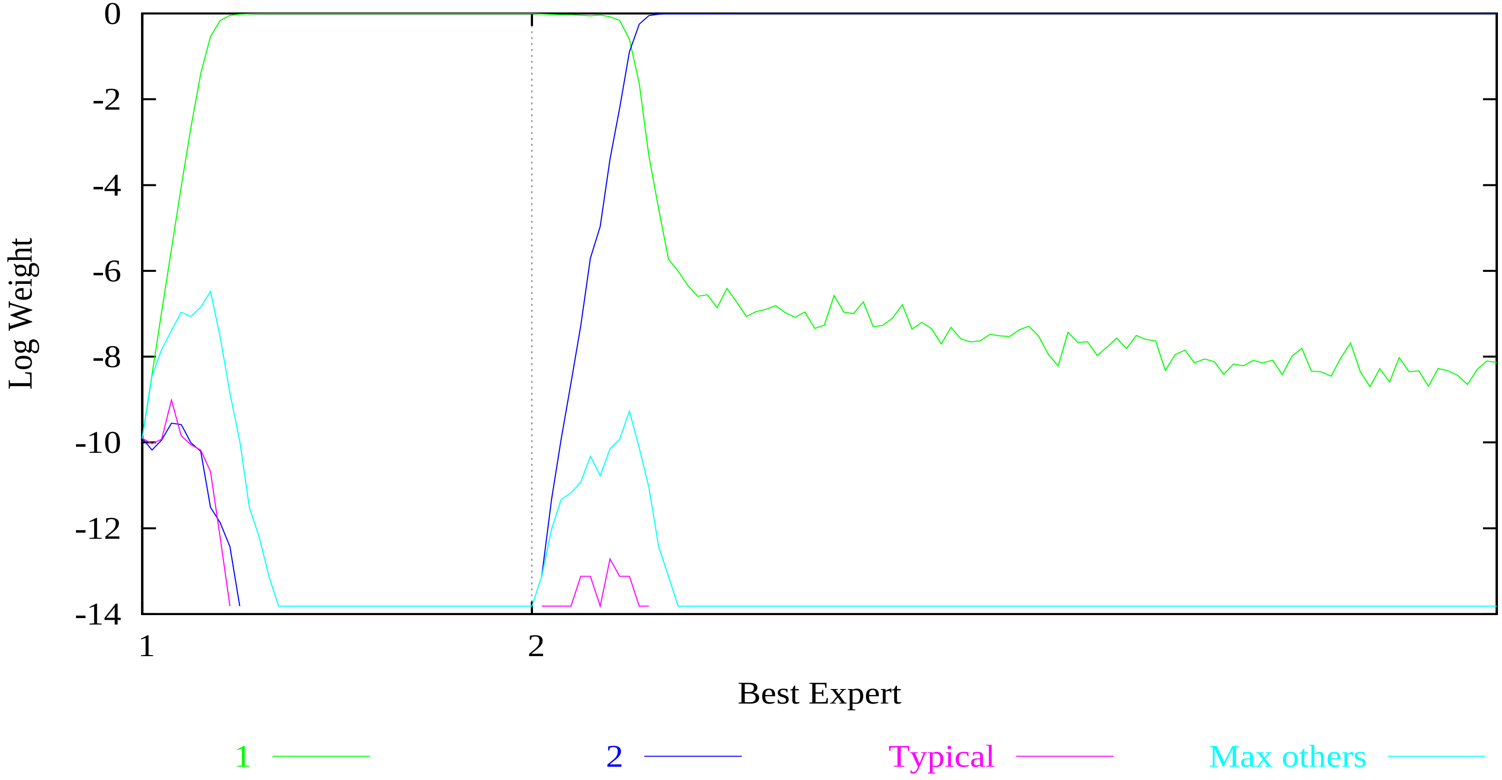
Alternates to Mixing

- What we need for bounds

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

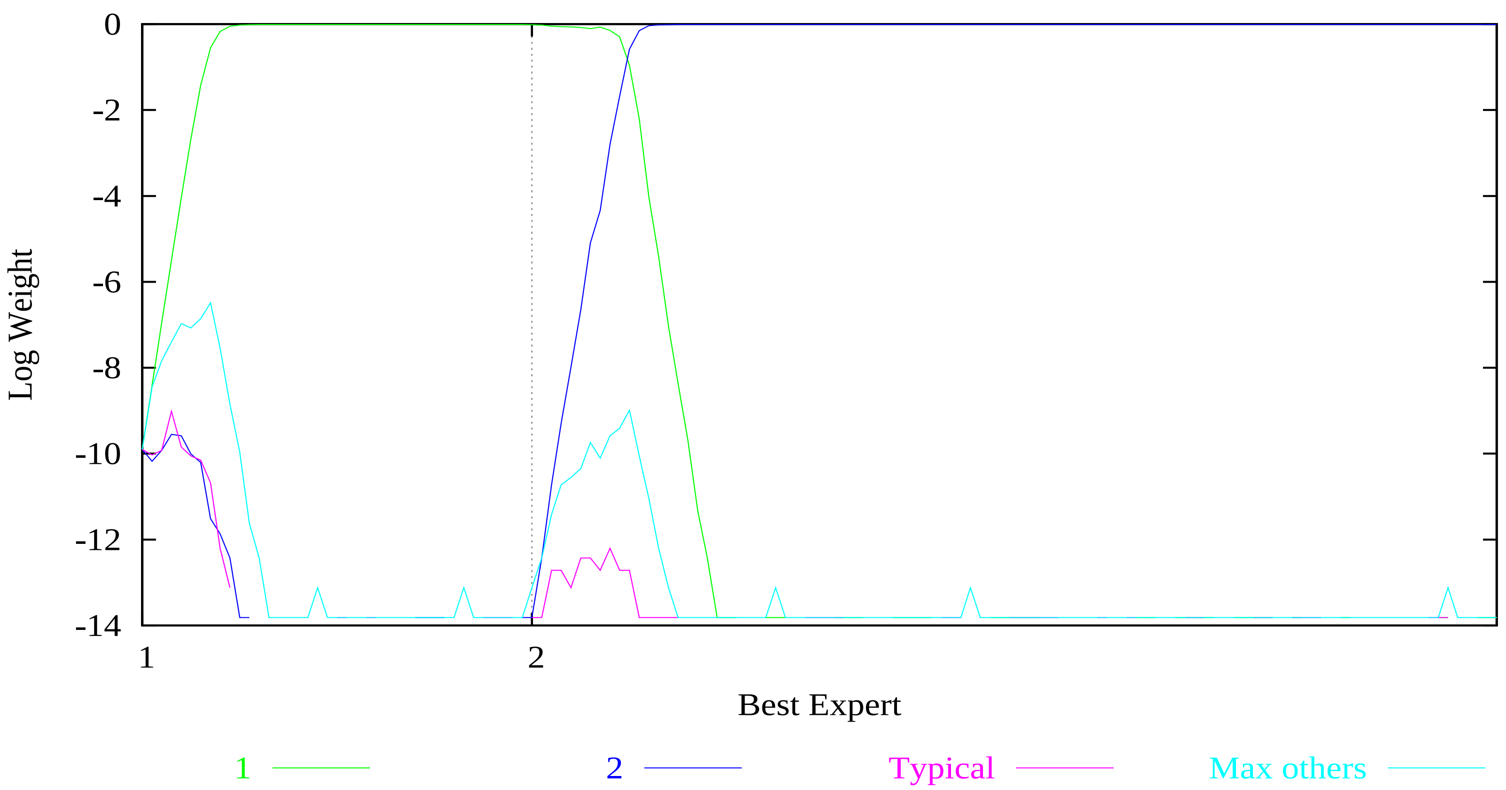
Mixing Update	$\boldsymbol{v}_{t+1} = \sum_{q=0}^{t} \beta_{t+1,q}\boldsymbol{v}_q^m$
Max Update	$\boldsymbol{v}_{t+1} = \frac{1}{\text{normaliz.}} \max_{q=0,\ldots,t} \beta_{t+1,q}\boldsymbol{v}_q^m$
Projection Update	$\boldsymbol{v}_{t+1} = \arg\min_{\boldsymbol{v}\in(*)} \Delta(\boldsymbol{v}, \boldsymbol{v}_t^m)$

Short Term Versus Long Term Memory
Fixed Share to Decaying Past

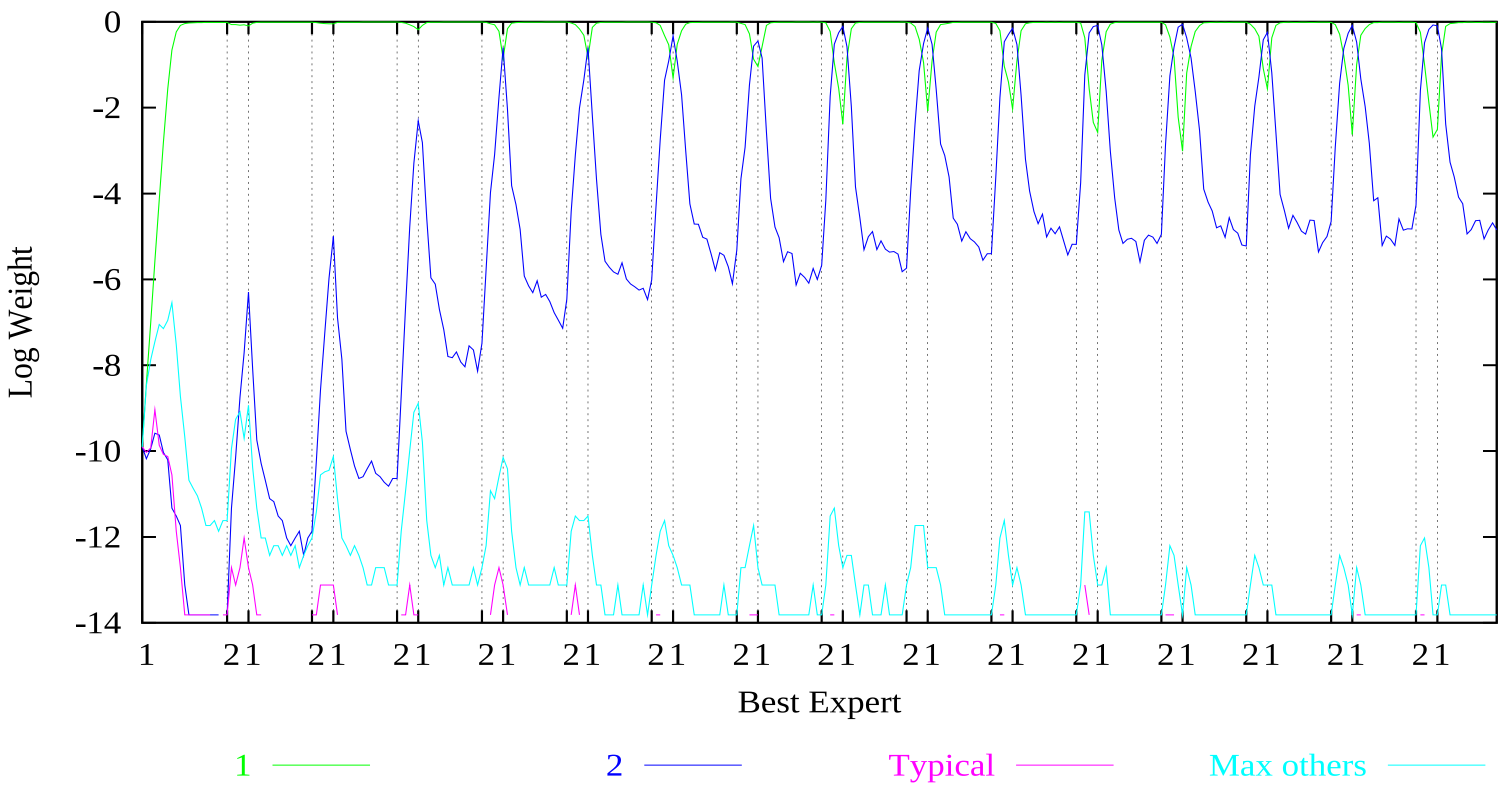


- Larger alpha gives better long-term memory

Fixed Share to Start Vector

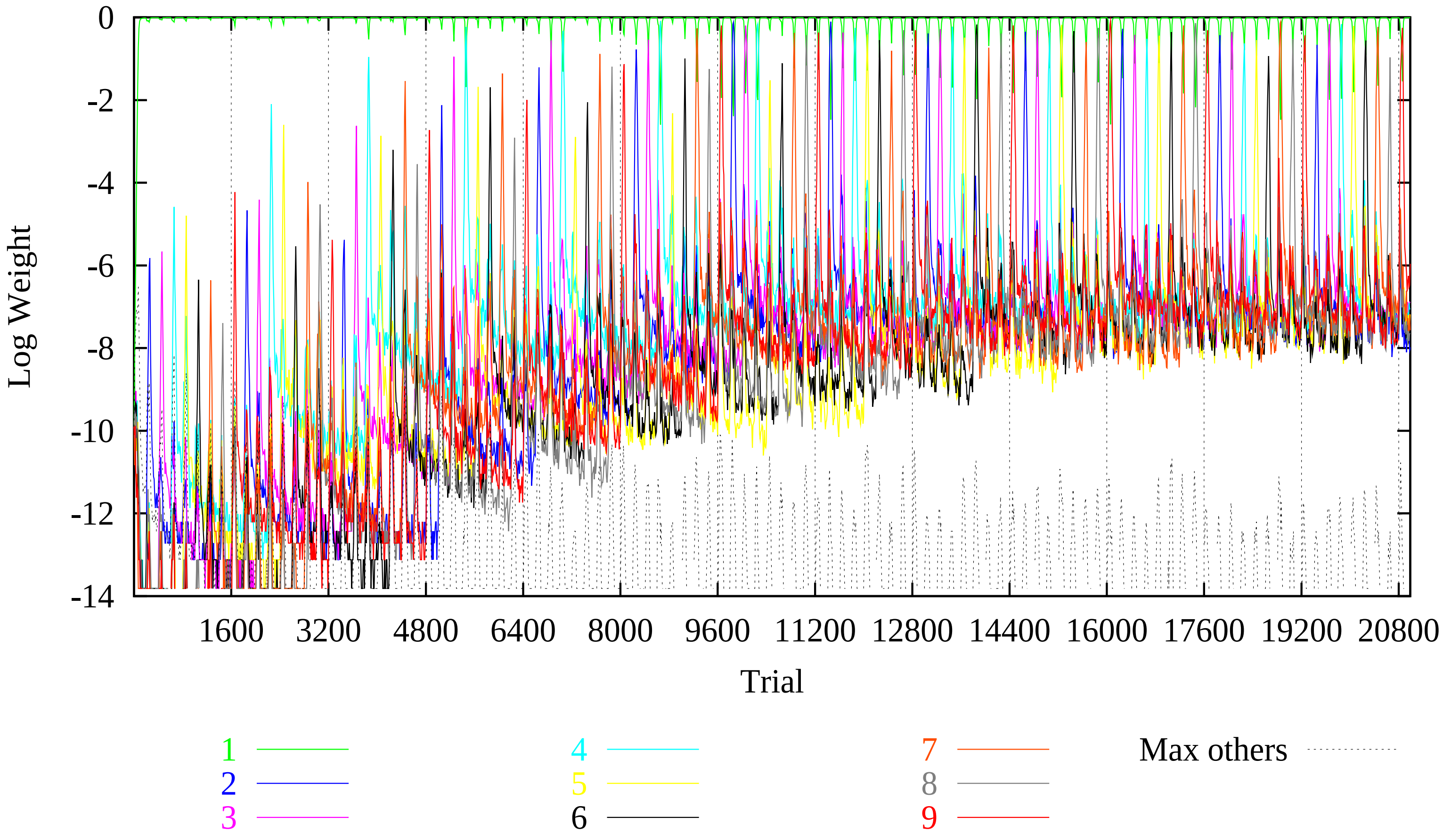


Many Short Sections of the Same Experts



- Long sections 140 trials, short sections 60 trials
- $T = 3200$ trials, $n = 20000$ experts, $k = 30$ shifts, $m = 2$ experts in the small subset

Parallel Ratcheting



Future

- Bayesian interpretation
- Variable share [HW98]
- Lower bounds
- Automatic tuning
- Mixing Update works for EG family
- Connections to Universal Coding
- Applications
  - Load balancing
  - Switching between a few users
  - Segmentation

Thanks to

Yoav and Mark

: -)