Online Learning of Non-stationary Sequences

Claire Monteleoni
MIT CSAIL
cmontel@csail.mit.edu

Joint work with Tommi Jaakkola

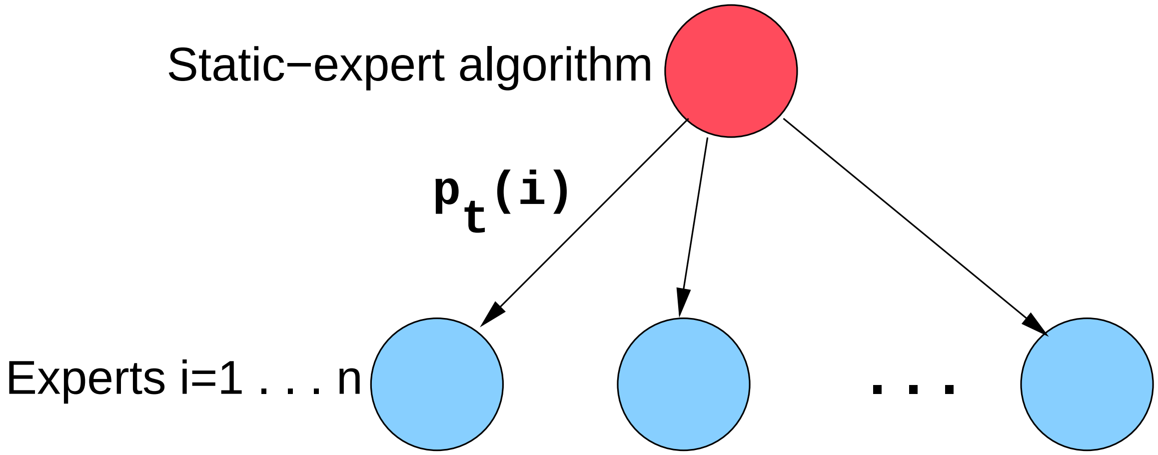
Outline

- Online learning framework
- Upper and lower regret bounds for a class of online learning algorithms
- An algorithm that simultaneously learns the switching-rate, α, at optimal discretization
- A stronger bound on regret of new algorithm
- Application to wireless networks

Online Learning Framework

- Typical set-up: receive one (x_t, y_t) example at a time
  - view x_t first, to test current predictions
  - regression, estimation or classification
- No statistical assumptions about observations
  - no stationarity assumptions on generating process
  - labels could even be adversarial
- Learner makes prediction on each example, and receives associated prediction loss.
  - loss on all examples counts – no separate "training" period.

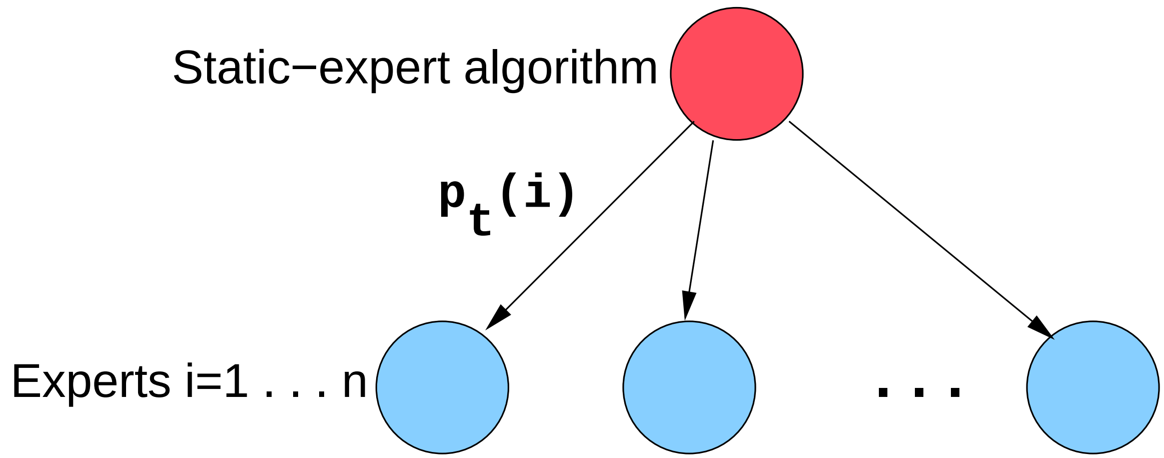
Online Learning Framework



- Algorithm[1] bases prediction on a set of n experts.
  - in this framework, x_t is the vector of experts' predictions
  - experts' prediction mechanisms unknown, can vary
    - over time
    - over experts
  - algorithm maintains a distribution over the experts $p_t(i)$.

[1] Static-expert due to [Littlestone and Warmuth, 1989]

Online Learning Framework



- $L(i,t)$ is non-negative prediction loss of expert i at time t (depends on the true label $y_t \in \mathcal{Y}$).

- Bayesian updates are $p_{t+1}(i) \propto p_t(i)\, e^{-L(i,t)}$.

- $L(p_t,t)$ is loss of the algorithm.

- Objective: bound prediction loss to that of best expert, or best sequence of experts, over finite, known, time horizon T.

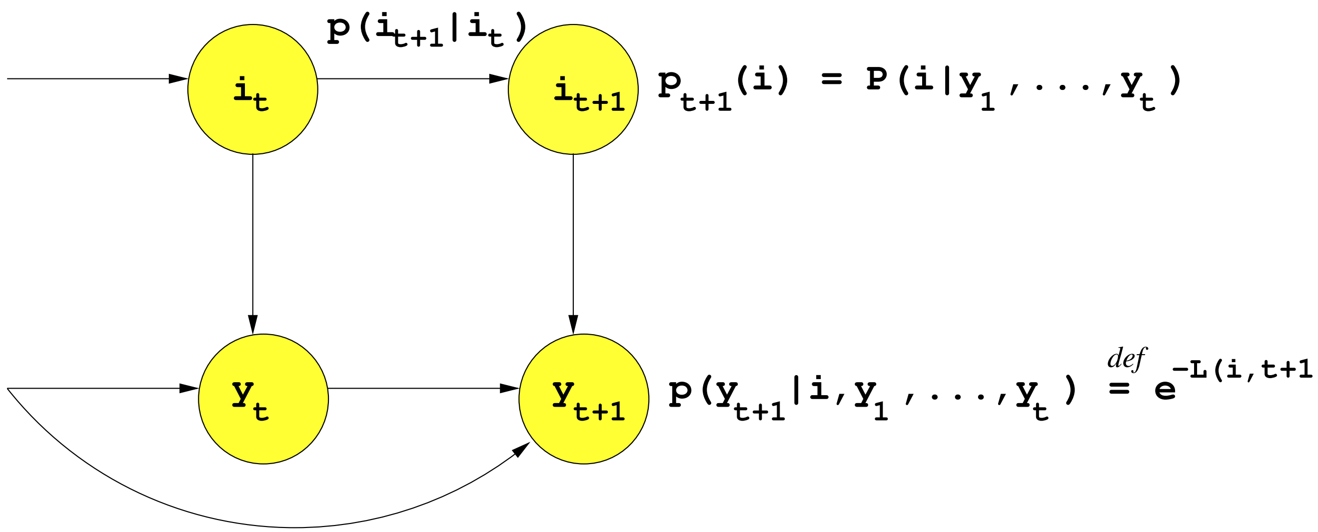
Related Work

- Algorithms for universal prediction, with performance guarantees:
  - relative to best expert [Littlestone and Warmuth, 1989]
  - relative to best sequence of experts [Herbster and Warmuth, 1998], [Vovk, 1999]
  - proven for many pairings of loss and prediction functions [Haussler et al., 1998]
- Algorithms with similar guarantees for:
  - adaptive game playing [Freund and Schapire, 1999]
  - online portfolio management [Helmbold et al., 1996]
  - paging [Blum et al., 1999]
  - k-armed bandit problem [Auer et al., 1995]
- Other relative performance measures for universal prediction, e.g. systematic variations [Foster and Vohra, 1999].

Outline

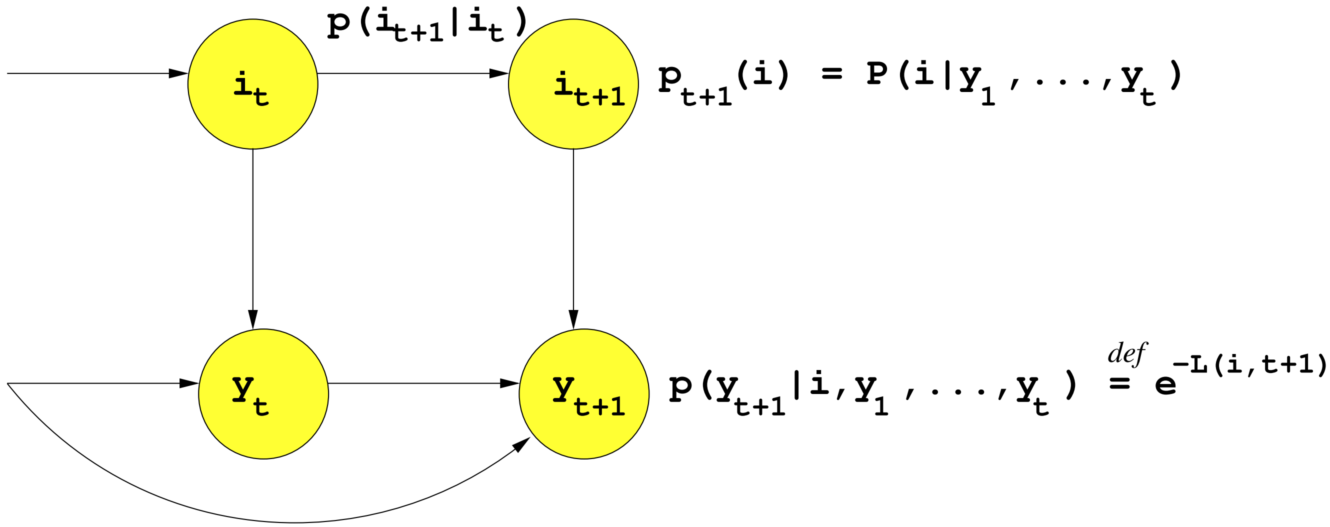
- Online learning framework
  - related work
  - HMM view of existing algorithms
  - our motivation

Algorithms



- Existing algorithms can be viewed as Bayesian updates in this graphical model
  - identity of current best expert is hidden (state) variable
  - $p(i_t|i_{t-1})$ defined by transition matrix Θ.
  - prediction, $P(y_t|y_1,\ldots,y_{t-1}) = \sum_{i=1}^{n} p_t(i)\, p(y_t|i,y_1,\ldots,y_{t-1})$

Algorithms



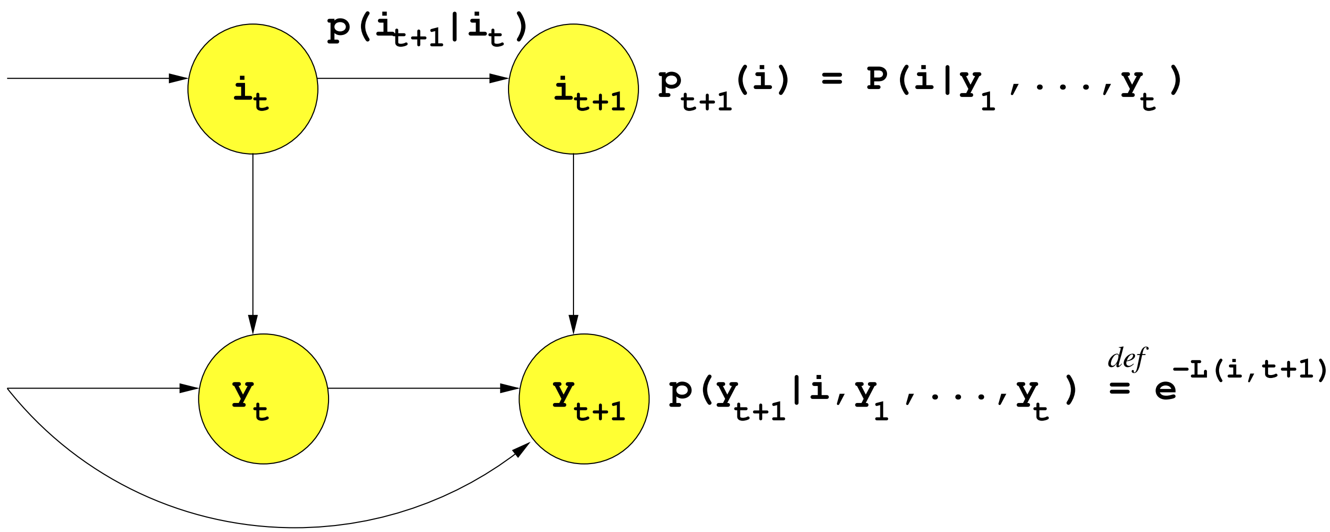
– Set emission probabilities, $p(y_t|i, y_1, \ldots, y_{t-1}) = e^{-L(i,t)}$, so $L(i,t) = -\log p(y_t|i, y_1, \ldots, y_{t-1})$.

– Bayesian updates of $p_t(i)$:

$$p_{t+1}(i) = \frac{1}{Z_{t+1}} \sum_{j=1}^{n} p_t(j) e^{-L(j,t)} p(i|j;\Theta)$$

where $p_1(i) = 1/n$. (cf. forward propagation in HMMs)

Algorithms



– Log-loss of the algorithm

$$
\begin{aligned}
L(p_t, t) &= -\log \sum_{i=1}^{n} p_t(i)\, p(y_t | i, y_1, \ldots, y_{t-1}) \\
&= -\log \sum_{i=1}^{n} p_t(i) e^{-L(i,t)}
\end{aligned}
$$

Note: can bound other loss functions [Haussler et al., 1998]

Transition Dynamics

- Transition probability matrix Θ is learner's model of non-stationarity of observation sequence.
- Choosing Θ according to

$$\theta_{ij} = \begin{cases} (1-\alpha) & i = j \\ \frac{\alpha}{n-1} & i \neq j \end{cases}$$

yields `Fixed-share` algorithm of [Herbster and Warmuth, 1998].

  – `Static-expert` algorithm of [Littlestone and Warmuth, 1989] when follows by setting $\alpha = 0$.

Our Motivation

- Improve online learning in (possibly) non-stationary case.
  - remove prior assumptions
  - existing algorithms take switching-rate, α, as a parameter.
- Design new algorithm to learn α online, simultaneous to original learning task.
- Yields algorithm whose regret is upper bounded by $\mathcal{O}(\log T)$.
  - whereas regret of existing algorithms:
    - upper bound $\mathcal{O}(T)$.
    - lower bound can be $\mathcal{O}(T)$.
- Regret-optimal discretization requires regret bound WRT `Fixed-share`(α^*)
  - where α^* is hindsight-optimal setting of switching-rate α, for the sequence observed.

Outline

- Online learning framework
- Upper and lower regret bounds for a class of online learning algorithms
  - technique for regret bounds
  - upper bound
  - lower bound

Regret

- Cumulative loss of the Bayesian algorithm (`Fixed-share`), using parameter α, over T training examples is

$$L_T(\alpha) = \sum_{t=1}^{T} L(p_{t;\alpha}, t)$$

- This can be expressed as the negative log-probability of all the observations, given the model (cf. HMMs):

$$L_T(\alpha) = -\log[\sum_{\vec{s}} \phi(\vec{s}) p(\vec{s}; \alpha)]$$

where $\vec{s} = \{i_1, \ldots, i_T\}$, $\phi(\vec{s}) = \prod_{t=1}^{T} e^{-L(i_t,t)}$, and

$$p(\vec{s}; \alpha) = p_1(i_1) \prod_{t=2}^{T} p(i_t | i_{t-1}; \alpha)$$

- "Regret" for using α, instead of hindsight-optimal, α^* for that sequence: $L_T(\alpha) - L_T(\alpha^*) = -\log \frac{\sum_{\vec{s}} \phi(\vec{s}) p(\vec{s};\alpha)}{\sum_{\vec{r}} \phi(\vec{r}) p(\vec{r};\alpha^*)}$

$$
\begin{aligned}
&= - \log \left[\sum_{\vec{s}} \left(\frac{\phi(\vec{s}) p(\vec{s};\alpha^*)}{\sum_{\vec{r}} \phi(\vec{r}) p(\vec{r};\alpha^*)} \right) \frac{p(\vec{s};\alpha)}{p(\vec{s};\alpha^*)} \right] \\
&= - \log \left[\sum_{\vec{s}} Q(\vec{s};\alpha^*) \frac{p(\vec{s};\alpha)}{p(\vec{s};\alpha^*)} \right] = -\log \left[\sum_{\vec{s}} Q(\vec{s};\alpha^*) e^{\log \frac{p(\vec{s};\alpha)}{p(\vec{s};\alpha^*)}} \right] \\
&= - \log \left[\sum_{\vec{s}} Q(\vec{s};\alpha^*) e^{(T-1)\left(\hat{\alpha}(\vec{s}) \log \frac{\alpha}{\alpha^*} + (1-\hat{\alpha}(\vec{s})) \log \frac{1-\alpha}{1-\alpha^*} \right)} \right]
\end{aligned}
$$

- $Q(\vec{s}|\alpha^*)$ is the posterior probability over the choices of experts along the sequence, induced by α^*.[2]
- $\hat{\alpha}(\vec{s})$ is the empirical fraction of non-self-transitions in $\vec{s}$.

[2] Q and α^* summarize the observed sequence.

Technique for Regret Bounds

- Regret WRT hindsight-optimal algorithm can be expressed as:

$$L_T(\alpha) - L_T(\alpha^*) = -\log\left[E_{\hat{\alpha}\sim Q}\, e^{(T-1)[D(\hat{\alpha}\|\alpha^*) - D(\hat{\alpha}\|\alpha)]}\right]$$

- Upper and lower bound regret, by finding optimizing Q in $\mathcal{Q}$, the set of all distributions, of this expression.

- Upper bound:

$$\max_{Q\in\mathcal{Q}}\left\{-\log\left[E_{\hat{\alpha}\sim Q}\, e^{(T-1)[D(\hat{\alpha}\|\alpha^*) - D(\hat{\alpha}\|\alpha)]}\right]\right\}$$

subject to constraint:

$$(1)\qquad \frac{d}{d\alpha}(L_T(\alpha) - L_T(\alpha^*))_{|\alpha=\alpha^*} = 0$$

Technique for Regret Bounds

- Lower bound:

$$\min_{Q \in \mathcal{Q}} \left\{ -\log \left[E_{\hat{\alpha} \sim Q}\, e^{(T-1)[D(\hat{\alpha}\|\alpha^*) - D(\hat{\alpha}\|\alpha)]} \right] \right\}$$

subject to constraint (1) and

$$(2) \qquad \frac{d^2}{d\alpha^2}(L_T(\alpha) - L_T(\alpha^*))_{|\alpha=\alpha^*} = \frac{\beta^*(T-1)}{\alpha^*(1-\alpha^*)}$$

where β^*, is relative quality of regret minimum at α^*, defined as:

$$\beta^* = \frac{\alpha^*(1-\alpha^*)}{T-1} \frac{d^2}{d\alpha^2}(L_T(\alpha) - L_T(\alpha^*))_{|\alpha=\alpha^*}$$

where normalization guarantees $\beta^* \leq 1$. And $\beta^* \geq 0$ for any α^* that minimizes $L_T(\alpha)$.

Upper Bound on Regret

Theorem 1: *For a Bayes learner on the graphical model above, with arbitrary transition matrix Θ, the regret on a sequence of T observations with respect to the hindsight-optimal transition matrix Θ^* for that sequence, is:*

$$L_T(\Theta) - L_T(\Theta^*) \leq (T-1) \max_{i \in \{1,\ldots,n\}} D(\Theta_i^* \| \Theta_i)$$

Corollary: *For a* `Fixed-share(`α`)` *algorithm, the regret on T observations, with respect to the hindsight optimal α^* for that sequence is:*

$$L_T(\alpha) - L_T(\alpha^*) \leq (T-1)\, D(\alpha^* \| \alpha)$$

Bound vanishes when $\alpha = \alpha^*$, and no direct dependence on n (unlike previous work). The maximizing Q is a point mass at α^*.

A Lower Bound on Regret

A non-trivial lower bound using an additional statistic on observed sequence, β^*.

Theorem 2: *Define* $\underline{Q}(1) = q_1 = [1 + \frac{T-1}{1-\beta^*}\frac{1-\alpha^*}{\alpha^*}]^{-1}$, $\underline{Q}(\frac{\alpha^*-q_1}{1-q_1}) = 1 - q_1$, *when* $\alpha \geq \alpha^*$, *and* $\underline{Q}(0) = q_0 = [1+\frac{T-1}{1-\beta^*}\frac{\alpha^*}{1-\alpha^*}]^{-1}$, $Q(\frac{\alpha^*}{1-q_0}) = 1-q_0$, *when* $\alpha < \alpha^*$. *Then for a* `Fixed-share(`α`)` *algorithm, the regret on any* T *observations consistent with* α^* *and* β^* *is:*

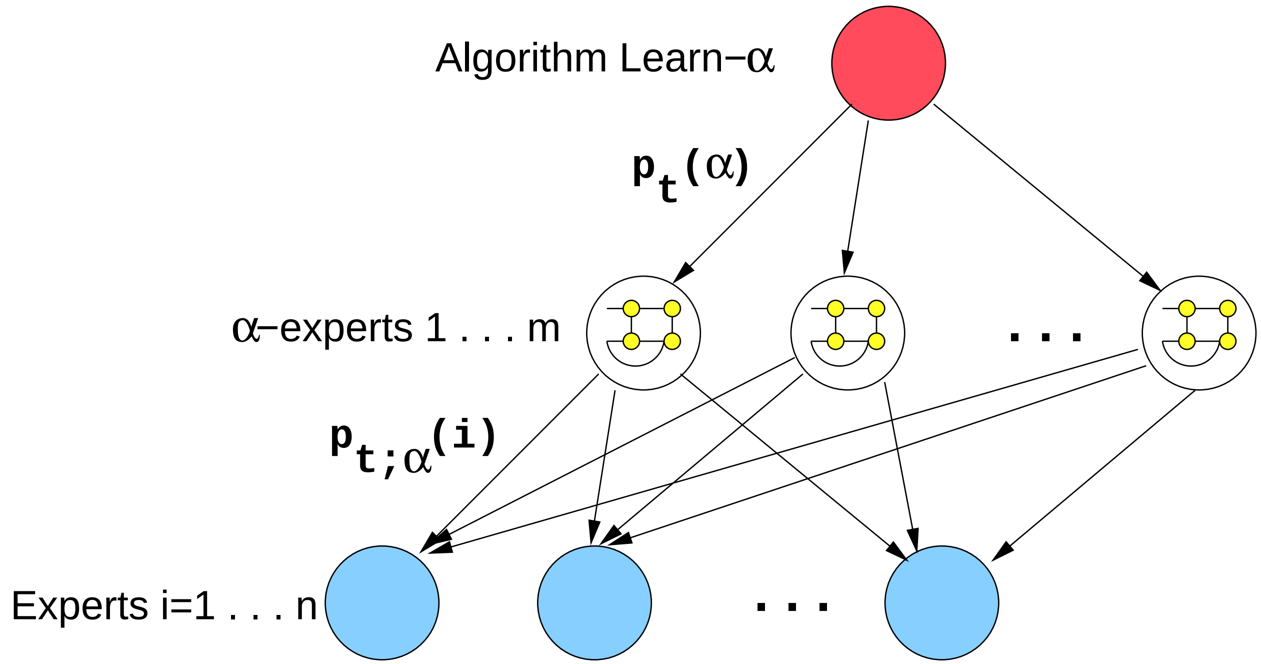
$$L_T(\alpha) - L_T(\alpha^*) \geq -\log\left[E_{\hat{\alpha}\sim\underline{Q}}\, e^{(T-1)[D(\hat{\alpha}\|\alpha^*)-D(\hat{\alpha}\|\alpha)]}\right]$$

- Bound is non-trivial only when $\beta^* > 0$ (sequences for which α^* is non-trivial minimizer)
- As $\beta^* \to 1$, upper and lower bounds agree: $(T-1)D(\alpha^*\|\alpha)$

Outline

- Online learning framework
- Upper and lower regret bounds for a class of online learning algorithms
- An algorithm that simultaneously learns the switching-rate, α, at optimal discretization
  - Algorithm `Learn-`α
  - Regret-optimal discretization
- A stronger bound on regret of new algorithm

Regret-Optimal Learning of α



- Algorithm `Learn-`α: a hierarchical algorithm that simultaneously learns the switching-rate α online
  - track the best "α-expert" (`Static-expert` updates).
  - set of m α-experts, i.e. `Fixed-share(`α`)` algorithms.
  - posterior over switching-rates:

    $$p_t(\alpha) = P(\alpha|y_{t-1},\ldots,y_1) = c\cdot e^{-L_{t-1}(\alpha)}$$

Regret-Optimal Learning of α

Algorithm `Learn-`α

- Bayesian updates (cf. `Static-expert`):

$$p_{t+1}(\alpha_j) = \frac{1}{Z_{t+1}} p_t(\alpha_j) e^{-L(\alpha_j,t)}$$

where $p_1(\alpha_j) = 1/m$, and $L(\alpha_j, t) = L(p_{t;\alpha_j}, t)$.

- Loss of algorithm is thus

$$\begin{aligned} L^{top}(p_t, t) &= -\log \sum_{j=1}^{m} p_t(\alpha_j) e^{-L(\alpha_j,t)} \\ &= -\log \sum_{j=1}^{m} \sum_{i=1}^{n} p_t(\alpha_j) p_{t;\alpha_j}(i) e^{-L(i,t)} \end{aligned}$$

as is appropriate for a hierarchical Bayesian method.

Regret-Optimal Learning of α

- Optimal discretization: Find a regret-optimal discrete set of switching rates $\{\alpha_1, \ldots, \alpha_m\}$
- Optimize tradeoff between loss due to exploration (too many α_j's), and loss of α_{j^*} WRT α^* (too few).
- Choose the minimal set s.t. loss of α_{j^*} WRT α^* is bounded.
  - for any α^* we require that there is α_j s.t. the cumulative regret is upper bounded by $(T-1)\delta$.
  - by regret bound $L_T(\alpha_{j^*}) - L_T(\alpha^*) \le (T-1)\, D(\alpha^* \| \alpha_{j^*})$
  - so we require:

$$\max_{\alpha^* \in [0,1]} \min_{j=1,\ldots,m(\delta)} D(\alpha^* \| \alpha_j) = \delta$$

  - $m(\delta)$ is also computed by discretization algorithm.

Regret-Optimal Learning of α

Discretization algorithm:

- Set α_1 s.t.

$$\max_{\alpha^* \in [0,\alpha_1]} D(\alpha^* \| \alpha_1) = D(0\|\alpha_1) = \delta \quad \Longrightarrow \quad \alpha_1 = 1 - e^{-\delta}$$

- Set α_j (iteratively) s.t.

$$\max_{\alpha^* \in [\alpha_{j-1},\alpha_j]} \min\{D(\alpha^* \| \alpha_{j-1}), D(\alpha^* \| \alpha_j)\} = \delta$$

- Maximizing α^* has closed form solution, which is increasing function of α_j.
- Using this α^*, solve for α_j in $D(\alpha^* \| \alpha_{j-1}) = \delta$, e.g. via bisection search.
- Assign $\alpha_j \geq \frac{1}{2}$ by symmetry of $D(\cdot \| \cdot)$ on $[0, 1]$.

Upper Bound on Regret of Learn-α

Theorem 3: *The regret of* **Learn-α** *on a sequence of T observations, with respect to the hindsight-optimal* **Fixed-share(α^*)** *algorithm for that sequence is*

$$L_T^{top} - L_T(\alpha^*) \leq (T-1) \min_{j=1,\ldots,m(\delta)} D(\alpha^* \| \alpha_j) + \log(m(\delta))$$

Proof:

$$\begin{aligned} L_T^{top} &\leq \min_{j=1,\ldots,m(\delta)} L_T(\alpha_j) + \log(m(\delta)) \\ &\leq L_T(\alpha^*) + (T-1) \min_{j=1,\ldots,m(\delta)} D(\alpha^* \| \alpha_j) + \log(m(\delta)) \end{aligned}$$

by applying relative loss bound on `Static-expert`,[3] and then new relative loss bound on `Fixed-share`.

$\square$

[3] [Littlestone and Warmuth, 1989]

Upper Bound on Regret of `Learn-`α

- By discretization method, bound is

$$L_T^{top} - L_T(\alpha^*) \quad \leq \quad (T-1)\delta + \log m(\delta)$$

- δ is a free parameter so we can optimize the bound, without knowledge of the observation sequence.
  - since $\log m(\delta) \approx -1/2 \log \delta$ for small δ, regret bound becomes

$$L_T^{top} - L_T(\alpha^*) \quad \approx \quad (T-1)\delta - \frac{1}{2}\log \delta$$

  - optimize to attain $\delta^* = 1/(2T)$, and $m(\delta^*) = \sqrt{2T}$.
  - thus require $\mathcal{O}(\sqrt{T})$ settings of α.
    - independent of n.

Upper Bound on Regret of `Learn-`α

Optimized regret bound:

$$\frac{1}{2}\log T + c$$

- Upper bound on regret of `Learn-`α is thus $\mathcal{O}(\log T)$.
- cf. lower bound on regret of `Fixed-share`
  – can be $\mathcal{O}(T)$.

Algorithmic complexity:

- time $\mathcal{O}(nm)$, or $\mathcal{O}(n+m)$ time and $\mathcal{O}(m)$ space (in parallel).
- in optimized version: $\mathcal{O}(n\sqrt{T})$, or $\mathcal{O}(n+\sqrt{T})$ with space $\mathcal{O}(\sqrt{T})$.

Outline

- Online learning framework
- Upper and lower regret bounds for a class of online learning algorithms
- An algorithm that simultaneously learns the switching-rate, α, at optimal discretization
- A stronger bound on regret of new algorithm
- Application to wireless networks

Application to Wireless Networks

IEEE 802.11 Energy/Performance Tradeoff

- Energy: 802.11 wireless nodes consume more energy in AWAKE than SLEEP
- Performance: node cannot receive packets while sleeping $\rightarrow$ introduces latency
- IEEE 802.11 Power Saving Mode:
  - Base station can buffer packets while node is sleeping
  - Use of a fixed polling time (100ms) at which to WAKE, receive buffered packets, and then go back to sleep.
- Related work:
  - Adaptive control [Krashinsky and Balakrishnan, 2002]
  - Reinforcement Learning [Steinbach, 2002]

Algorithm Formulation for Application

- Problem is apt for online learning, specifically `Learn-`α
  - network conditions vary over time, and location, thus cannot set α beforehand.
- n experts: constant settings of polling time, T_i.
- Run `Learn-`α, using $m(\delta^*)$ α-experts, or sub-algorithms running `Fixed-share(`α`)`.
- Observe/update at epochs, t, only upon awakening. Define:
  - I_t: number of bytes buffered since last wake-up.
  - T_t: time slept for.

Algorithm Formulation for Application

- Loss per expert:

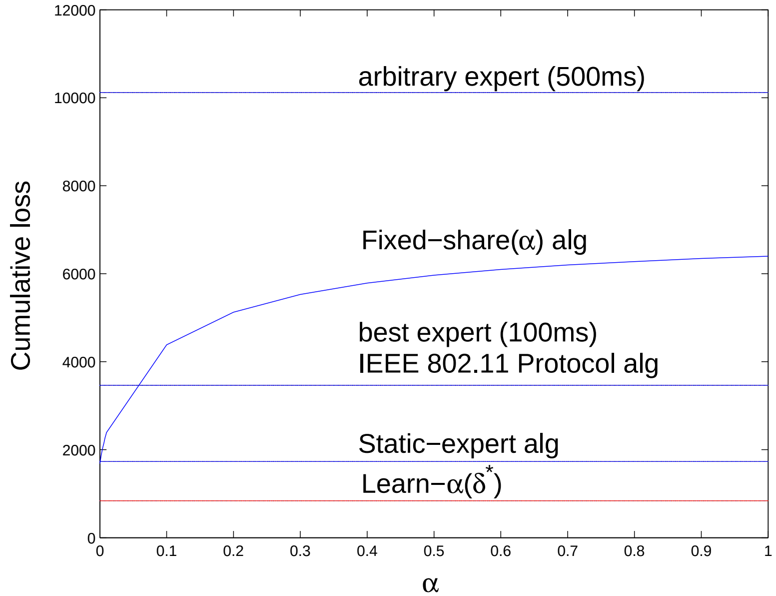
$$L(i,t) = \gamma \frac{I_t T_i^2}{2T_t} + \frac{1}{T_i}$$

  - first term approximates[4] latency introduced by buffering I_t bytes, scaled by how long i would have slept.
  - second term encodes energy penalty for waking often.
  - γ: user specified scaling to quantify preferred tradeoff.[5]
  - sum of convex functions $\implies$ unique minimum.

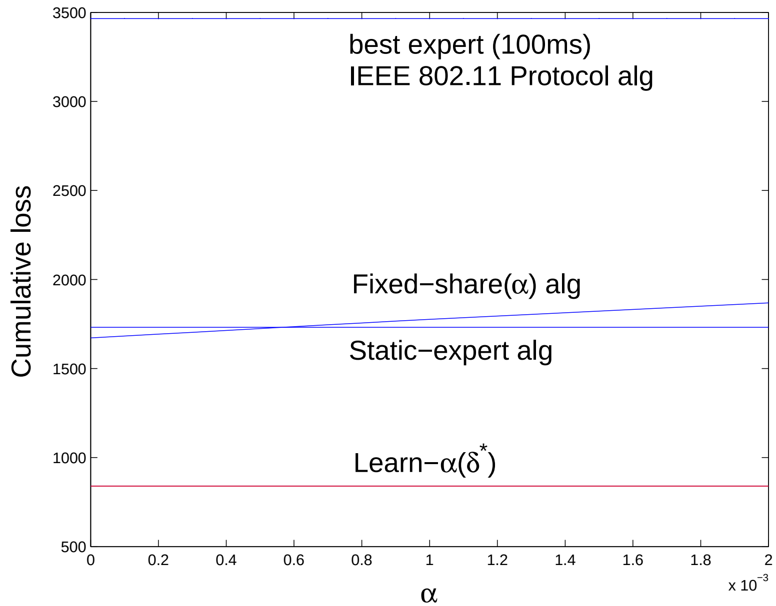
[4] Assume uniform arrival rate while sleeping, since cannot observe.

[5] Or, Lagrange multiplier on latency constraint, in an energy minimization.

Results

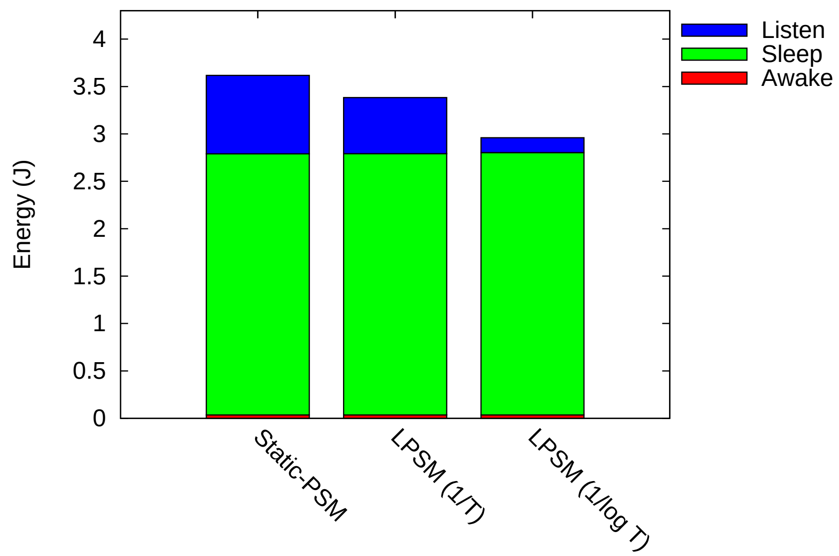


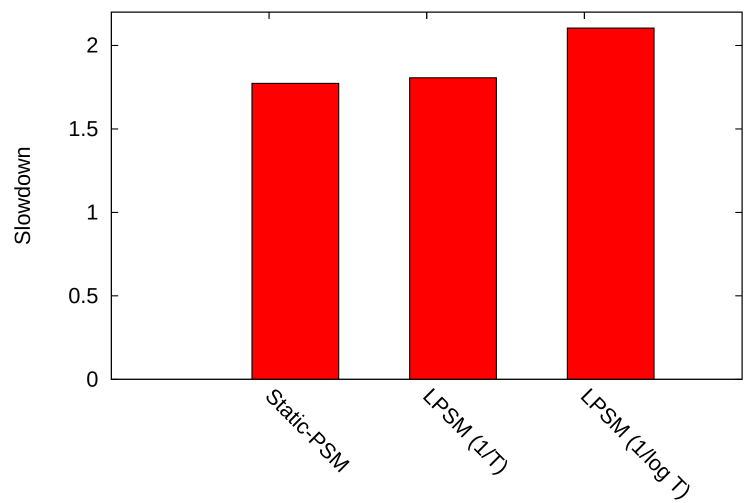
arbitrary expert (500ms)
Fixed−share(α) alg
best expert (100ms)
IEEE 802.11 Protocol alg
Static−expert alg
Learn−α(δ*)
Cumulative loss
α
0
2000
4000
6000
8000
10000
12000
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



best expert (100ms)
IEEE 802.11 Protocol alg
Fixed−share(α) alg
Static−expert alg
Learn−α(δ*)
Cumulative loss
α
500
1000
1500
2000
2500
3000
3500
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x 10−3

Results[6]





[6] Joint work with Hari Balakrishnan and Nick Feamster. `ns2` network simulation.

Summary and Future Work

- Upper and lower regret bounds for `Fixed-share` algorithms
  - new proof technique, comparison class
- Optimal discretization for learning the switching-rate online
  - new algorithm has stronger regret bound
- Application to wireless energy management with performance gains

- Upper bound for any transition matrix (listed here)
- Lower bound for any transition matrix in progress.
- Extension of optimal discretization to multi-dimensional simplex, (for learning transition matrix) in progress.

Some Proof Details

- **Proof of Theorem 1:** Upper Bound:
  Constraint (1) is equivalent to $E_{\hat{\alpha}\sim Q}\{\hat{\alpha}\} = \alpha^*$.
  Take expectation outside of logarithm. $\Box$
- **Proof of Theorem 2:** Lower Bound:
  (2) equivalent to $E_{\hat{\alpha}\sim Q}\left[(\hat{\alpha}-\alpha^*)^2\right] = \frac{(1-\beta^*)\alpha^*(1-\alpha^*)}{T'} \equiv \beta_2^*$, where $T' = T-1$.
  Find the form of the minimizing Q by inspecting $J(Q,\vec{\lambda})$ given by

$$E_{\hat{\alpha}\sim Q}\left[f(\hat{\alpha};\alpha,\alpha^*) - \lambda_1(\hat{\alpha}-\alpha^*) - \lambda_2\big((\hat{\alpha}-\alpha^*)^2 - \beta_2^*\big)\right]$$

  where $f(\hat{\alpha};\alpha,\alpha^*) = \exp\left\{T'\big(\hat{\alpha}\log\frac{\alpha}{\alpha^*} + (1-\hat{\alpha})\log\frac{1-\alpha}{1-\alpha^*}\big)\right\}$.
  $\Rightarrow$ Q can be non-zero only at two points, where one of the points is 0 or 1 (convexity argument, see paper for details).

Solve mean (1) and variance (2) constraints to find optimizing Q (here $\alpha < \alpha^*$, points 0 and a):

$$
\begin{aligned}
0 \times q_0 + a(1 - q_0) &= \alpha^* & (1) \\
q_0(0 - \alpha^*)^2 + (1 - q_0)(a - \alpha^*)^2 &= \frac{(1 - \beta^*)\alpha^*(1 - \alpha^*)}{T'} & (2)
\end{aligned}
$$

giving: $a = \frac{\alpha^*}{1 - q_0}, q_0 = \frac{1}{1 + \frac{T'}{1-\beta^*}\frac{\alpha^*}{1-\alpha^*}}$. Substitution yields bound.$\square$

Comparison of Upper Bounds

- [Herbster and Warmuth, 1998] bound loss relative to loss of the best k-partition of the observation sequence, where:
  - the best expert is assigned to each segment.
  - bound parameters: k, α^*.

$$L_T(\alpha) - L_T(\text{best } k\text{-partition}) \leq (T-1)[H(\alpha_k^*) + D(\alpha_k^* \| \alpha)] + k \log(n-1) + \log n$$

where $\alpha_k^* = k/(T-1)$.

- Bounds are comparable, but differ in comparison class.
  - Computing regret-optimal discretization for learning α required a bound with respect to α^*.

- $L_T(\alpha^*) - L_T(\text{best } k\text{-partition}) =$

$$= \quad -\log\frac{1}{n} - k\log\frac{\alpha^*}{n-1} - (T'-k)\log(1-\alpha^*)$$

  where the terms are the negative log-probability, given the `Fixed-share(`α^*`)`'s model of:

  1. choosing the start state
  2. making the k switches (done by the best k-partition)
  3. staying with one expert, during each of the k segments in the best k-partition.

- Bounds are comparable when $\alpha_k^* = \alpha^*$. Simplification and subsitution of $k = T'\alpha_k^*$ yields:

$$\begin{aligned} &= \quad -T'\log(1-\alpha_k^*) + T'\alpha_k^*\log(1-\alpha_k^*) - T'\alpha_k^*\log\alpha_k^* \\ &\qquad\qquad + k\log(n-1) + \log n \\ &= \quad T'H(\alpha_k^*) + k\log(n-1) + \log n \end{aligned}$$

which is exact form of difference in the bounds.

References

[Auer et al., 1995] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (1995). Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proc. of the 36th Annual Symposium on Foundations of Computer Science*, pages 322–331.

[Blum et al., 1999] Blum, A., Burch, C., and Kalai, A. (1999). Finely-competitive paging. In *IEEE 40th Annual Symposium on Foundations of Computer Science*, page 450, New York, New York.

[Foster and Vohra, 1999] Foster, D. P. and Vohra, R. (1999). Regret in the on-line decision problem. *Games and Economic*

Behavior, 29:7–35.

[Freund and Schapire, 1999] Freund, Y. and Schapire, R. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103.

[Haussler et al., 1998] Haussler, D., Kivinen, J., and Warmuth, M. K. (1998). Sequential prediction of individual sequences under general loss functions. *IEEE Transactions on Information Theory*, 44(5):1906–1925.

[Helmbold et al., 1996] Helmbold, D. P., Schapire, R. E., Singer, Y., and Warmuth, M. K. (1996). On-line portfolio selection using multiplicative updates. In *International Conference on Machine Learning*, pages 243–251.

[Herbster and Warmuth, 1998] Herbster, M. and Warmuth,

M. K. (1998). Tracking the best expert. *Machine Learning*, 32:151–178.

[Krashinsky and Balakrishnan, 2002] Krashinsky, R. and Balakrishnan, H. (2002). Minimizing energy for wireless web access with bounded slowdown. In *MobiCom 2002*, Atlanta, GA.

[Littlestone and Warmuth, 1989] Littlestone, N. and Warmuth, M. K. (1989). The weighted majority algorithm. In *IEEE Symposium on Foundations of Computer Science*, pages 256–261.

[Steinbach, 2002] Steinbach, C. (2002). A reinforcement-learning approach to power management. In *AI Technical Report, M.Eng Thesis*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[Vovk, 1999] Vovk, V. (1999). Derandomizing stochastic prediction strategies. *Machine Learning*, 35:247–282.