Advanced Machine Learning

Learning with Large Expert Spaces

MEHRYAR MOHRI MOHRI@

COURANT INSTITUTE & GOOGLE RESEARCH

Problem

- Learning guarantees:

$$R_T = O(\sqrt{T \log N}).$$

$\longrightarrow$ informative even for N very large.

- Problem: computational complexity of algorithm in $O(N)$. Can we derive more efficient algorithms when experts admit some structure and when loss is decomposable?

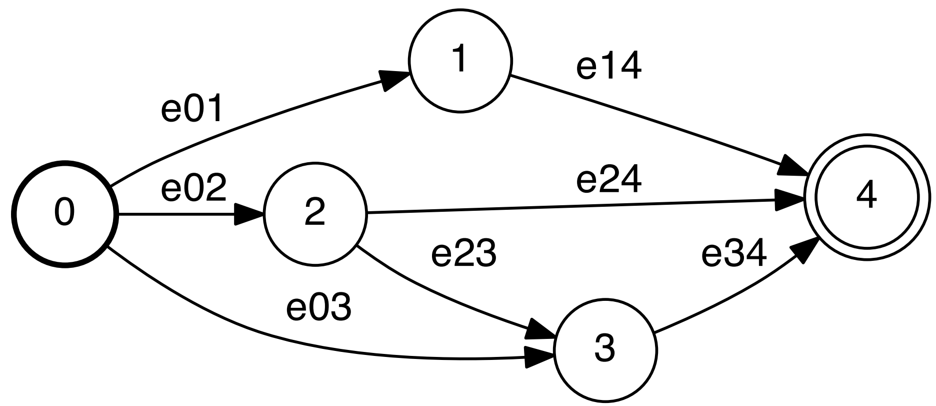
Example: Online Shortest Path

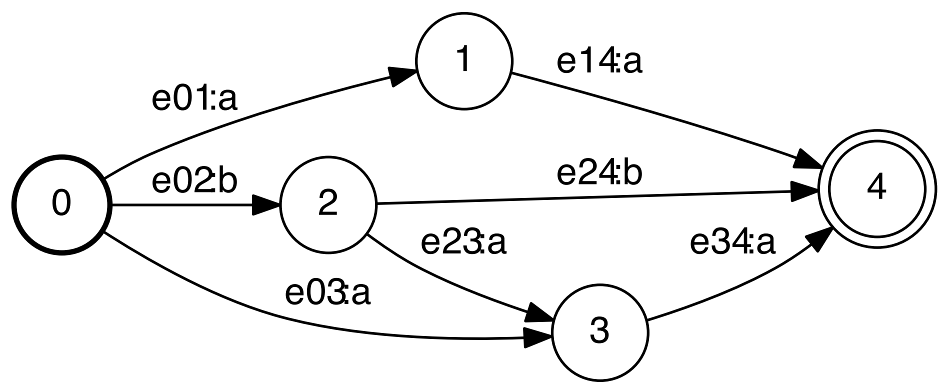
- Problems: path experts.
  - sending packets along paths of a network with routers (vertices); delays (losses).
  - car route selection in presence of traffic (loss).

Outline

- RWM with Path Experts
- FPL with Path Experts

Path Experts

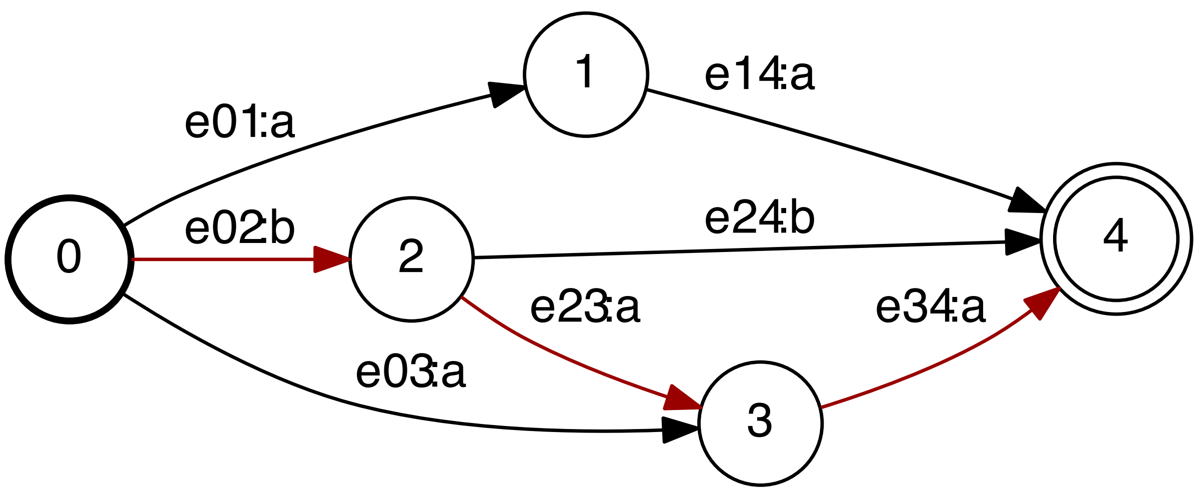




Additive Loss

- For path $\xi = e_{02}e_{23}e_{34}$,

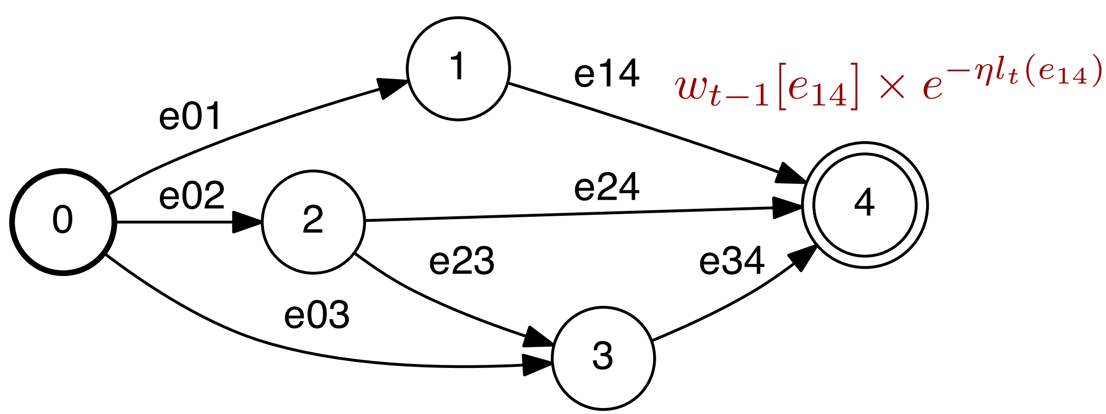
$$l_t(\xi) = l_t(e_{02}) + l_t(e_{23}) + l_t(e_{34}).$$



RWM + Path Experts

(Takimoto and Warmuth, 2002)

- Weight update: at each round t, update weight of path expert $\xi = e_1 \cdots e_n$:
  - $w_t[\xi] \leftarrow w_{t-1}[\xi]\, e^{-\eta l_t(\xi)}$; equivalent to
  - $w_t[e_i] \leftarrow w_{t-1}[e_i]\, e^{-\eta l_t(e_i)}$.



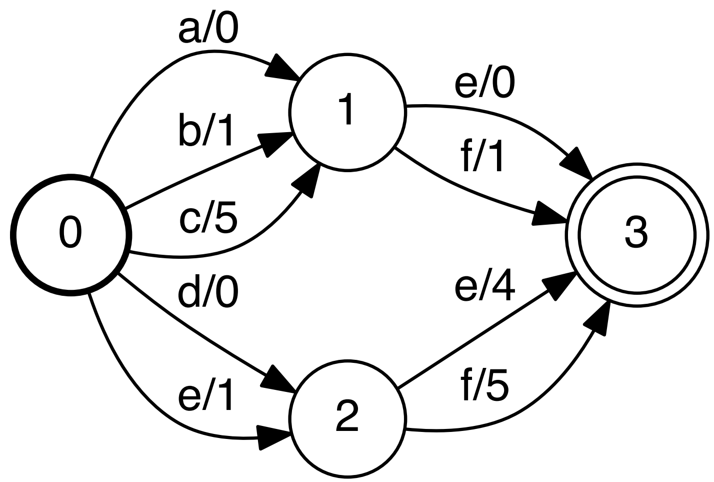
- Sampling: need to make graph/automaton stochastic.

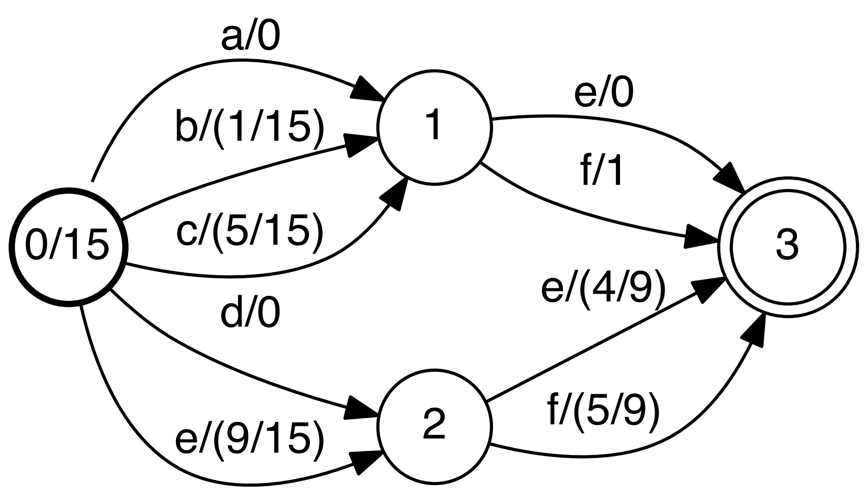
Weight Pushing Algorithm

(MM 1997; MM, 2009)

- Weighted directed graph $G = (Q, E, w)$ with set of initial vertices $I \subseteq Q$ and final vertices $F \subseteq Q$:
  - for any $q \in Q$,
$$d[q] = \sum_{\pi \in P(q,F)} w[\pi].$$
  - for any $e \in E$ with $d[\mathrm{orig}(e)] \neq 0$,
$$w[e] \leftarrow d[\mathrm{orig}(e)]^{-1} \cdot w[e] \cdot d[\mathrm{dest}(e)].$$
  - for any $q \in I$, initial weight
$$\lambda(q) \leftarrow d(q).$$

Illustration





Properties

- Stochasticity: for any $q \in Q$ with $d[q] \neq 0$,

$$\sum_{e \in E[q]} w'[e] = \sum_{e \in E[q]} \frac{w[e]\, d[\mathrm{dest(e)}]}{d[q]} = \frac{d[q]}{d[q]} = 1.$$

- Invariance: path weight preserved. Weight of path $\xi = e_1 \cdots e_n$ from I to F:

$$\begin{aligned}
&\lambda(\mathrm{orig}(e_1)) w'[e_1] \cdots w'[e_n] \\
&= d[\mathrm{orig}(e_1)] \frac{w[e_1] d[\mathrm{dest}(e_1)]}{d[\mathrm{orig}(e_1)]} \frac{w[e_2] d[\mathrm{dest}(e_2)]}{d[\mathrm{dest}(e_1)]} \cdots \\
&= w[e_1] \cdots w[e_n] d[\mathrm{dest}(e_n)] \\
&= w[e_1] \cdots w[e_n] = w[\xi].
\end{aligned}$$

Shortest-Distance Computation

- Acyclic case:
  - special instance of a generic single-source shortest-distance algorithm working with an arbitrary queue discipline and any *k*-closed semiring (MM, 2002).
  - linear-time algorithm with the topological order queue discipline, $O(|Q| + |E|)$.

Generic Single-Source SD Algo.

(MM, 2002)

Gen-Single-Source(G, s)

for $i \leftarrow 1$ **to** $|Q|$ **do**
    $d[i] \leftarrow r[i] \leftarrow \overline{0}$
$d[s] \leftarrow r[s] \leftarrow \overline{1}$
$\mathcal{Q} \leftarrow \{s\}$
while $\mathcal{Q} \neq \emptyset$ **do**
    $q \leftarrow \text{Head}(\mathcal{Q})$
    $\text{Dequeue}(\mathcal{Q})$
    $r' \leftarrow r[q]$
    $r[q] \leftarrow \overline{0}$
    for each $e \in E[q]$ **do**
        if $d[n[e]] \neq d[n[e]] \oplus (r' \otimes w[e])$ **then**
            $d[n[e]] \leftarrow d[n[e]] \oplus (r' \otimes w[e])$
            $r[n[e]] \leftarrow r[n[e]] \oplus (r' \otimes w[e])$
            if $n[e] \notin \mathcal{Q}$ **then**
                $\text{Enqueue}(\mathcal{Q}, n[e])$

Shortest-Distance Computation

- General case:
  - all-pairs shortest-distance algorithm in $(+, \times)$; for all pairs of vertices (p, q),
  $$d[p, q] = \sum_{\pi \in P(p,q)} w[\pi].$$
  - generalization of Floyd-Warshall algorithm to non-idempotent semirings (MM, 2002).
  - time complexity in $O(|Q|^3)$, space complexity in $O(|Q|^2)$.
  - alternative: approximation using generic single-source shortest-distance algorithm (MM, 2002).

Generic All-Pairs SD Algorithm

(MM, 2002)

$\textsc{Gen-All-Pairs}(G)$

for $i \leftarrow 1$ **to** $|Q|$ **do**
  for $j \leftarrow 1$ **to** $|Q|$ **do**
    $d[i, j] \leftarrow \bigoplus_{e \in E \cap P(i,j)} w[e]$
for $k \leftarrow 1$ **to** $|Q|$ **do**
  for $i \leftarrow 1$ **to** $|Q|, i \neq k$ **do**
    for $j \leftarrow 1$ **to** $|Q|, j \neq k$ **do**
      $d[i, j] \leftarrow d[i, j] \oplus (d[i, k] \otimes d[k, k]^* \otimes d[k, j])$
  for $i \leftarrow 1$ **to** $|Q|, i \neq k$ **do**
    $d[k, i] \leftarrow d[k, k]^* \otimes d[k, i]$
    $d[i, k] \leftarrow d[i, k] \otimes d[k, k]^*$
  $d[k, k] \leftarrow d[k, k]^*$

In-place version.

Learning Guarantee

- Theorem: let $\mathcal{N}$ be total number of path experts and M an upper bound on the loss of a path expert. Then, the (expected) regret of RWM is bounded as follows:

$$\mathcal{L}_T \leq \mathcal{L}_T^{\min} + 2M\sqrt{T \log \mathcal{N}}.$$

Exponentiated Weighted Avg

- Computation of the prediction at each round:

$$\widehat{y}_t = \frac{\sum_{\xi \in P(I,F)} w_t[\xi] y_{t,\xi}}{\sum_{\xi \in P(I,F)} w_t[\xi]}.$$

- Two single-source shortest-distance computations:
  - edge weight $w_t[e]$ (denominator).
  - edge weight $w_t[e] y_t[e]$(numerator).

FPL + Path Experts

- Weight update: at each round, update weight of edge e,

$$w_t[e] \leftarrow w_{t-1}[e] + l_t(e).$$

- Prediction: at each round, shortest path after perturbing each edge weight:

$$w'_t[e] \leftarrow w_t[e] + p_t(e),$$

where $\mathrm{p}_t \sim U([0, 1/\epsilon]^{|E|})$

or $\mathrm{p}_t \sim$ Laplacian with density $f(\mathbf{x}) = \frac{\epsilon}{2} e^{-\epsilon \|\mathbf{x}\|_1}$.

Learning Guarantees

- Theorem: assume that edge losses are in $[0, 1]$. Let $l_{\max}$ be the length of the longest path from I to F and M an upper bound on the loss of a path expert. Then,
  - the (expected) regret of FPL is bounded as follows:
  $$\mathrm{E}[R_T] \leq 2\sqrt{l_{\max} M |E| T} \leq 2 l_{\max} \sqrt{|E| T}.$$
  - the (expected) regret of FPL* is bounded as follows:
  $$\begin{aligned} \mathrm{E}[R_T] &\leq 4\sqrt{\mathcal{L}_T^{\min} |E| l_{\max} (1 + \log |E|)} + 4 |E| l_{\max} (1 + \log |E|) \\ &\leq 4 l_{\max} \sqrt{T |E| (1 + \log |E|)} + 4 |E| l_{\max} (1 + \log |E|) \\ &= O(l_{\max} \sqrt{T |E| \log |E|}). \end{aligned}$$

Proof

- For FPL, use bound of previous lectures with

$$X_1 = |E| \quad W_1 = l_{\max} \quad R = M \leq l_{\max}.$$

- For FPL*, use bound of previous lecture with

$$X_1 = |E| \quad W_1 = l_{\max} \quad N = |E|.$$

Computational Complexity

- For an acyclic graph:
  - T updates of all edge weights.
  - T runs of a linear-time single-source shortest-path.
  - overall $O(T(|Q| + |E|))$.

Extensions

- Component hedge algorithm (Koolen, Warmuth, and Kivinen, 2010):
  - optimal regret complexity: $R_T = O(M\sqrt{T \log |E|})$.
  - special instance of mirror descent.
- Non-additive losses (Cortes, Kuznetsov, MM, Warmuth, 2015):
  - extensions of RWM and FPL.
  - rational and tropical losses.

References

- Corinna Cortes, Vitaly Kuznetsov, and Mehryar Mohri. Ensemble methods for structured prediction. In ICML. 2014.
- Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Manfred K. Warmuth. On-line learning algorithms for path experts with non-additive losses. In COLT, 2015.
- Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
- T. van Erven, W. Kotlowski, and Manfred K. Warmuth. Follow the leader with dropout perturbations. In COLT, 2014.
- Adam T. Kalai, Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.* 71(3): 291-307. 2005.
- Wouter M. Koolen, Manfred K. Warmuth, and Jyrki Kivinen. Hedging structured concepts. In COLT, pages 93–105, 2010.

References

- Nick Littlestone, Manfred K. Warmuth: The Weighted Majority Algorithm. *FOCS 1989*: 256-261.
- Mehryar Mohri. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23:2, 1997.
- Mehryar Mohri. Semiring Frameworks and Algorithms for Shortest-Distance Problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321-350, 2002.
- Mehryar Mohri. Weighted automata algorithms. *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science, pages 213-254. Springer, 2009.
- Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. JMLR, 4:773–818, 2003.