Regret Minimization for Online Buffering Problems Using the Weighted Majority Algorithm

Sascha Geulen, Berthold Vöcking, *Melanie Winkler*

Department of Computer Science, RWTH Aachen University

June 27, 2010

Online Buffering

Toy example:

- buffer of bounded size B
- in every time step $t = 1, \ldots, T$:
  - demand $d^t \leq B$
  - $p^t \in [0, 1]$, price per unit of the resource OR
  - $f^t(x)$, price function to buy x units

How much should be purchased in time step t?

Main Application

- **Battery Management of Hybrid cars**
  - two energy resources (combustion / electrical)
  - given requested torque of the car, battery level
  - determine torque of combustion engine

Online Learning

Motivation:

- online buffering problems have been studied in Worst-Case Analysis
- algorithm is "threat-based", i.e. buys enough to ensure the competitive factor in the next step for all possible extensions of the price sequence

Online Learning Applied to Online Buffering

Algorithm 1 (`Randomized Weighted Majority (RWM)`)

1: $w_i^1 = 1, q_i^1 = \frac{1}{N}$, for all $i \in \{1, \dots, N\}$
2: **for** $t = 1, \dots, T$ **do**
3: choose expert e^t at random according to $Q^t = (q_1^t, \dots, q_N^t)$
4: $w_i^{t+1} = w_i^t (1 - \eta)^{c_i^t}$, for all i
5: $q_i^{t+1} = \frac{w_i^{t+1}}{\sum_{j=1}^N w_j^{t+1}}$, for all i
6: **end for**

Problem:

$$\begin{bmatrix} p^t \\ d^t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 1/4 \end{bmatrix} \begin{bmatrix} 1 \\ 1/4 \end{bmatrix} \begin{bmatrix} 0 \\ 1/4 \end{bmatrix} \begin{bmatrix} 1 \\ 1/4 \end{bmatrix} \right)^{T'} .$$

- The first expert purchases $1/2$ unit in the initial step and afterwards one unit in the third step of every round.
- The second expert purchases one unit in the first step of every round.

Online Learning for Online Buffering

Algorithm 2 (`Shrinking Dartboard (SD)`)

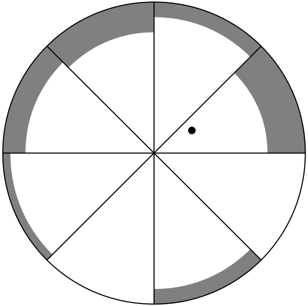
$w_i^1 = 1, q_i^1 = \frac{1}{N}$, for all i
choose expert e^1 at random according to $Q^1 = (q_1^1, \ldots, q_N^1)$
for $t = 2, \ldots, T$ **do**
    $w_i^t = w_i^{t-1}(1-\eta)^{c_i^{t-1}}$, for all i
    $q_i^t = \frac{w_i^t}{\sum_{j=1}^N w_j^t}$, for all i
    with probability $\frac{w_{e^t}^t}{w_{e^t}^{t-1}}$ do not change expert, i.e., set $e^t = e^{t-1}$
        else choose e^t at random according to $Q^t = (q_1^t, \ldots, q_N^t)$
end for

Shrinking Dartboard Algorithm

Idea: dartboard of size N, area of size 1 for expert i

1. set *active area* of expert i to 1
2. throw dart into active area to choose an expert
3. if weight of expert i decreases
   - decrease active area of that expert
4. dart outside of active area $\Rightarrow$ throw new dart

$\Rightarrow$ distribution to choose an expert is the same as for RWM in every step, but depends on e^{t-1}



Theorem

For $\eta = \min\{\sqrt{\ln N/(BT)}, 1/2\}$, the expected cost of SD satisfies

$$C_{\text{SD}}^T \leq C_{\text{best}}^T + O(\sqrt{BT \log N}).$$

Regret of Shrinking Dartboard

Proof idea:

Observation: $E[c_{\text{SD}}] \leq \sum_t c_{\text{chosen expert}} + B \cdot E[\text{number of expert changes}]$

1. expected cost of chosen expert $\Leftrightarrow$ cost of RWM: $(1+\eta)C^T_{\text{best}} + \frac{\ln N}{\eta}$
2. additional cost for every expert change are at most B
   - due to difference in number of units in the storage
3. estimate number of expert changes
   - W^t, remaining size of dartboard in step t, $(W^t = \sum_{i=1}^N w_i^t)$
   - size of dartboard larger than weight of best expert, $(W^{T+1} \geq (1+\eta)^{C^T_{\text{best}}})$
   - W^{T+1} equals product of fraction of dartboard which remains from t to $t+1$ multiplied by N, $(N \prod_{t=1}^T (1 - \frac{W^t - W^{t+1}}{W^t}))$
4. combining those equations leads to $C^T_{\mathrm{SD}} \leq C^T_{\mathrm{best}} + O(\sqrt{BT \log N})$.

Weighted Fractional Algorithm

Algorithm 3 (`Weighted Fractional (WF)`)

1: $w_i^1 = 1, q_i^1 = \frac{1}{N}$, for all i
2: **for** $t = 2, \dots, T$ **do**
3: purchase $x^t = \sum_{i=1}^N q_i x_i$ units, x_i amount purchased by i
4: $w_i^t = w_i^{t-1}(1-\eta)^{c_i^{t-1}}$, for all i
5: $q_i^t = \frac{w_i^t}{\sum_{j=1}^N w_j^t}$, for all i
6: **end for**

Idea: purchased amount is a weighted sum of the recommendations of the experts

Theorem

Suppose the price functions $f^t(x)$ are convex, for $1 \leq t \leq T$. Then for $\eta = \min\{\sqrt{\ln N/(BT)}, 1/2\}$ the cost of WF satisfies

$$C_{\mathrm{WF}}^T \leq C_{\mathrm{best}}^T + O(\sqrt{BT \log N}).$$

Lower Bound

Theorem

For every T, there exists a sequence of length T together with N experts s.t. every learning algorithm with a buffer of size B suffers a regret of $\Omega(\sqrt{BT\log N})$.

Proof idea:

$$\begin{bmatrix} p^t \\ d^t \end{bmatrix} = \left[\begin{pmatrix} 2 \\ 0 \end{pmatrix}^B \begin{pmatrix} \{0,4\} \\ 0 \end{pmatrix}^B \begin{pmatrix} 4 \\ 1 \end{pmatrix}^B \right]^{T'}$$

a) The expert purchases B units in the first phase.

b) The expert purchases B units in the second phase.

- every expert chooses one of the strategies uniformly at random in every round
- cost of experts: N independent random walks of length T' with step length B
- expected minimum of those random walks $2/3T - \Omega(\sqrt{BT\log N})$, expected cost $2/3T$

Summary

- Shrinking Dartboard, which achieves low regret for online buffering
  - Similar regret bound also possible for Follow the Perturbed Leader [Kalai, Vempala, 2005]
- Weighted Fractional achieves low regret also against adaptive adversary
- The regret bounds of the algorithms are tight

Thank you for your attention!
Any questions?