Multiplicative Weights Update (MWU) Algorithm

Setting:

- n experts, T rounds
- In each round, we have to pick an expert $i \in [n]$
- When picking expert i in round t: loss $f_i^t \in [-1, 1]$ (or gain $g_i^t \in [-1, 1]$)

Goal: to be competitive with best expert (in hindsight)

Algorithm:

- Maintains weights w_i^t and probabilities p_i^t for all experts in round t
- Initial weights: $w^1 = (1, \dots, 1)$, parameter $\varepsilon > 0$
- In round t:
  1. $\forall i \in [n] : \Phi^t := \sum_{i=1}^n w_i^t, \; p_i^t := \frac{w_i^t}{\Phi^t}$
  2. $\forall i \in [n] : w_i^{r+1} := w_i^r \cdot (1 - \varepsilon f_i^r) \quad (w_i^r := w_i^r \cdot (1 + \varepsilon g_i^r))$

Loss / Gain / Regret:

- **Total loss/gain:**

  loss $:= \sum_{t=1}^T \langle p^t, f^t \rangle \quad$ (gain $:= \sum_{t=1}^T \langle p^t, g^t \rangle$)

- **Loss/gain for expert i:**

  loss$_i := \sum_{t=1}^T f_i^t \quad$ (gain$_i := \sum_{t=1}^T g_i^t$)

- **Regret:**

  regret$_i :=$ loss $- \cancel{\min_{i \in [n]}}$ loss$_i \quad$ (regret$_i := \cancel{\max_{i \in [n]}}$ gain$_i -$ gain)

regret $:= \max_{i \in [n]}$ regret$_i$

Theorem:

$$\boxed{\forall i \in [n] : \underline{\text{regret}_i} \leq \varepsilon \cdot \sum_{i=1}^T |f_i^t| + \frac{\ln n}{\varepsilon} \leq \varepsilon T + \frac{\ln n}{n}.}$$

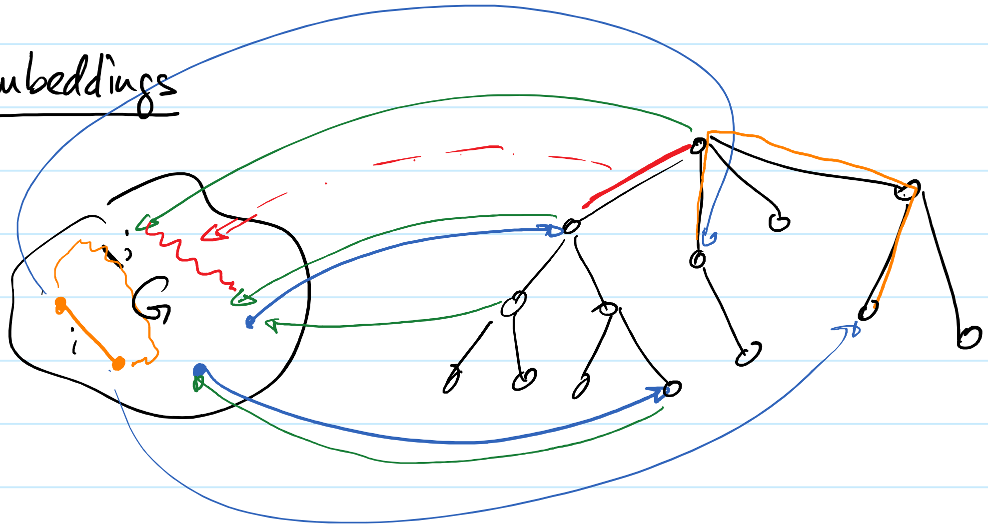
g_i^t

More Applications of Multiplicative Weights

10 May 2019 09:40

Cut-Based Tree Embeddings

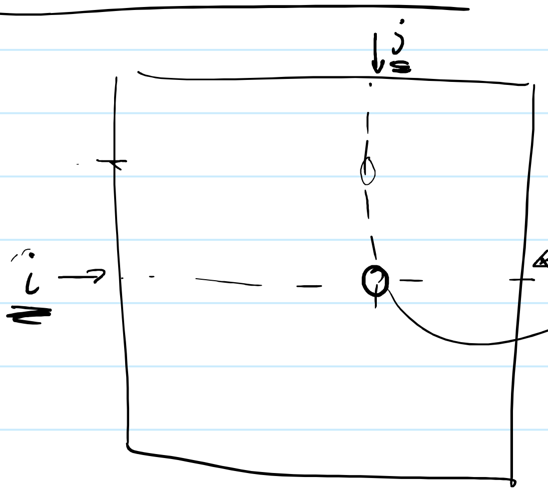
Tree Embedding



Mapping M maps every edge of G to a path in G

multiset of edges

M as an $|E| \times |E|$ matrix:



"characteristic vector of $M(i)$"

M_{ij}: # time path $M(i)$ uses edge j

Assume that
- edge i has length $\ell_i > 0$
- edge i has capacity $c_i > 0$

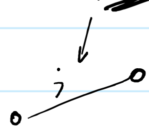
stretch of edge i for mapping M

$$\mathrm{stretch}_M(i) := \frac{\sum_{j \in E} M_{ij} \ell_j}{\ell_i}$$

rel. load of edge j for mapping M

$$\mathrm{rload}_M(j) := \frac{\sum_{i \in E} M_{ij} c_i}{c_j}$$

load when routing c_i units of flow on each edge $i \in E$



M_{ij}: # times edge j has to carry traffic c_i

10 May 2019 09:40

Solving a comm. problem P on G

- Assume there is a solution to P on G with congestion $\leq \underline{\underline{\alpha}}$

$$\max_{e\in E} \frac{\text{traffic}(e)}{c_e}$$

- Mapping that solution by using M incurs load

$$\leq \alpha\, c_e \,\text{rload}_M(e) \quad \text{on edge } e$$

- If M is a mapping to a tree in the described way, optimal sol. on tree incurs at most the same load.

Consider the following game:

EDGE player:

pick edge e of G, goal: maximize $\text{rload}_M(e)$

MAP player:

pick mapping $M \in \mathcal{M}$ (embedding), goal: minimize $\text{rload}_M(e)$

all decomp. trees / all spanning trees

(game value: max. expected rel. load of a best possible embedding M)

MWU Algorithm:

m experts, one for each edge $i\in E$

p^t: distr. on edges, p^1: unif. distr.

<u>assume:</u> EDGE player chooses edge according to distr. p^t

let M_t be the best response of MAP player

$$M_t := \underbrace{\arg\min_{M\in\mathcal{M}} \sum_{i\in E} p_i^t \cdot \text{rload}_M(i)}$$

find M that min. weight avg. rel. load

10 May 2019 09:40

gain expert i in round t:

$$g_i^t := \frac{\mathrm{rload}_{M_t}(i)}{C} \in [-1, 1]$$

C ← normalization factor

$$C \geq \max_{M \in \mathcal{M}, i \in E} \mathrm{rload}_M(i)$$

Assume that for all p, we can find mapping $M \in \mathcal{M}$ s.t

$$\sum_{i \in E} p_i \mathrm{rload}_M(i) \leq \beta \qquad (\text{will see that } \beta = O(\log n))$$

total gain of MWU alg:

$$\mathrm{gain} = \sum_{t=1}^{T} \underbrace{\sum_{i \in E} p_i^t \cdot \frac{\mathrm{rload}_{M_t}(i)}{C}} \leq \frac{\beta T}{C}$$

$$\mathrm{gain} = \mathrm{gain}_i - \mathrm{regret}_i$$

$$\mathrm{gain}_i = \sum_{t=1}^{T} \frac{\mathrm{rload}_{M_t}(i)}{C}$$

$$\mathrm{regret}_i \leq \varepsilon \cdot \sum_{t=1}^{T} |g_i^t| + \frac{\ln m}{\varepsilon} = \overbrace{\underline{\varepsilon \cdot \sum_{t=1}^{T} \frac{\mathrm{rload}_{M_t}(i)}{C}}}^{\leq \varepsilon\beta} + \frac{\ln m}{\varepsilon}$$

In the end, choose unif. distr. on $M_1, M_2, \ldots, M_T$

Expected rel. load on edge i

$$= \underline{\frac{1}{T} \cdot \sum_{t=1}^{T} \mathrm{rload}_{M_t}(i)} = \frac{C}{T} \cdot \mathrm{gain}_i = \frac{C}{T}(\mathrm{gain} + \mathrm{regret}_i)$$

$$\leq \beta + \frac{C}{T}\mathrm{regret}_i \leq (1+2\varepsilon)\beta$$

$$\frac{C}{T}\mathrm{regret}_i \leq \varepsilon \cdot \beta + \frac{\ln m}{\varepsilon} \cdot \frac{C}{T}, \quad \text{choose } \underline{T \geq \frac{C \ln m}{\varepsilon^2 \beta}} \rightarrow \frac{\ln m}{\varepsilon}\frac{C}{T} \leq \varepsilon\beta$$

10 May 2019 09:40

Find a low average rel. load embedding (mapping)

Given: distr. λ_i on edges $i \in E$ $\quad (\lambda_i \geq 0, \ \sum \lambda_i = 1)$

Goal: Find M s.t. $\underbrace{\sum_{j \in E} \lambda_j \cdot \frac{\text{load}_M(j)}{c_j}}_{(*)} \leq \beta$

$$\text{load}_M(j) = \sum_{i \in E} M_{ij} \cdot c_i$$

$$(*) = \sum_{j \in E} \sum_{i \in E} \lambda_j \cdot \frac{c_i}{c_j} M_{ij}$$

$$= \sum_{i \in E} \sum_{j \in E} \lambda_j \frac{c_i}{c_j} M_{ij}$$

define edge length $\ell_i := \frac{\lambda_i}{c_i}$

$$= \sum_{i \in E} \sum_{j \in E} \lambda_i \frac{\ell_j}{\ell_i} M_{ij}$$

$$= \sum_{i \in E} \lambda_i \underbrace{\frac{\sum_{j \in E} M_{ij} \cdot \ell_j}{\ell_i}}_{\text{stretch}_M(i)} \overset{!}{\leq} \beta$$

→ find M s.t. weighted avg. stretch $\leq \beta$

→ for decomp. trees, we know how to do this for $\beta = O(\log n)$

→ for spanning trees, this can be done with

$$\beta = O(\log n \cdot \log\log n \cdot (\log\log\log n)^3)$$