(Online) Multi-Armed Bandit Learning

"online learning with partial feedback"

Model

- k actions, n rounds
- action j's payoff in round i: v_j^i
- in round i:
  - (a) choose action j^i
  - (b) learn payoffs $v_{j^i}^i$
  - (c) obtain payoff $v_{j^i}^i$
- ALG's payoff $= \sum_{i=1}^{n} v_{j^i}^i$

Goal: total payoff close to best in hindsight

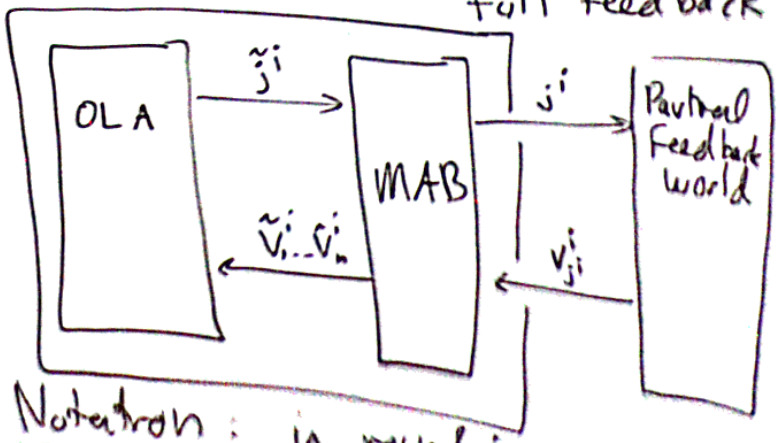
Note identical to online learning except only learn $v_{j^i}^i$ not $v_1^i, \ldots, v_k^i$

Note: if we don't take action j, don't learn if j is good or not

Challenge: "tradeoff of exploration and exploitation"

Reducing MAB to Online Learning

"reduce partial feedback to full feedback"



Notation: in round i

- probabilistic/mixed actions: $\pi^i = (\pi_1^i, \ldots, \pi_k^i)$
  $\pi_j^i = Pr[j^i = j]$
- choose action $j^i \sim \pi^i$
- payoffs $v^i = (v_1^i, \ldots v_k^i)$

- expected payoff:

$$E[v_j^i] = \sum_{j=1}^{k} E[v_j^i \mid j^i = j] P_r[j^i = j]$$

$$= \sum_{j=1}^{k} v_j^i \times \pi_j^i$$

$$= \vec{v}^i \cdot \vec{\pi}^i \quad \text{(vector dot product)}$$

Challenge 1: what to input to OLA?

Idea 1: report "unbiased estimator" of true payoffs.

- if MAB uses probabilities $\vec{\pi}^i = (\pi_1^i, \ldots \pi_k^i)$
- samples $j^i \sim \vec{\pi}^i$
- real payoffs: $\vec{v}^i = (v_1^i, \ldots v_k^i)$ only know v_j^i
- report: $\tilde{\vec{v}}^i = (0, \ldots, 0, \frac{v_j^i}{\pi_j^i}, 0, \ldots, 0)$

Lemma 1: expected report is equal to actual payoff

Proof $E[\tilde{v}_j^i] = E[\tilde{v}_j^i \mid j^i = j] Pr[j^i = j] + E[\tilde{v}_j^i \mid j^i \neq j] Pr[j^i \neq j]$

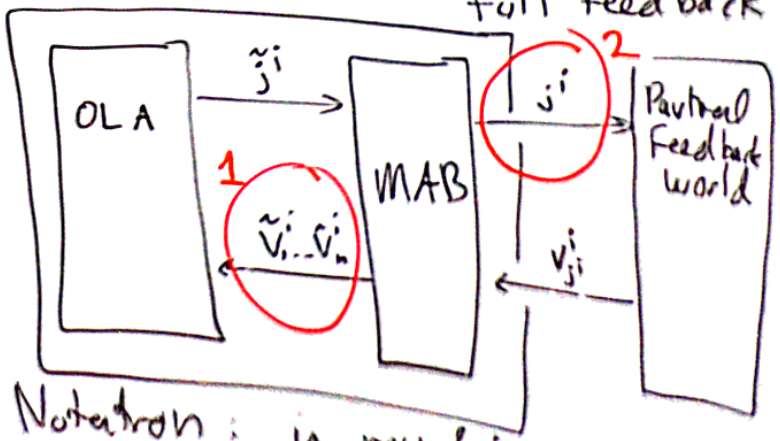
$= \frac{v_j^i}{\pi_j^i} \times \pi_j^i + 0 \times Pr[j^i \neq j]$

$= v_j^i$

Challenge: "tradeoffs of exploration and exploitation"

Reducing MAB to Online Learning

"reduce partial feedback to full feedback"



Notation: in round i

- probabilities of actions: $\vec{\pi}^i = (\pi_1^i, \ldots, \pi_k^i)$, $\pi_j^i = Pr[j^i = j]$
- choose action $j^i \sim \vec{\pi}^i$
- payoffs: $\vec{v}^i = (v_1^i, \ldots v_k^i)$

Note: for actual payoffs $v_j^i \in [0,h]$

- reported payoffs $\tilde{v}_j^i \in \{0, v_j^i/\pi_j^i\}$
- upper bound on reported payoffs is $\hat{h} = \max_{i,j} v_j^i/\pi_j^i$
- if π_j^i is small, $\hat{h}$ is big.
- recall OLA regret $= O(h)\sqrt{\frac{\log k}{n}}$

Challenge 2: keep $\hat{h}$ small.

Idea 2: pick random action with some minimal probability ϵ.

- if OLA recommends $\hat{\pi}^i$
- MAB uses π^i with $\pi_j^i = (1-\epsilon)\hat{\pi}_j^i + \epsilon/k$
- then $\pi_j^i \geq \epsilon/k$.

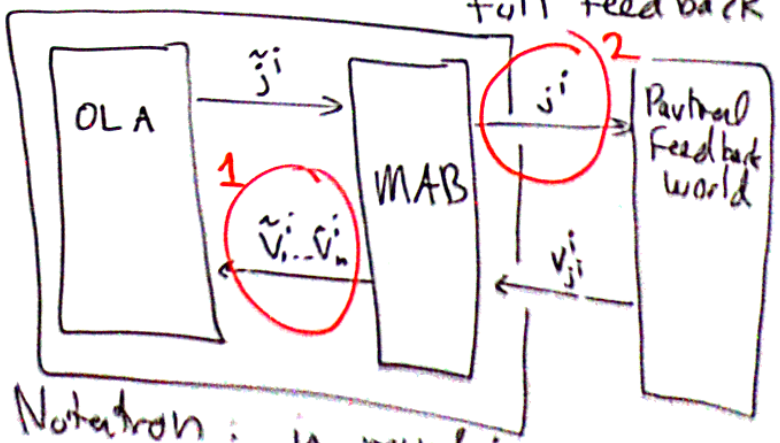
Lemma 2: if $\pi_j^i \geq \epsilon/k$ then $\tilde{v}_j^i \leq \hat{h} = \frac{kh}{\epsilon}$

Proof: $\tilde{v}_j^i \leq v_j^i/\pi_j^i \leq \frac{v_j^i k}{\epsilon} \leq \frac{hk}{\epsilon} = \hat{h}$

Challenge: "tradeoff of exploration and exploitation"

Reducing MAB to Online Learning

"reduce partial feedback to full feedback"



Notation: in round i

- probabilities of actions: $\pi^i = (\pi_1^i, \ldots, \pi_k^i)$, $\pi_j^i = \Pr[j^i = j]$
- choose action $j^i \sim \pi^i$
- payoffs $w^i = (v_1^i, \ldots v_k^i)$

ALG: MAB reduction to OLA

in round i:

1) $\tilde{\pi}^i \leftarrow$ OLA

2) draw $j^i \sim \pi^i$

$$\pi^i_j = (1-\epsilon)\tilde{\pi}^i_j + \epsilon/k$$

3) take action j^i; learn $v^i_{j^i}$

4) report to OLA

$$\hat{v}^i_j = \begin{cases} v^i_j/\pi^i_j & \text{if } j = j^i \\ 0 & \text{otherwise.} \end{cases}$$

Thm: for payoffs in $[0,\tilde{h}]$, if OLA satisfies $E[OLA] \geq (1-\epsilon)OPT(\tilde{v}_{1:T}) - \frac{\tilde{h}}{\epsilon}\ln k$

then for payoffs in $[0,h]$, MAB satisfies

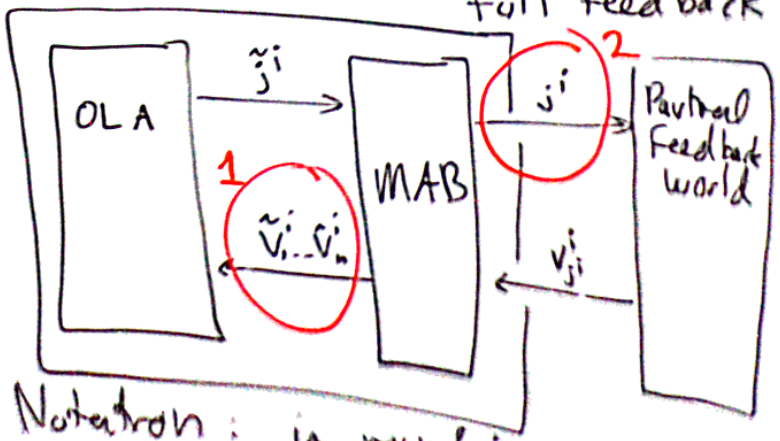
$$E[MAB] \geq (1-2\epsilon)OPT(v_{1:T}) - \frac{hk}{\epsilon^2}\ln k$$

Cor: MAB with Exp Weights has vanishing regret.

Challenge: "tradeoff of exploration and exploitation"

Reducing MAB to Online Learning

"reduce partial feedback to full feedback"



Notation: in round i

- probability vector of alg: $\pi^i = (\pi^i_1, \ldots, \pi^i_k)$
  $\pi^i_j = Pr[j^i = j]$
- choose action $j^i \sim \pi^i$
- payoffs $v^i = (v^i_1, \ldots v^i_k)$

3. combine

$$E[MAB] \geq \underbrace{(1-\epsilon)^2}_{(1-2\epsilon+\epsilon^2)} \underbrace{\sum V_{j^*}^i}_{OPT} - R$$

$$\geq (1-2\epsilon)OPT - R$$

$$= (1-2\epsilon)OPT - \frac{\tilde{h}}{\epsilon}\ln k$$

$$= (1-2\epsilon)OPT - \frac{hk}{\epsilon^2}\ln k$$

Thm: for payoffs in $[0, \tilde{h}]$, if OLA satisfies

$$E[OLA] \geq (1-\epsilon)OPT(\tilde{V}^1) - \frac{\tilde{h}}{\epsilon}\ln k$$

then for payoffs in $[0, h]$, MAB satisfies

$$E[MAB] \geq (1-2\epsilon)OPT(V^1) - \frac{hk}{\epsilon^2}\ln k$$

Cor: MAB with Exp Weights has vanishing regret.

Analysis

"online learning works on unbiased estimators"

0. $R = \frac{\tilde{h}}{\epsilon}\ln k$

$j^* = \text{argmax}_j \sum_i V_j^i$

1. OLA's guarantee on $\tilde{V}^1, \ldots, \tilde{V}^n$

$$OLA = \sum_i \tilde{\pi}^i \cdot \tilde{V}^i \geq (1-\epsilon)\sum_i \tilde{V}_{j^*}^i - R$$

$$E[OLA] = \sum_i E[\tilde{\pi}^i \cdot \tilde{V}^i] \geq (1-\epsilon)\sum_i E[\tilde{V}_{j^*}^i] - R$$

$$= \sum_i E[\tilde{\pi}^i \cdot V^i] \geq (1-\epsilon)\sum_i V_{j^*}^i - R$$

2. MAB's performance on $V^1, \ldots V^n$

$$MAB = \sum_i \pi^i \cdot V^i$$

$$= (1-\epsilon)\sum_i \hat{\pi}^i \cdot V^i + \frac{\epsilon}{K}\sum_{ij} V_j^i$$

$$\geq (1-\epsilon)\sum_i \hat{\pi} \cdot V^i$$