Algorithmic Game Theory

Learning in Games

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(May 4, 2018)

Plan

Online learning and prediction

- single agent learns to select the best action

Learning in normal form games

- the same algorithms used by multiple agents

Learning in extensive form games

- generalizing these ideas to sequential games

DeepStack

Algorithmic Game Theory

Introduction to Online Learning and Prediction

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(April 25, 2017)

Introduction

Online learning and prediction

- learning from data that become available in sequence
- adapting prediction (behavior) after each data point
- optimizing overall precision (not only after all data arrive)

Applications

- investing in best fond
- web advertisements
- selecting the best (e.g., page replacement) algorithm

Introduction

Why do we care about online learning in games?

- repeated play against an unknown opponent
- (repeated) play of an unknown game
- understanding how equilibria may occur in real world
- computationally efficient equilibrum approximation algorithms

Prediction with expert advice

a_1

a_2

a_3

Problem definition

Set of n actions (experts) $A = \{a_1, a_2, \ \dots, a_n\}$

Set of time steps $t = \{1,2, \dots, T\}$

In each step

Decision-maker selects a mixed strategy σ^t

An adversary selects rewards $u^t: A \to [0,1]$ (adaptive vs oblivious)

Action $\mathrm{a}^{\mathrm{t}} \in A$ is selected based on σ^t

The decision-maker receives reward $u^t(a^t)$ (learns the whole u^t)

External Regret

	σ^0	u^0	σ^1	u^1	σ^2	u^2		σ^T	u^T
a_1	0.2	0	0.1	1	0.3	0			
a_2	0.5	0.5	0.4	0.5	0.3	1	…		
a_3	0.3	1	0.5	0	0.4	0			
$\sigma^t \cdot u^t$	$x^0 = 0.55$		$x^1 = 0.3$		$x^2 = 0.3$			x^T	

Goal: play as well as the best expert

- **Immediate regret** at time t for not choosing action i
  - $r^t(i) = u^t(i) - x^t$
- **Cumulative external regret** for playing $\sigma^0, \sigma^1 \ldots \sigma^T$
  - $R^T = max_{i \in A} \sum_{t=0}^{T} r^t(i) = max_{i \in A} \sum_{t=0}^{T} u^t(i) - \sum_{t=0}^{T} x^t$
- **Average external regret** for playing $\sigma^0, \sigma^1 \ldots \sigma^T$
  - $\bar{r}^T = \frac{1}{T} R^T$

Swap Regret

	σ^0	u^0	σ^1	u^1	σ^2	u^2		σ^T	u^T
a_1	0.2	0	0.1	1	0.3	0			
a_2	0.5	0.5	0.4	0.5	0.3	1	…		
a_3	0.3	1	0.5	0	0.4	0			
$\sigma^t \cdot u^t$	$x^0 = 0.55$		$x^1 = 0.3$		$x^2 = 0.3$			x^T	

Goal: minimize regret for not playing a $\delta(a)$ instead of a for some $\delta: A \to A$

Cumulative swap regret for playing $\sigma^0, \sigma^1 \dots \sigma^T$

$$R^T = max_\delta \sum_{t=0}^{T} \sum_{i \in A} \sigma^t(i)(u^t(\delta(i)) - u^t(i))$$

Internal regret

allows switching only all occurrences of a_i by a_j

External ⊂ Swap, Internal ⊂ Swap

No-regret algorithms

An algorithm has **no regret** if for any $u^0, u^1 \ldots u^T$ produces $\sigma^0, \sigma^1 \ldots \sigma^T$ such that $\bar{r}^T \rightarrow 0$ as $T \rightarrow \infty$.

Why not simply to maximize reward?

$$maximize \sum_{t=0}^{T} x^t$$

The adversary may choose $\forall i \in A, \; u^t(i) = 0$ and we have minimal reward regardless of the used algorithm.

Any algorithm has (optimal) 0 regret.

Regret towards best strategy in hindsight

$$R_{best}{}^{T} = \sum_{t=0}^{T} max_{i\in A}\, u^t(i) - \sum_{t=0}^{T} x^t$$

Proposition: There is no algorithm with no regret towards the best sequence of choices.

Proof: Let $A = \{U, D\}$. For an arbitrary sequence of strategies σ^t, choose a reward vector $u^t = (0,1)$ if $\sigma^t(U) \geq \frac{1}{2}$ and $u^t = (1,0)$ otherwise.

The cumulative reward of the algorithm $\sum_{t=0}^{T} x^t \leq \frac{T}{2}$, while the best strategy in hindsight has reward $\sum_{t=0}^{T} max_{i\in A}\, u^t(i) = T$. Therefore

$$R_{best}{}^{T} \geq \frac{T}{2} \text{ and } \bar{r}_{best}^{T} \rightarrow z \geq \frac{1}{2}$$

Regret of deterministic algorithms

Proposition: There is no deterministic no-external-regret algorithm.

Proof: We assume that the adversary selects rewards u^t knowing strategy σ^t. (For example, it can simulate the deterministic algorithm from the beginning.) Therefore, with $n = 2$, he can always give reward 0 for the selected action and 1 for the other action. One of the action got reward 1 at least $T/2$ times, therefore $\bar{r}^t \geq \frac{1}{2}$.

Lower bound on external regret

Theorem: No (randomized) algorithm over n actions has expected external regret vanishing faster than $\Theta(\sqrt{\ln(n)/T})$.

Proof sketch: Assume n=2. Consider an adversary that, independently on each step t, chooses uniformly at random between the cost vectors (1, 0) and (0, 1) regardless of the decision-making algorithm. The cumulative expected reward is exactly $T/2$. In hindsight, however, with constant probability one of the two fixed actions has cumulative reward $\mathrm{T}/2 + \Theta(\sqrt{T})$. The reason is that T fair coin flips have standard deviation $\Theta(\sqrt{T})$.

Lower bound on external regret

Theorem: There exist no-regret algorithms with expected external regret $O(\sqrt{\ln(n)\,/T})$.

Proof: We will show Randomized Weighted Majority algorithm.

Corollary: There exists a decision-making algorithm that, for every $\epsilon > 0$, has expected regret less than ϵ after $O(\ln(n)/\epsilon^2)$ iterations.

Randomized Weighted Majority

Aka Hedge or multiplicative weights (MW) algorithm. It is easier to analyze in costs $c(i) = (1 - u(i))$. The algorithm maintains weights $w(i)$ for each action $i \in A$.

Initialize $w^1(i) = 1$ for every $i \in A$

For each time $t = 1,2, \dots, T$

Let $W^t = \sum_{i \in A} w^t(i)$ and play $\sigma^t(i) = w^t(i)/W^t$

Given costs c^t, set $w^{t+1}(i) = w^t(i)(1-\gamma)^{c^t(i)}$ for each $i \in A$

(Equivalently $w^{t+1}(i) = w^t(i)e^{-\eta c^t(i)}$ for $\eta = -\ln(1-\gamma)$)

Hedge Regret Bound

Theorem: Expected external regret of Hedge is $\bar{r}^T < 2\sqrt{ln(n)/T}$

Proof: W.L.O.G. we assume oblivious adversary.

Let $OPT = \min_{i \in A} \sum_{t=1}^{T} c^t(i)$ be the cost for optimal action i^* and

$\nu^t = \sum_{i \in A} \sigma^t(i) c^t(i) = \sum_{i \in A} \frac{w^t(i)}{W^t} c^t(i)$ be the algorithms cost at t.

$$W^T \geq w^T(i^*) = w^1(i^*) \prod_{t=1}^{T} (1-\gamma)^{c^t(i^*)} = (1-\gamma)^{OPT}$$

$$W^{t+1} = \sum_{i \in A} w^{t+1}(i) = \sum_{i \in A} w^t(i)(1-\gamma)^{c^t(i)}$$

$$\leq \sum_{i \in A} w^t(i)\left(1-\gamma c^t(i)\right) = W^t(1-\gamma\nu^t)$$

$$(1-\gamma)^{OPT} \leq W^T \leq W^1 \prod_{t=1}^{T} (1-\gamma\nu^t)$$

$$OPT \ln(1-\gamma) \leq \ln n + \sum_{t=1}^{T} \ln(1-\gamma\nu^t)$$

$$\ldots \sum_t^T \nu^t \leq OPT + \gamma T + \frac{\ln n}{\gamma} \quad \Rightarrow \frac{1}{\mathrm{T}} \sum_t^T \nu^t \leq \frac{OPT}{T} + 2\sqrt{\frac{\ln n}{T}}$$

Hedge Implementation Tricks

Weights $w^t(i)$ may quickly become very small.

We can instead store cumulative cost $C^t(i) = \sum_{\tau=1}^{t} c^\tau(i)$.

Than $w^t(i) = (1-\gamma)^{C^t(i)}$
and $\sigma^t(i) = \frac{w^t(i)}{\sum_{j \in A} w^t(j)} = \frac{1}{1+\sum_{j \neq i}(1-\gamma)^{(C^t(j)-C^t(i))}}$

We can see that $\sigma^t(i)$ depends only on differences between $C^t(i)$, therefore we can use $C^t(i) - K$ for any constant K.

Regret Matching

The algorithm maintains cummulative regrets $\mathrm{R}(i)$ for each action $i \in A$.

Initialize $R^1(i) = 0$ for every $i \in A$

For each time $t = 1,2, \dots, T$

Let $S^t = \sum_{i \in A} \max(0, R^t(i))$ and play $\sigma^t(i) = \max(0, R^t(i))/S^t$

Given rewards u^t, for each $i \in A$ set

$$R^{t+1}(i) = R^t(i) + r^t(i) = R^t(i) + (u^t(i) - \sum_{j \in A} \sigma^t(j) u^t(j))$$

Regret Matching+

The algorithm maintains cumulative regrets-like values $\mathrm{Q}(i)$ for each action $i \in A$.

Initialize $Q^1(i) = 0$ for every $i \in A$

For each time $t = 1,2, \dots, T$

Play $\sigma^t(i) = Q^t(i) / \sum_{j \in A} Q^t(j)$

Given rewards u^t, for each $i \in A$ set

$$Q^{t+1}(i) = \max(0, Q^t(i) + r^t(i)) = \max(0,\ u^t(i) - \sum_{j \in A} \sigma^t(j) u^t(j))$$

RM+ Regret Bound

Lemma: Regret-like values $Q^t(i)$ are an upper bound on $R^t(i)$.

Proof: $Q^{t+1}(i) - Q^t(i) = \max\left(0, Q^t(i) + r^t(i)\right) - Q^t(i)$
$\geq Q^t(i) + r^t(i) - Q^t(i) = r^t(i)$

Lemma: For any i and value functions $Q^T(i) \leq \sqrt{nT}$.

Proof: $\left(\max_{i \in A} Q^T(i)\right)^2 = \max_{i \in A} Q^T(i)^2 \leq \sum_{i \in A} Q^T(i)^2 =$

$= \sum_{i \in A} \max(0, Q^{T-1}(i) + u^T(i) - \sum_{j \in A} \sigma^T(j) u^T(j))^2$

$\ldots \leq \sum_i Q^{T-1}(i)^2 + n$

By induction $Q^T(i)^2 \leq nT$.

Adversarial Multi-Armed Bandit Problem

a_1

a_2 ___ 0.5 ___

a_3 ___ 0

Problem definition

- Set of n actions (experts) $A = \{a_1, a_2, \ldots, a_n\}$
- Set of time steps $t = \{1,2, \ldots, T\}$
- In each step
  - Decision-maker selects a mixed strategy σ^t
  - An adversary selects rewards $u^t: A \rightarrow [0,1]$ (adaptive vs oblivious)
  - Action $\mathrm{a}^t \in A$ is selected based on σ^t
  - The decision-maker receives reward $u^t(a^t)$ (learns **only** $u^t(a^t)$)

Adversarial MAB

Goal is to minimize regret as before.

The problem is harder than prediction with expert advice

- No deterministic strategy has no regret
- No algorithm has regret below $\Theta(\sqrt{\ln(n)/T})$

Importance Sampling Trick

	σ^0	u^0	σ^1	u^1	σ^2	u^2		σ^T	u^T
a_1	0.2	0	**0.1**	1	0.3	0			
a_2	**0.5**	0.5	0.4	0.5	0.3	1	…		
a_3	0.3	1	0.5	0	**0.4**	0			

How to estimate $U^T(i) = \sum_{t=1}^{T} u^t(i)$ from limited observations?

After choosing i^t, update $\widetilde{U}^t(i) \mathrel{+}= \frac{u^t(i)}{\sigma^t(i)}$ and $\widetilde{U}^t(j) \mathrel{+}= 0$ for $j \neq i$.

$$\mathbf{E}\widetilde{U}^T(i) = \sum_{t=1}^{T} \sigma^t(i) \frac{u^t(i)}{\sigma^t(i)} + \left(1 - \sigma^t(i)\right)0 = \sum_{t=1}^{T} u^t(i) = U^T(i)$$

Exp3

Exponential weights for Exploration and Exploitation.

It is easier to analyze in costs $c(i) = (1 - u(i))$. The algorithm maintains estimates of cumulative loss $\mathrm{C}(i)$ for each action $i \in A$.

Initialize $C^1(i) = 0$ for every $i \in A$

For each time $t = 1,2, \dots, T$

Let $\sigma^t(i) = (1-\gamma)^{C^t(i)} / \sum_{j \in A} (1-\gamma)^{C^t(j)}$

Play action i^t from distribution σ^t, receive cost $c^t(i^t)$

Update $C^t(i^t) += c^t(i^t)/\sigma^t(i^t)$

Expected Regret and Pseudo-regret

Expected external regret

$$\mathbf{E}R^T = \mathbf{E}\max_{b\in A}\left(\sum_{t=1}^{T} u^t(b) - u^t(i^t)\right)$$

Pseudo-regret

$$\bar{R}^T = \max_{b\in A}\mathbf{E}\sum_{t=1}^{T} u^t(b) - \mathbf{E}\sum_{t=1}^{T} u^t(i^t)$$

Observation: $\bar{R}^T \leq \mathbf{E}R^T$

Exp3 Regret Bounds

Theorem: For Exp3 run with a suitable γ holds $\bar{R}^T \leq \sqrt{2Tn \ln n}$.

Exp3.P

Initialize $G^1(i) = 0$ for every $i \in A$

For each time $t = 1,2, \dots, T$

Let $\sigma^t(i) = (1-\alpha)\frac{(1-\gamma)^{G^t(i)}}{\sum_{j \in A}(1-\gamma)^{G^t(j)}} + \frac{\alpha}{n}$

Play action i^t from distribution σ^t, receive reward $= u^t(i^t)$

Update $G^t(i^t) \mathrel{+}= \frac{u^t(i^t)+\beta}{\sigma^t(i^t)}$ and $G^t(j) \mathrel{+}= \frac{\beta}{\sigma^t(j)}$ for $j \neq i^t$

Exp3.P Regret Bound

Theorem: For any $\delta \in (0,1)$ there are γ, α, β such that with probability at least $(1 - \delta)$,

$$R^T \leq 5.15\sqrt{Tn \ln \frac{n}{\delta}}$$

Summary

It is possible to perform as well as taking the best action in the limit very tiny amount of information about the problem.

References

Blum, Avrim, and Yishay Mansour. "From external to internal regret." Journal of Machine Learning Research 8.Jun (2007): 1307-1324.

T. Roughgarden, “Lecture Notes: Algorithmic Game Theory,” tech. rep., Stanford, 2013.

Tammelin, Oskari, Neil Burch, Michael Johanson, and Michael Bowling. "Solving Heads-Up Limit Texas Hold'em." In Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

Bubeck, Sébastien, and Nicolo Cesa-Bianchi. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems." Foundations and Trends in Machine Learning 5.1 (2012): 1-122.