Fast Convergence of Regularized Learning in Games

Malte Schledjewski

Saarbrücken Graduate School of Computer Science

2016

The paper

Fast Convergence of Regularized Learning in Games

- Vasilis Syrgkanis, Microsoft Research
- Alekh Agarwal, Microsoft Research
- Haipeng Luo, Princeton University
- Robert E. Schapire, Microsoft Research

published 2015

Online learning

Online learning

General model

You try to map inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.

Online learning

General model

You try to map inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.

For each time step t:

Online learning

General model

You try to map inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.

For each time step t:

- You receive x_t.

Online learning

General model

You try to map inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.

For each time step t:

- You receive x_t.
- You predict the output as p_t.

Online learning

General model

You try to map inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.

For each time step t:

- You receive x_t.
- You predict the output as p_t.
- You receive the correct output y_t.

Online learning

General model

You try to map inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.

For each time step t:

- You receive x_t.
- You predict the output as p_t.
- You receive the correct output y_t.
- You suffer some loss $\ell(y_t, p_t)$.

Online learning

General model

You try to map inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.

For each time step t:

- You receive x_t.
- You predict the output as p_t.
- You receive the correct output y_t.
- You suffer some loss $\ell(y_t, p_t)$.
- You update your model.

Online learning

General model

You try to map inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.
For each time step t:

- You receive x_t.
- You predict the output as p_t.
- You receive the correct output y_t.
- You suffer some loss $\ell(y_t, p_t)$.
- You update your model.

Your goal is minimal accumulated loss.

Online learning

Prediction with Expert Advice

Consider d experts giving you advices.

Choose the best advice.

For each time step t:

Online learning

Prediction with Expert Advice

Consider d experts giving you advices.

Choose the best advice.

For each time step t:

- You receive x_t, a vector of d advices.

Online learning

Prediction with Expert Advice

Consider d experts giving you advices.

Choose the best advice.

For each time step t:

- You receive x_t, a vector of d advices.
- You chose expert p_t and follow his advice.

Online learning

Prediction with Expert Advice

Consider d experts giving you advices.
Choose the best advice.
For each time step t:

- You receive x_t, a vector of d advices.
- You chose expert p_t and follow his advice.
- You receive y_t, the vector of costs for following each of the advices.

Online learning

Prediction with Expert Advice

Consider d experts giving you advices.
Choose the best advice.
For each time step t:

- You receive x_t, a vector of d advices.
- You chose expert p_t and follow his advice.
- You receive y_t, the vector of costs for following each of the advices.
- You suffer some loss $\ell(y_t, p_t) = y_{t,p_t}$.

Online learning

Prediction with Expert Advice

Consider d experts giving you advices.
Choose the best advice.
For each time step t:

- You receive x_t, a vector of d advices.
- You chose expert p_t and follow his advice.
- You receive y_t, the vector of costs for following each of the advices.
- You suffer some loss $\ell(y_t, p_t) = y_{t,p_t}$.
- You update your model.

Regret

How do you know how good you are?

Regret

How do you know how good you are?
Compare yourself to the experts.

Regret

How do you know how good you are?
Compare yourself to the experts.

In each round there is an expert with minimal loss so far.
This is the leading expert.

Regret

How do you know how good you are?
Compare yourself to the experts.

In each round there is an expert with minimal loss so far.
This is the leading expert.

Regret

$r(T) :=$ (your cumulated loss) - (the leader's cumulated loss)

Regret

How do you know how good you are?
Compare yourself to the experts.

In each round there is an expert with minimal loss so far.
This is the leading expert.

Regret

$r(T)$:= (your cumulated loss) - (the leader's cumulated loss)

No-regret algorithm

A no-regret algorithm always achieves regret that is sublinear in T.

Follow the Leader

Follow the Leader

Always trust the currently leading expert with his advice for the next round.

Follow the Leader

Follow the Leader

Always trust the currently leading expert with his advice for the next round.

Worst case regret is not sublinear

Example:
Binary classification: $y \in \{A, B\}$
Two experts: one always predicts A, the other one always B
Your loss is 0 if you predict right or 1 if you predict wrong.

Follow the Leader

Follow the Leader

Always trust the currently leading expert with his advice for the next round.

Worst case regret is not sublinear

Example:
Binary classification: $y \in \{A, B\}$
Two experts: one always predicts A, the other one always B
Your loss is 0 if you predict right or 1 if you predict wrong.

In the worst case your prediction is always false.
Your regret is at least $T/2$.

Deterministic or not?

An adversaries perspective

Finite amount of experts and deterministic behaviour allow easy construction of worst case scenario.
Always make the algorithm's prediction false.

Deterministic or not?

An adversaries perspective

Finite amount of experts and deterministic behaviour allow easy construction of worst case scenario.
Always make the algorithm's prediction false.

Idea: randomness

Instead of picking one expert just give the probabilities of choosing the experts.
The adversary is not allowed to know the draw.
We then try to minimize accumulated expected loss.

Stability is also important

Follow the Leader – regret bound by cheating

Let $f_1, \ldots, f_T$ be the sequence of loss functions and $w_1, \ldots, w_T$ be the probabilities determined by *Follow the Leader*, and w^* the leading probabilities.

$$r(T) = \sum_{t=1}^{T} (f_t(w_t) - f_t(w^*)) \leq \sum_{t=1}^{T} \left(\underbrace{f_t(w_t) - f_t(w_{t+1})} \right)$$

Stability is also important

Follow the Leader – regret bound by cheating

Let $f_1, \ldots, f_T$ be the sequence of loss functions and $w_1, \ldots, w_T$ be the probabilities determined by *Follow the Leader*, and w^* the leading probabilities.

$$r(T) = \sum_{t=1}^{T} (f_t(w_t) - f_t(w^*)) \leq \sum_{t=1}^{T} \left(\underbrace{f_t(w_t) - f_t(w_{t+1})}_{\text{stability}} \right)$$

Stability is also important

Follow the Leader – regret bound by cheating

Let $f_1, \dots, f_T$ be the sequence of loss functions and $w_1, \dots, w_T$ be the probabilities determined by *Follow the Leader*, and w^* the leading probabilities.

$$r(T) = \sum_{t=1}^{T} (f_t(w_t) - f_t(w^*)) \leq \sum_{t=1}^{T} \left(\underbrace{f_t(w_t) - f_t(w_{t+1})}_{\text{stability}} \right)$$

Follow the Regularized Leader

$$w_T = \operatorname*{argmin}_{w \in \Delta} \left(\sum_{t=1}^{T-1} f_t(w) \right) + \frac{1}{\eta} \mathcal{R}(w)$$

Stability is also important

Follow the Leader – regret bound by cheating

Let $f_1, \ldots, f_T$ be the sequence of loss functions and $w_1, \ldots, w_T$ be the probabilities determined by *Follow the Leader*, and w^* the leading probabilities.

$$r(T) = \sum_{t=1}^{T} (f_t(w_t) - f_t(w^*)) \leq \sum_{t=1}^{T} \left(\underbrace{f_t(w_t) - f_t(w_{t+1})}_{\text{stability}} \right)$$

Follow the Regularized Leader with entropic regularizer

$$w_T = \underset{w \in \Delta}{\operatorname{argmin}} \left(\sum_{t=1}^{T-1} f_t(w) \right) + \frac{1}{\eta} \sum_{i=1}^{d} w_i log(w_i)$$

Online learning summary

- Expert Advice Framework

Online learning summary

- Expert Advice Framework
- Regret

Online learning summary

- Expert Advice Framework
- Regret
- Randomness + stability

Online learning summary

- Expert Advice Framework
- Regret
- Randomness + stability

Follow the Regularized Leader

Follow the Regularized Leader is a no-regret algorithm with $r(T) \in O(\sqrt{T})$.

Games

Matching pennies

$A \diagdown B$	Heads	Tails
Heads	$1 \diagdown -1$	$-1 \diagdown 1$
Tails	$-1 \diagdown 1$	$1 \diagdown -1$

Matching pennies

$A \diagdown B$	Heads	Tails
Heads	$1 \diagdown -1$	$-1 \diagdown 1$
Tails	$-1 \diagdown 1$	$1 \diagdown -1$

Each player has to chose one of the possible strategies
$S = \{\text{Heads}, \text{Tails}\}$.

Matching pennies

$A \diagdown B$	Heads	Tails
Heads	$1 \diagdown -1$	$-1 \diagdown 1$
Tails	$-1 \diagdown 1$	$1 \diagdown -1$

Each player has to chose one of the possible strategies $S = \{\text{Heads}, \text{Tails}\}$.
In each round the players ~~suffer loss~~ gain utility.

Matching pennies

$A \diagdown B$	Heads	Tails
Heads	$1 \diagdown -1$	$-1 \diagdown 1$
Tails	$-1 \diagdown 1$	$1 \diagdown -1$

Each player has to chose one of the possible strategies $S = \{\text{Heads}, \text{Tails}\}$.
In each round the players ~~suffer loss~~ gain utility.

Each player wants to maximize his accumulated utility.

Game

For a game G of n players:

Each player i has

- a finite strategy space S_i and a
- utility function $u_i : S_1 \times \ldots \times S_n \to [0, 1]$.

Game

For a game G of n players:
Each player i has

- a finite strategy space S_i and a
- utility function $u_i : S_1 \times \ldots \times S_n \to [0, 1]$.

In each round t:
The player chooses a ~~strategy~~ mixed strategy $\mathbf{w}_i^t$ to play.

Game

For a game G of n players:
Each player i has

- a finite strategy space S_i and a
- utility function $u_i : S_1 \times \ldots \times S_n \to [0, 1]$.

In each round t:
The player chooses a ~~strategy~~ mixed strategy $\mathbf{w}_i^t$ to play.
Then the player receives $\mathbf{u}_i^t$, the expected utility for each of his strategies x: $\mathbf{u}_i^t = (u_{i,x}^t)_{x \in S_i}$ with $u_{i,x}^t = \mathbb{E}_{s_{-i} \sim w_{-i}^t} [u_i(x, s_{-i})]$

Game

For a game G of n players:
Each player i has

- a finite strategy space S_i and a
- utility function $u_i : S_1 \times \ldots \times S_n \to [0, 1]$.

In each round t:
The player chooses a ~~strategy~~ mixed strategy $\mathbf{w}_i^t$ to play.
Then the player receives $\mathbf{u}_i^t$, the expected utility for each of his strategies x: $\mathbf{u}_i^t = (u_{i,x}^t)_{x \in S_i}$ with $u_{i,x}^t = \mathbb{E}_{s_{-i} \sim w_{-i}^t}[u_i(x, s_{-i})]$
The expected utility for a player i in iteration t is therefore $\langle w_i^t, u_i^t \rangle$.

Playing under nice conditions

Nice opponents

Assume all players to use no-regret algorithms.

Nice opponents

Assume all players to use no-regret algorithms.

For two-player zero-sum games each player's average regret converges at the rate of $O(1/T)$ instead of $O(1/\sqrt{T})$.

Nice opponents

Assume all players to use no-regret algorithms.

For two-player zero-sum games each player's average regret converges at the rate of $O(1/T)$ instead of $O(1/\sqrt{T})$.

Can this be generalized?

RVU property

RVU – Regret bounded by Variation in Utilities

A vanishing regret algorithm has the RVU property with parameters $\alpha > 0$ and $0 < \beta \leq \gamma$ if for any sequence of utilities $\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^T$ the regret is bounded as

$$r(T) \leq \alpha + \beta \sum_{t=1}^{T} \max_{x \in S_i} \left| u_{i,x}^t - u_{i,x}^{t-1} \right|_1^2 - \gamma \sum_{t=1}^{T} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|_1^2$$

All players' average regret

Fast convergence of all players' average regret

Suppose that the algorithm of each player i satisfies the RVU property with parameters α, β and γ such that $\beta \leq \gamma/(n-1)^2$. Then $\sum_{i \in N} r_i(T) \leq \alpha n$ and therefore all players' average regret converges at a rate of $O(1/T)$.

All players' average regret

Fast convergence of all players' average regret

Suppose that the algorithm of each player i satisfies the RVU property with parameters α, β and γ such that $\beta \leq \gamma/(n-1)^2$. Then $\sum_{i \in N} r_i(T) \leq \alpha n$ and therefore all players' average regret converges at a rate of $O(1/T)$.

$$\sum_{i \in N} r_i(T) \leq \sum_{i \in N} \left(\alpha + \beta \sum_{t=1}^{T} \max_{x \in S_i} \left| u_{i,x}^t - u_{i,x}^{t-1} \right|^2 - \gamma \sum_{t=1}^{T} \left\| \mathbf{w}_i^t - \mathbf{w}_i^{t-1} \right\|_1^2 \right)$$

All players' average regret

Fast convergence of all players' average regret

Suppose that the algorithm of each player i satisfies the $\texttt{RVU}$ property with parameters α, β and γ such that $\beta \leq \gamma/(n-1)^2$. Then $\sum_{i \in N} r_i(T) \leq \alpha n$ and therefore all players' average regret converges at a rate of $O(1/T)$.

$$\sum_{i \in N} r_i(T) \leq \sum_{i \in N} \left(\alpha + \beta \sum_{t=1}^{T} \max_{x \in S_i} \left| u_{i,x}^t - u_{i,x}^{t-1} \right|^2 - \gamma \sum_{t=1}^{T} \left\| \mathbf{w}_i^t - \mathbf{w}_i^{t-1} \right\|_1^2 \right)$$

$$= \alpha n + \sum_{t=1}^{T} \left(\beta \sum_{i \in N} \max_{x \in S_i} \left| u_{i,x}^t - u_{i,x}^{t-1} \right|^2 - \gamma \sum_{i \in N} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|_1^2 \right)$$

All players' average regret

Fast convergence of all players' average regret

Suppose that the algorithm of each player i satisfies the RVU property with parameters α, β and γ such that $\beta \leq \gamma/(n-1)^2$. Then $\sum_{i \in N} r_i(T) \leq \alpha n$ and therefore all players' average regret converges at a rate of $O(1/T)$.

$$\sum_{i \in N} r_i(T) \leq \sum_{i \in N} \left(\alpha + \beta \sum_{t=1}^{T} \max_{x \in S_i} \left| u_{i,x}^{t} - u_{i,x}^{t-1} \right|^2 - \gamma \sum_{t=1}^{T} \left\| \mathbf{w}_i^t - \mathbf{w}_i^{t-1} \right\|_1^2 \right)$$

$$= \alpha n + \sum_{t=1}^{T} \left(\beta \sum_{i \in N} \max_{x \in S_i} \left| u_{i,x}^{t} - u_{i,x}^{t-1} \right|^2 - \gamma \sum_{i \in N} \left\| \mathbf{w}_i^t - \mathbf{w}_i^{t-1} \right\|_1^2 \right)$$

Proof - intermediate step

$$\max_{x \in S_i} \left| u_{i,x}^{t} - u_{i,x}^{t-1} \right|$$

Proof - intermediate step

$$\max_{x \in S_i} \left| u_{i,x}^{t} - u_{i,x}^{t-1} \right|$$

$$= \max_{x \in S_i} \left| \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^{t}} [u_i(x, \mathbf{s}_{-i})] - \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^{t-1}} [u_i(x, \mathbf{s}_{-i})] \right|$$

Proof - intermediate step

$$\max_{x \in S_i} \left| u_{i,x}^{t} - u_{i,x}^{t-1} \right|$$

$$= \max_{x \in S_i} \left| \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^{t}} [u_i(x, \mathbf{s}_{-i})] - \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^{t-1}} [u_i(x, \mathbf{s}_{-i})] \right|$$

$$= \max_{x \in S_i} \left| \sum_{\tilde{s} \in \mathbf{s}_{-i}} u_i(x, \tilde{s}) \left(\mathrm{Prob}^{t}(\tilde{s}) - \mathrm{Prob}^{t-1}(\tilde{s}) \right) \right|$$

Proof - intermediate step

$$\max_{x \in S_i} \left| u_{i,x}^{t} - u_{i,x}^{t-1} \right|$$

$$= \max_{x \in S_i} \left| \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^{t}} [u_i(x, \mathbf{s}_{-i})] - \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^{t-1}} [u_i(x, \mathbf{s}_{-i})] \right|$$

$$= \max_{x \in S_i} \left| \sum_{\tilde{s} \in \mathbf{s}_{-i}} u_i(x, \tilde{s}) \left(\mathrm{Prob}^{t}(\tilde{s}) - \mathrm{Prob}^{t-1}(\tilde{s}) \right) \right|$$

$$\leq \max_{x \in S_i} \sum_{\tilde{s} \in \mathbf{s}_{-i}} u_i(x, \tilde{s}) \left| (\mathrm{Prob}^{t}(\tilde{s}) - \mathrm{Prob}^{t-1}(\tilde{s})) \right|$$

Proof - intermediate step

$$\max_{x \in S_i} \left| u_{i,x}^t - u_{i,x}^{t-1} \right|$$

$$= \max_{x \in S_i} \left| \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^t} [u_i(x, \mathbf{s}_{-i})] - \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^{t-1}} [u_i(x, \mathbf{s}_{-i})] \right|$$

$$= \max_{x \in S_i} \left| \sum_{\tilde{s} \in \mathbf{s}_{-i}} u_i(x, \tilde{s}) \left(\mathrm{Prob}^t(\tilde{s}) - \mathrm{Prob}^{t-1}(\tilde{s}) \right) \right|$$

$$\leq \max_{x \in S_i} \sum_{\tilde{s} \in \mathbf{s}_{-i}} u_i(x, \tilde{s}) \left| (\mathrm{Prob}^t(\tilde{s}) - \mathrm{Prob}^{t-1}(\tilde{s})) \right|$$

$$\leq \sum_{\tilde{s} \in \mathbf{s}_{-i}} \left| \prod_{j \neq i} w_{j,\tilde{s}_j}^t - \prod_{j \neq i} w_{j,\tilde{s}_j}^{t-1} \right|$$

Proof - intermediate step

$$\max_{x \in S_i} \left| u_{i,x}^t - u_{i,x}^{t-1} \right|$$

$$= \max_{x \in S_i} \left| \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^t} [u_i(x, \mathbf{s}_{-i})] - \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{w}_{-i}^{t-1}} [u_i(x, \mathbf{s}_{-i})] \right|$$

$$= \max_{x \in S_i} \left| \sum_{\tilde{s} \in \mathbf{s}_{-i}} u_i(x, \tilde{s}) \left(\mathrm{Prob}^t(\tilde{s}) - \mathrm{Prob}^{t-1}(\tilde{s}) \right) \right|$$

$$\leq \max_{x \in S_i} \sum_{\tilde{s} \in \mathbf{s}_{-i}} u_i(x, \tilde{s}) \left| (\mathrm{Prob}^t(\tilde{s}) - \mathrm{Prob}^{t-1}(\tilde{s})) \right|$$

$$\leq \sum_{\tilde{s} \in \mathbf{s}_{-i}} \left| \prod_{j \neq i} w_{j,\tilde{s}_j}^t - \prod_{j \neq i} w_{j,\tilde{s}_j}^{t-1} \right| \leq \sum_{j \neq i} \| \mathbf{w}_j^t - \mathbf{w}_j^{t-1} \|$$

Proof - intermediate step

$$
\begin{aligned}
&\max_{x\in S_i}\left|u_{i,x}^{t}-u_{i,x}^{t-1}\right| \\
&=\max_{x\in S_i}\left|\mathbb{E}_{\mathbf{s}_{-i}\sim\mathbf{w}_{-i}^{t}}[u_i(x,\mathbf{s}_{-i})]-\mathbb{E}_{\mathbf{s}_{-i}\sim\mathbf{w}_{-i}^{t-1}}[u_i(x,\mathbf{s}_{-i})]\right| \\
&=\max_{x\in S_i}\left|\sum_{\tilde{s}\in\mathbf{s}_{-i}}u_i(x,\tilde{s})\left(\mathrm{Prob}^{t}(\tilde{s})-\mathrm{Prob}^{t-1}(\tilde{s})\right)\right| \\
&\leq\max_{x\in S_i}\sum_{\tilde{s}\in\mathbf{s}_{-i}}u_i(x,\tilde{s})\left|(\mathrm{Prob}^{t}(\tilde{s})-\mathrm{Prob}^{t-1}(\tilde{s}))\right| \\
&\leq\sum_{\tilde{s}\in\mathbf{s}_{-i}}\left|\prod_{j\neq i}w_{j,\tilde{s}_j}^{t}-\prod_{j\neq i}w_{j,\tilde{s}_j}^{t-1}\right|\leq\sum_{j\neq i}\|\mathbf{w}_j^{t}-\mathbf{w}_j^{t-1}\| \\
&\Rightarrow\sum_{i\in N}\max_{x\in S_i}\left|u_{i,x}^{t}-u_{i,x}^{t-1}\right|^2\leq(n-1)^2\sum_{i\in N}\left\|\mathbf{w}_i^{t}-\mathbf{w}_i^{t-1}\right\|_1^2
\end{aligned}
$$

Proof - continued

$$\sum_{i \in N} r_i(T) \leq \alpha n + \sum_{t=1}^{T} \left(\beta \sum_{i \in N} \max_{x \in S_i} \left| u_{i,x}^{t} - u_{i,x}^{t-1} \right|^2 - \gamma \sum_{i \in N} \left\| \mathbf{w}^{t} - \mathbf{w}^{t-1} \right\|_1^2 \right)$$

Proof - continued

$$\sum_{i \in N} r_i(T) \leq \alpha n + \sum_{t=1}^{T} \left(\beta \sum_{i \in N} \max_{x \in S_i} \left| u_{i,x}^t - u_{i,x}^{t-1} \right|^2 - \gamma \sum_{i \in N} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|_1^2 \right)$$

$$= \alpha n + \sum_{t=1}^{T} \left(\beta (n-1)^2 \sum_{i \in N} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|_1^2 - \gamma \sum_{i \in N} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|_1^2 \right)$$

Proof - continued

$$
\begin{aligned}
\sum_{i\in N} r_i(T) \leq{}& \alpha n + \sum_{t=1}^{T}\left(\beta \sum_{i\in N}\max_{x\in S_i}\left|u_{i,x}^{t} - u_{i,x}^{t-1}\right|^2 - \gamma\sum_{i\in N}\left\|\mathbf{w}^{t} - \mathbf{w}^{t-1}\right\|_1^2\right)\\
={}&\alpha n + \sum_{t=1}^{T}\left(\beta(n-1)^2\sum_{i\in N}\left\|\mathbf{w}^{t} - \mathbf{w}^{t-1}\right\|_1^2 - \gamma\sum_{i\in N}\left\|\mathbf{w}^{t} - \mathbf{w}^{t-1}\right\|_1^2\right)\\
={}&\alpha n + \sum_{t=1}^{T}\left(\underbrace{\left(\beta(n-1)^2 - \gamma\right)}_{\leq 0}\sum_{i\in N}\left\|\mathbf{w}^{t} - \mathbf{w}^{t-1}\right\|_1^2\right)\\
\leq{}&\alpha n
\end{aligned}
$$

Optimistic Follow the Regularized Leader

Optimistic Follow the Regularized Leader

Let $\mathcal{R}$ be a suitable regularizer and $\mathbf{M}_i^T$ be an adaptive prediction sequence:

$$\mathbf{w}_i^T = \operatorname*{argmax}_{\mathbf{w} \in \Delta(S_i)} \left\langle \mathbf{w}, \left(\sum_{t=1}^{T-1} \mathbf{u}_i^t \right) + \mathbf{M}_i^T \right\rangle - \frac{\mathcal{R}(\mathbf{w})}{\eta}.$$

Optimistic Follow the Regularized Leader

Optimistic Follow the Regularized Leader

Let $\mathcal{R}$ be a suitable regularizer and $\mathbf{M}_i^T$ be an adaptive prediction sequence:

$$\mathbf{w}_i^T = \operatorname*{argmax}_{\mathbf{w} \in \Delta(S_i)} \left\langle \mathbf{w}, \left(\sum_{t=1}^{T-1} \mathbf{u}_i^t \right) + \mathbf{M}_i^T \right\rangle - \frac{\mathcal{R}(\mathbf{w})}{\eta}.$$

Recency bias

Optimistic Follow the Regularized Leader has the `RVU` property with

- one-step recency bias $\mathbf{M}_i^t = \mathbf{u}_i^{t-1}$
- H-step recency bias $\mathbf{M}_i^t = \sum_{\tau=t-H}^{t-1} \mathbf{u}_i^\tau / H$
- geometrically discounted recency bias $\mathbf{M}_i^t = \frac{1}{\sum_{\tau=0}^{t-1} \delta^{-\tau}} \sum_{\tau=0}^{t-1} \delta^{-\tau} \mathbf{u}_i^\tau$

One-step recency bias

With $\mathbf{M}_i^t = \mathbf{u}_i^{t-1}$ and using stepsize η, *Optimistic Follow the Regularized Leader* satisfies the `RVU` property with constants $\alpha = R/\eta$, $\beta = \eta$ and $\gamma = 1/(4\eta)$ where $R = \max_i \left(\sup_{\mathbf{f} \in \Delta(S_i)} \mathcal{R}(\mathbf{f}) - \inf_{\mathbf{f} \in \Delta(S_i)} \mathcal{R}(\mathbf{f}) \right)$.

Other contributions

Meta-algorithm

They show a meta-algorithm that uses any tunable algorithm that satisfies the RVU property so that the RVU property is preserved but also that the worst case rate against adversarial environments is $O(1/\sqrt{T})$.

Other contributions

Meta-algorithm

They show a meta-algorithm that uses any tunable algorithm that satisfies the RVU property so that the RVU property is preserved but also that the worst case rate against adversarial environments is $O(1/\sqrt{T})$.

Fast convergence of each player's average regret

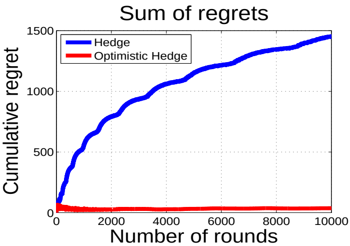
If either all players use

- *Optimistic Follow the Regularized Leader* with $\mathbf{M}_i^t = \mathbf{u}_i^{t-1}$ or
- all use the meta-algorithm with the same input algorithm that satisfies a certain stability condition

then each player's average regret converges at the rate of $O(T^{-3/4})$.

Experimental validation

Results I



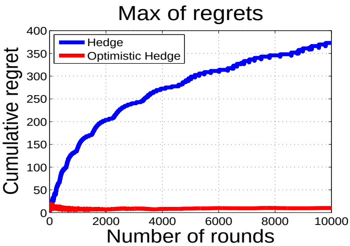


Figure: Maximum and sum of individual regrets over time under the Hedge (blue) and Optimistic Hedge (red) dynamics.

Results II

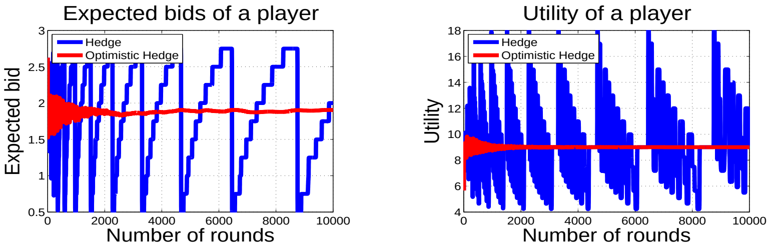


Figure: Expected bid and per-iteration utility of a player on one of the four items over time, under Hedge (blue) and Optimistic Hedge (red) dynamics.

Results

- When all players use no-regret algorithms with RVU property: **All players' average regret** converges at rate $O(1/T)$ instead of $O(1/\sqrt{T})$.

Results

- When all players use no-regret algorithms with $\mathtt{RVU}$ property: **All players' average regret** converges at rate $O(1/T)$ instead of $O(1/\sqrt{T})$.
- **Stability** and **recency bias** are key ingredients for fast converging algorithms, for which *Optimistic Follow the Regularized Leader* is an example.

Results

- When all players use no-regret algorithms with RVU property: **All players' average regret** converges at rate $O(1/T)$ instead of $O(1/\sqrt{T})$.
- **Stability** and **recency bias** are key ingredients for fast converging algorithms, for which *Optimistic Follow the Regularized Leader* is an example.
- Every tunable no-regret algorithm with the RVU property can be used by a **meta-algorithm** that than also satisfies the RVU property and guarantees a worst case rate of $O(1/\sqrt{T})$.

Results

- When all players use no-regret algorithms with RVU property: **All players' average regret** converges at rate $O(1/T)$ instead of $O(1/\sqrt{T})$.
- **Stability** and **recency bias** are key ingredients for fast converging algorithms, for which *Optimistic Follow the Regularized Leader* is an example.
- Every tunable no-regret algorithm with the RVU property can be used by a **meta-algorithm** that than also satisfies the RVU property and guarantees a worst case rate of $O(1/\sqrt{T})$.
- When all players use the same algorithm chosen from OFRL with $\mathbf{M}_i^t = \mathbf{u}_i^{t-1}$ or the meta-algorithm with the same input algorithm that satisfies the stability condition: Each player's **individual regret** converges at rate $O(T^{-3/4})$ instead of $O(1/\sqrt{T})$.

Discussion

- Is `RVU` necessary? (probably not)
- Is observing only the other's players moves instead of the expected utilities also enough to get faster rates?
- A precise quantification of the desired behaviour, which is necessary for stable trajectories for $\mathbf{w}_i$, is of great interest.