

Reduced Variance Payoff Estimation in Adversarial Bandit Problems

Levente Kocsis **Csaba Szepesvári**

Computer and Automation Research Institute of the
Hungarian Academy of Sciences
Kende u. 13-17, Budapest 1111, Hungary
E-mail: **szcsaba@sztaki.hu**

ECML-2005 Workshop on
Reinforcement Learning in Non-stationary Environments
Porto, 2005

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Prediction with Expert Advice: Motivation

Motivation: Opponent Modelling in Poker

- Knowing your opponent gives rise to substantial performance gains
- Opponent types
- Which of the available types is your opponent?

Prediction with Expert Advice: Motivation

Motivation: Opponent Modelling in Poker

- Knowing your opponent gives rise to substantial performance gains
- Opponent types
- Which of the available types is your opponent?

Prediction with Expert Advice: Motivation

Motivation: Opponent Modelling in Poker

- Knowing your opponent gives rise to substantial performance gains
- Opponent types
- Which of the available types is your opponent?

Prediction with Expert Advice (Bandit Problems)

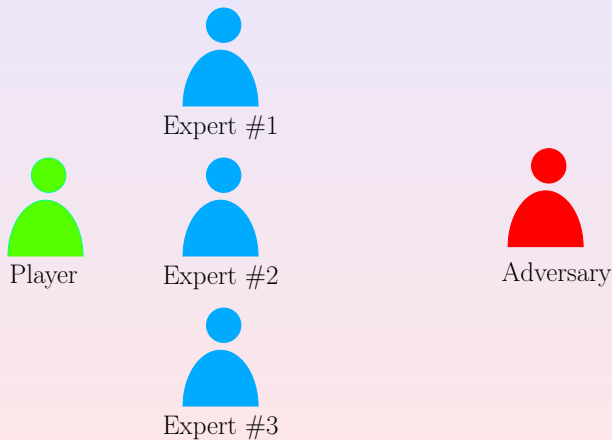


Player

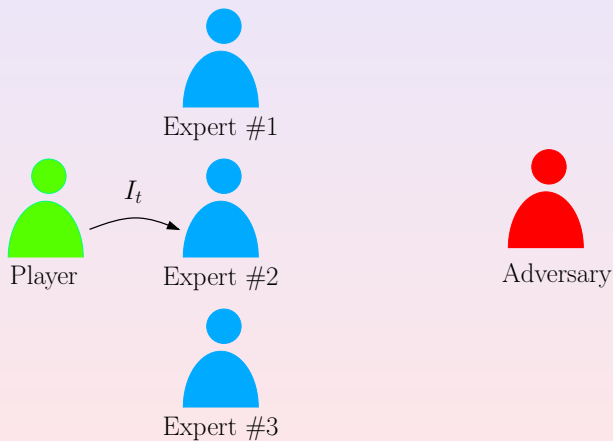


Adversary

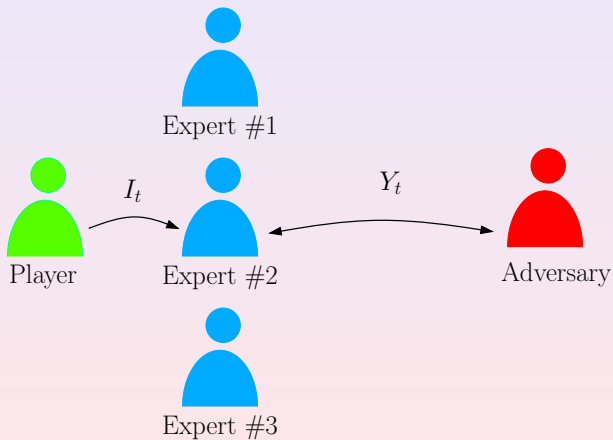
Prediction with Expert Advice (Bandit Problems)



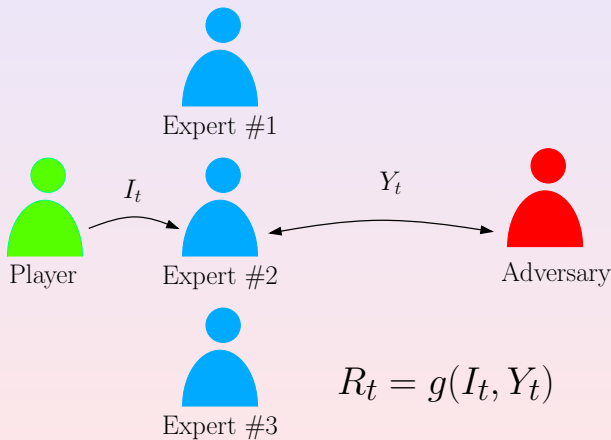
Prediction with Expert Advice (Bandit Problems)



Prediction with Expert Advice (Bandit Problems)



Prediction with Expert Advice (Bandit Problems)



Non-stationarity?

Any powerful adversary is allowed –
even **non-stationary** ones:
The adversary's “strategy” can change with time!

Setting the goal

- **Goal #1: Maximize total reward**

Very ambitious!! (too ambitious!)

- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest
- Consequences:

→ We need to have a single good expert for each of the adversaries of interest

⇒ We might need a large number of experts

⇒ Bounds should scale well with the # experts¹

- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(I_t, Y_t)$;
we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Setting the goal

- Goal #1: Maximize total reward
Very ambitious!! (too ambitious!)
- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest
- Consequences:
 - We need a way to compare each expert to each of the adversaries of interest
 - We might need a large number of experts
 - Bounds should scale well with the # experts¹
- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(I_t, Y_t)$;
 we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Setting the goal

- Goal #1: Maximize total reward
Very ambitious!! (too ambitious!)
- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest
- Consequences:
 - ⇒ there should be a single good expert for each of the adversaries of interest
 - ⇒ We might need a large number of experts
 - ⇒ Bounds should scale well with the # experts¹
- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(I_t, Y_t)$;
we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Setting the goal

- Goal #1: Maximize total reward
Very ambitious!! (too ambitious!)
- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest

- Consequences:

⇒ There should be a single good expert for each of the adversaries of interest

⇒ We might need a large number of experts

⇒ Bounds should scale well with the # experts¹

- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(I_t, Y_t)$;
we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Setting the goal

- Goal #1: Maximize total reward
Very ambitious!! (too ambitious!)
- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest

- **Consequences:**

- ⇒ There should be a single good expert for each of the adversaries of interest
- ⇒ We might need a large number of experts
- ⇒ Bounds should scale well with the # experts¹

- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(I_t, Y_t)$;
we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Setting the goal

- Goal #1: Maximize total reward
Very ambitious!! (too ambitious!)
- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest
- Consequences:

⇒ There should be a single good expert for each of the adversaries of interest

⇒ We might need a large number of experts

⇒ Bounds should scale well with the # experts¹

- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(I_t, Y_t)$;
we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Setting the goal

- Goal #1: Maximize total reward
Very ambitious!! (too ambitious!)
- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest
- Consequences:
 - ⇒ There should be a single good expert for each of the adversaries of interest
 - ⇒ We might need a large number of experts
 - ⇒ Bounds should scale well with the # experts¹
- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(l_t, Y_t)$;
we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Setting the goal

- Goal #1: Maximize total reward
Very ambitious!! (too ambitious!)
- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest
- Consequences:
 - ⇒ There should be a single good expert for each of the adversaries of interest
 - ⇒ We might need a large number of experts
 - ⇒ **Bounds should scale well with the # experts¹**
- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(l_t, Y_t)$;
we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Setting the goal

- Goal #1: Maximize total reward
Very ambitious!! (too ambitious!)
- Goal #2: Minimize loss over the total reward of the best single expert (regret)!
- Goal #2.1: .. uniformly over all adversaries of interest
- Consequences:
 - ⇒ There should be a single good expert for each of the adversaries of interest
 - ⇒ We might need a large number of experts
 - ⇒ Bounds should scale well with the # experts¹
- Formally: $G_{i,n} = \sum_{t=1}^n g(i, Y_t)$, $\hat{G}_n = \sum_{t=1}^n g(I_t, Y_t)$;
we want

$$R_n = \max_i G_{i,n} - \hat{G}_n \rightarrow \min.$$

¹Alternative is to consider tracking

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - **Some Previous Results**
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Results for Adversarial Bandit Problems

Theorem

For any time horizon n , the expected total regret of the Exp3 algorithm is at most

$$2\sqrt{2nN \ln N}$$

P. Auer et. al: “The nonstochastic multi-armed bandit problem”, *SIAM Journal on Computing*, 32:48–77, 2002.

Stationary environment²: $R_n = O(\ln n)$.

²T.L. Lai and H. Robbins: “Asymptotically efficient adaptive allocation rules” *Adv. in Appl. Math.*, 6:4-22, 1985.

The Exp3 Algorithm

Parameters: η – learning rate, γ – exploration rate.

Initialization: $w_0 = (1, \dots, 1)^T$;

For each round $t = 1, 2, \dots$

(1) select $I_t \in \{1, \dots, N\}$ randomly according to

$$p_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{\sum_{k=1}^N w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;

(3) Compute the feedbacks $g'_t(i, Y_t)$, $i = 1, \dots, N$:

$$g'_t(i, Y_t) = \mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t}.$$

(4) compute $w_{i,t} = w_{i,t-1} e^{\eta g'_t(i, Y_t)}$.

The Exp3 Algorithm

Parameters: η – learning rate, γ – exploration rate.

Initialization: $w_0 = (1, \dots, 1)^T$;

For each round $t = 1, 2, \dots$

(1) select $I_t \in \{1, \dots, N\}$ randomly according to

$$p_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{\sum_{k=1}^N w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;

(3) Compute the feedbacks $g'_t(i, Y_t)$, $i = 1, \dots, N$:

$$g'_t(i, Y_t) = \mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t}.$$

(4) compute $w_{i,t} = w_{i,t-1} e^{\eta g'_t(i, Y_t)}$.

The Exp3 Algorithm

Parameters: η – learning rate, γ – exploration rate.

Initialization: $w_0 = (1, \dots, 1)^T$;

For each round $t = 1, 2, \dots$

(1) select $I_t \in \{1, \dots, N\}$ randomly according to

$$p_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{\sum_{k=1}^N w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;

(3) Compute the feedbacks $g'_t(i, Y_t)$, $i = 1, \dots, N$:

$$g'_t(i, Y_t) = \mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t}.$$

(4) compute $w_{i,t} = w_{i,t-1} e^{\eta g'_t(i, Y_t)}$.

The Exp3 Algorithm

Parameters: η – learning rate, γ – exploration rate.

Initialization: $w_0 = (1, \dots, 1)^T$;

For each round $t = 1, 2, \dots$

(1) select $I_t \in \{1, \dots, N\}$ randomly according to

$$p_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{\sum_{k=1}^N w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;

(3) Compute the feedbacks $g'_t(i, Y_t)$, $i = 1, \dots, N$:

$$g'_t(i, Y_t) = \mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t}.$$

(4) compute $w_{i,t} = w_{i,t-1} e^{\eta g'_t(i, Y_t)}$.

The Exp3 Algorithm

Parameters: η – learning rate, γ – exploration rate.

Initialization: $w_0 = (1, \dots, 1)^T$;

For each round $t = 1, 2, \dots$

(1) select $I_t \in \{1, \dots, N\}$ randomly according to

$$p_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{\sum_{k=1}^N w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;

(3) Compute the feedbacks $g'_t(i, Y_t)$, $i = 1, \dots, N$:

$$g'_t(i, Y_t) = \mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t}.$$

(4) compute $w_{i,t} = w_{i,t-1} e^{\eta g'_t(i, Y_t)}$.

The Exp3 Algorithm

Parameters: η – learning rate, γ – exploration rate.

Initialization: $w_0 = (1, \dots, 1)^T$;

For each round $t = 1, 2, \dots$

(1) select $I_t \in \{1, \dots, N\}$ randomly according to

$$p_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{\sum_{k=1}^N w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;

(3) Compute the feedbacks $g'_t(i, Y_t)$, $i = 1, \dots, N$:

$$g'_t(i, Y_t) = \mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t}.$$

(4) compute $w_{i,t} = w_{i,t-1} e^{\eta g'_t(i, Y_t)}$.

The Exp3 Algorithm

Parameters: η – learning rate, γ – exploration rate.

Initialization: $w_0 = (1, \dots, 1)^T$;

For each round $t = 1, 2, \dots$

(1) select $I_t \in \{1, \dots, N\}$ randomly according to

$$p_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{\sum_{k=1}^N w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;

(3) Compute the feedbacks $g'_t(i, Y_t)$, $i = 1, \dots, N$:

$$g'_t(i, Y_t) = \mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t}.$$

(4) compute $w_{i,t} = w_{i,t-1} e^{\eta g'_t(i, Y_t)}$.

Interpretation

- Fully observable case; $g(i, Y_t)$ is known for each round.
- Exponentially weighted predictors:³

$$w_{i,n} = \exp(\eta \sum_{t=1}^n g(i, Y_t)), \quad \gamma = 0$$

- Here $g'_t(i, Y_t)$ is an unbiased estimate of the reward $g(i, Y_t)$,

$$\sum_{t=1}^n g_t(i, Y_t) \sim \sum_{t=1}^n g'_t(i, Y_t).$$

³Weighted majority – Littlestone & Warmuth (1994), Aggregating strategies – Vovk (1990), Freund & Schapire (1997, 1999).

Interpretation

- Fully observable case; $g(i, Y_t)$ is known for each round.
- Exponentially weighted predictors:³

$$w_{i,n} = \exp\left(\eta \sum_{t=1}^n g(i, Y_t)\right), \quad \gamma = 0$$

- Here $g'_t(i, Y_t)$ is an unbiased estimate of the reward $g(i, Y_t)$,

$$\sum_{t=1}^n g_t(i, Y_t) \sim \sum_{t=1}^n g'_t(i, Y_t).$$

³Weighted majority – Littlestone & Warmuth (1994), Aggregating strategies – Vovk (1990), Freund & Schapire (1997, 1999).

Interpretation

- Fully observable case; $g(i, Y_t)$ is known for each round.
- Exponentially weighted predictors:³

$$w_{i,n} = \exp(\eta \sum_{t=1}^n g(i, Y_t)), \quad \gamma = 0$$

- Here $g'_t(i, Y_t)$ is an unbiased estimate of the reward $g(i, Y_t)$,

$$\sum_{t=1}^n g_t(i, Y_t) \sim \sum_{t=1}^n g'_t(i, Y_t).$$

³Weighted majority – Littlestone & Warmuth (1994), Aggregating strategies – Vovk (1990), Freund & Schapire (1997, 1999).

Exp3: Unbiasedness of Payoff Estimates

$$\begin{aligned}\mathbb{E}[g'_t(i, Y_t) | \mathcal{F}_t] &= \sum_{j=1}^N \mathbb{E}[\mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t} | \mathcal{F}_t] \mathbb{P}(I_t = j | \mathcal{F}_t) \\ &= g(i, Y_t) / p_{i,t} \mathbb{P}(I_t = i | \mathcal{F}_t) \\ &= g(i, Y_t).\end{aligned}$$

Exp3: Unbiasedness of Payoff Estimates

$$\begin{aligned}\mathbb{E}[g'_t(i, Y_t) | \mathcal{F}_t] &= \sum_{j=1}^N \mathbb{E}[\mathbb{I}(I_t = i) g(I_t, Y_t) / p_{i,t} | \mathcal{F}_t] \mathbb{P}(I_t = j | \mathcal{F}_t) \\ &= g(i, Y_t) / p_{i,t} \mathbb{P}(I_t = i | \mathcal{F}_t) \\ &= g(i, Y_t).\end{aligned}$$

Exp3: Unbiasedness of Payoff Estimates

$$\begin{aligned}\mathbb{E}[g'_t(i, Y_t) \mid \mathcal{F}_t] &= \sum_{j=1}^N \mathbb{E}[\mathbb{I}(l_t = i) g(l_t, Y_t) / p_{i,t} \mid \mathcal{F}_t] \mathbb{P}(l_t = j \mid \mathcal{F}_t) \\ &= g(i, Y_t) / p_{i,t} \mathbb{P}(l_t = i \mid \mathcal{F}_t) \\ &= g(i, Y_t).\end{aligned}$$

Exp3: Unbiasedness of Payoff Estimates

$$\begin{aligned}\mathbb{E}[g'_t(i, Y_t) \mid \mathcal{F}_t] &= \sum_{j=1}^N \mathbb{E}[\mathbb{I}(l_t = i) g(l_t, Y_t) / p_{i,t} \mid \mathcal{F}_t] \mathbb{P}(l_t = j \mid \mathcal{F}_t) \\ &= g(i, Y_t) / p_{i,t} \mathbb{P}(l_t = i \mid \mathcal{F}_t) \\ &= g(i, Y_t).\end{aligned}$$

Example: Dynamic Pricing, single product

- **Player:** A vendor – wants to sell a product
- **Adversary:** A new customer in each round
- **Problem:** Select the “right” price (high enough to make profit, low enough to cut a deal!)
- **Expert i** suggests price $p_{1,i}$
- **Highest price** the customer is willing to accept: p_2 (unknown! never revealed!)
- **Payoff:** $g_i = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2)$; vendor learns only g_i !
- $y = (p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2).$$

Example: Dynamic Pricing, single product

- **Player:** A vendor – wants to sell a product
- **Adversary:** A new customer in each round
- **Problem:** Select the “right” price (high enough to make profit, low enough to cut a deal!)
- **Expert i** suggests price $p_{1,i}$
- **Highest price** the customer is willing to accept: p_2 (unknown! never revealed!)
- **Payoff:** $g_i = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2)$; vendor learns only g_{i_t} !
- $y = (p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2).$$

Example: Dynamic Pricing, single product

- **Player:** A vendor – wants to sell a product
- **Adversary:** A new customer in each round
- **Problem:** Select the “right” price (high enough to make profit, low enough to cut a deal!)
- **Expert i** suggests price $p_{1,i}$
- **Highest price** the customer is willing to accept: p_2 (unknown! never revealed!)
- **Payoff:** $g_i = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2)$; vendor learns only g_{i_t} !
- $y = (p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2).$$

Example: Dynamic Pricing, single product

- **Player:** A vendor – wants to sell a product
- **Adversary:** A new customer in each round
- **Problem:** Select the “right” price (high enough to make profit, low enough to cut a deal!)
- **Expert i** suggests price $p_{1,i}$
- **Highest price** the customer is willing to accept: p_2 (unknown! never revealed!)
- **Payoff:** $g_i = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2)$; vendor learns only g_{i_t} !
- $y = (p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2).$$

Example: Dynamic Pricing, single product

- **Player:** A vendor – wants to sell a product
- **Adversary:** A new customer in each round
- **Problem:** Select the “right” price (high enough to make profit, low enough to cut a deal!)
- **Expert i** suggests price $p_{1,i}$
- **Highest price** the customer is willing to accept: p_2 (unknown! never revealed!)
- **Payoff:** $g_i = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2)$; vendor learns only g_{i_t} !
- $y = (p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2).$$

Example: Dynamic Pricing, single product

- **Player:** A vendor – wants to sell a product
- **Adversary:** A new customer in each round
- **Problem:** Select the “right” price (high enough to make profit, low enough to cut a deal!)
- **Expert i** suggests price $p_{1,i}$
- **Highest price** the customer is willing to accept: p_2 (unknown! never revealed!)
- **Payoff:** $g_i = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2)$; vendor learns only g_t !
- $y = (p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2).$$

Example: Dynamic Pricing, single product

- **Player:** A vendor – wants to sell a product
- **Adversary:** A new customer in each round
- **Problem:** Select the “right” price (high enough to make profit, low enough to cut a deal!)
- **Expert i** suggests price $p_{1,i}$
- **Highest price** the customer is willing to accept: p_2 (unknown! never revealed!)
- **Payoff:** $g_i = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2)$; vendor learns only g_t !
- $y = (p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2).$$

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - **Issues**
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Some issues

- Practical performance is often very poor

We would like to use bandit algorithms in Poker for opponent modelling!

- Bound scales with N like $O(N \ln N)$

- Fully observable case: $O(\ln N)$

- Possible remedy(?): Allow best expert change with time⁴

- More information/structure is often available – why not exploiting it?

⁴M.K. Warmuth, M. Heister: "Tracking the best expert", *Machine Learning*, 32:151–178, 1998.

Some issues

- Practical performance is often very poor
We would like to use bandit algorithms in Poker for opponent modelling!
- Bound scales with N like $O(N \ln N)$
 - Fully observable case: $O(\ln N)$
 - Possible remedy(?): Allow best expert change with time⁴
- More information/structure is often available – why not exploiting it?

⁴M.K. Warmuth, M. Hebbster: "Tracking the best expert", *Machine Learning*, 32:151–178, 1998.

Some issues

- Practical performance is often very poor
We would like to use bandit algorithms in Poker for opponent modelling!
- Bound scales with N like $O(N \ln N)$
 - Fully observable case: $O(\ln N)$
 - Possible remedy(?): Allow best expert change with time⁴
- More information/structure is often available – why not exploiting it?

⁴M.K. Warmuth, M. Hebster: “Tracking the best expert”, *Machine Learning*, 32:151–178, 1998.

Some issues

- Practical performance is often very poor
We would like to use bandit algorithms in Poker for opponent modelling!
- Bound scales with N like $O(N \ln N)$
 - Fully observable case: $O(\ln N)$
 - Possible remedy(?): Allow best expert change with time⁴
- More information/structure is often available – why not exploiting it?

⁴M.K. Warmuth, M. Hebbster: “Tracking the best expert”, *Machine Learning*, 32:151–178, 1998.

Some issues

- Practical performance is often very poor
We would like to use bandit algorithms in Poker for opponent modelling!
- Bound scales with N like $O(N \ln N)$
 - Fully observable case: $O(\ln N)$
 - Possible remedy(?): Allow best expert change with time⁴
- More information/structure is often available – why not exploiting it?

⁴M.K. Warmuth, M. Hebster: “Tracking the best expert”, *Machine Learning*, 32:151–178, 1998.

Some issues

- Practical performance is often very poor
We would like to use bandit algorithms in Poker for opponent modelling!
- Bound scales with N like $O(N \ln N)$
 - Fully observable case: $O(\ln N)$
 - Possible remedy(?): Allow best expert change with time⁴
- More information/structure is often available – why not exploiting it?

⁴M.K. Warmuth, M. Hebster: “Tracking the best expert”, *Machine Learning*, 32:151–178, 1998.

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Side information

Bandit Problems with Side Information

- 1 Player receives information C_t about the environment state
- 2 Player selects expert I_t
- 3 Expert I_t plays against the adversary, knowing C_t
- 4 Player receives payoff $g(I_t, Y_t)$

Side information

Bandit Problems with Side Information

- 1 Player receives information C_t about the environment state
- 2 Player selects expert I_t
- 3 Expert I_t plays against the adversary, knowing C_t
- 4 Player receives payoff $g(I_t, Y_t)$

Side information

Bandit Problems with Side Information

- 1 Player receives information C_t about the environment state
- 2 Player selects expert I_t
- 3 Expert I_t plays against the adversary, knowing C_t
- 4 Player receives payoff $g(I_t, Y_t)$

Side information

Bandit Problems with Side Information

- 1 Player receives information C_t about the environment state
- 2 Player selects expert I_t
- 3 Expert I_t plays against the adversary, knowing C_t
- 4 Player receives payoff $g(I_t, Y_t)$

The Key Observation

Hypothesis

- In Exp3 any unbiased estimate of the immediate payoffs will do..
- Estimates with smaller variance should lead to more efficient algorithms.

The Key Observation

Hypothesis

- In Exp3 any unbiased estimate of the immediate payoffs will do..
- Estimates with smaller variance should lead to more efficient algorithms.

The Key Observation

Hypothesis

- In Exp3 any unbiased estimate of the immediate payoffs will do..
- Estimates with smaller variance should lead to more efficient algorithms.

Example: Dynamic Pricing, multiple products

- Side information: the cost of the product v
- Payoff: $p_1 \mathbb{I}(p_1 \leq p_2) + (1 - \alpha)v \mathbb{I}(p_1 < p_2)$.
- $y = (v, p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2) + (1 - \alpha)v \mathbb{I}(p_{1,i} > p_2).$$

- Hypothesis: One should be able to reduce payoff variance given the knowledge of v

Example: Dynamic Pricing, multiple products

- **Side information:** the cost of the product v
- **Payoff:** $p_1 \mathbb{I}(p_1 \leq p_2) + (1 - \alpha)v \mathbb{I}(p_1 < p_2)$.
- $y = (v, p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2) + (1 - \alpha)v \mathbb{I}(p_{1,i} > p_2).$$

- Hypothesis: One should be able to reduce payoff variance given the knowledge of v

Example: Dynamic Pricing, multiple products

- **Side information:** the cost of the product v
- **Payoff:** $p_1 \mathbb{I}(p_1 \leq p_2) + (1 - \alpha)v \mathbb{I}(p_1 < p_2)$.
- $y = (v, p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2) + (1 - \alpha)v \mathbb{I}(p_{1,i} > p_2).$$

- Hypothesis: One should be able to reduce payoff variance given the knowledge of v

Example: Dynamic Pricing, multiple products

- **Side information:** the cost of the product v
- **Payoff:** $p_1 \mathbb{I}(p_1 \leq p_2) + (1 - \alpha)v \mathbb{I}(p_1 < p_2)$.
- $y = (v, p_2, p_{1,1}, \dots, p_{1,N})$, so

$$g(i, y) = p_{1,i} \mathbb{I}(p_{1,i} \leq p_2) + (1 - \alpha)v \mathbb{I}(p_{1,i} > p_2).$$

- **Hypothesis:** One should be able to reduce payoff variance given the knowledge of v

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - **Theoretical Results**
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

The Exp3G Algorithm

Parameters: real numbers $0 < \eta, \gamma < 1$.

Initialization: $w_0 = (1, \dots, 1)^T$;

For each round $t = 1, 2, \dots$

(1) **observe** C_t , select $I_t \in \{1, \dots, N\}$ according to

$$p_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{\sum_{k=1}^N w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;

(3) **based on** g_t , C_t , **compute the feedbacks** $g'_t(i, Y_t)$,
 $i = 1, \dots, N$;

(4) compute $w_{i,t} = w_{i,t-1} e^{\eta g'_t(i, Y_t)}$.

Exp3G: Expected Regret

Assumptions:

$$(A1) \quad \mathbb{E}[g(i, Y_t) | C_t, I^{t-1}, Y^{t-1}] \leq 1;$$

$$(A2) \quad \mathbb{E}[g'_t(i, Y_t) | C_t, I^{t-1}, Y^{t-1}] = \mathbb{E}[g(i, Y_t) | C_t, I^{t-1}, Y^{t-1}];$$

$$(A3) \quad \text{Var}[g'_t(i, Y_t) | C_t, I^{t-1}, Y^{t-1}] \leq \sigma^2;$$

$$(A4) \quad |g'_t(i, Y_t)| \leq B.$$

Theorem

Consider algorithm Exp3G. Assume A1–A4. Then for $\gamma = 0$ and a suitable $\eta = \eta_n$, n sufficiently large,

$$\max_i \mathbb{E}\left[\sum_{t=1}^n g(i, Y_t)\right] - \mathbb{E}\left[\sum_{t=1}^n g(I_t, Y_t)\right] \leq \sqrt{(1 + \sigma^2)n \ln N}.$$

Exp3G: PAC-bounds on Regret

Theorem

Consider algorithm Exp3G. Assume A1–A4 and further, assume that $|g(i, Y_t)| \leq 1$. Then, for any $\delta > 0$, for a suitable $\eta = \eta_n$ with n sufficiently large, the following bound on the regret of Exp3G holds with probability at least $1 - \delta$:

$$\max_i G_{in} - \hat{G}_n \leq n^{1/2} \left(((1 + \sigma^2) \ln N)^{1/2} + (2 + \sqrt{2}\sigma) \ln \left(\frac{N+1}{\delta} \right)^{1/2} \right) + \frac{2(B+1)}{3} \ln \left(\frac{N+1}{\delta} \right).$$

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - **Likelihood-ratio Based Payoff Estimation**
 - Control-variates
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Likelihood-ratio-Based Payoff Estimation

Assumptions

- Explicit model of the game played (actions).
- Experts have probabilistic action selection strategies.
- Probability of any action (action-sequence) can be queried for *any* expert.

Payoff-estimate (single-stage games):

$$g'_t(i, Y_t) = \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t).$$

- This estimate is unbiased.
- Extension to multi-stage games is straightforward (use the chain-rule to show unbiasedness).

Likelihood-ratio-Based Payoff Estimation

Assumptions

- Explicit model of the game played (actions).
- Experts have probabilistic action selection strategies.
- Probability of any action (action-sequence) can be queried for *any* expert.

Payoff-estimate (single-stage games):

$$g'_t(i, Y_t) = \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t).$$

- This estimate is unbiased.
- Extension to multi-stage games is straightforward (use the chain-rule to show unbiasedness).

Likelihood-ratio-Based Payoff Estimation

Assumptions

- Explicit model of the game played (actions).
- Experts have probabilistic action selection strategies.
- Probability of any action (action-sequence) can be queried for *any* expert.

Payoff-estimate (single-stage games):

$$g'_t(i, Y_t) = \frac{\pi_i(\mathbf{A}_t | \mathbf{C}_t)}{\pi_{I_t}(\mathbf{A}_t | \mathbf{C}_t)} g(I_t, Y_t).$$

- This estimate is unbiased.
- Extension to multi-stage games is straightforward (use the chain-rule to show unbiasedness).

Likelihood-ratio-Based Payoff Estimation

Assumptions

- Explicit model of the game played (actions).
- Experts have probabilistic action selection strategies.
- Probability of any action (action-sequence) can be queried for *any* expert.

Payoff-estimate (single-stage games):

$$g'_t(i, Y_t) = \frac{\pi_i(\mathbf{A}_t | \mathbf{C}_t)}{\pi_{I_t}(\mathbf{A}_t | \mathbf{C}_t)} g(I_t, Y_t).$$

- This estimate is unbiased.
- Extension to multi-stage games is straightforward (use the chain-rule to show unbiasedness).

LExp: LR-based Payoff Estimation # 1

Simple LR-based estimate:

$$g'_t(i, Y_t) = \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t).$$

Insight

Large likelihood ratios are likely to yield large variance.

Idea: When $\pi_i(A_t|C_t)/\pi_{I_t}(A_t|C_t)$ is big, make it equal to 0 and compensate for the bias introduced.

LExp: LR-based Payoff Estimation # 2

- Simple LR-based estimate:

$$g'_t(i, Y_t) = \frac{\pi_i(A_t|C_t)}{\pi_{l_t}(A_t|C_t)} g(l_t, Y_t).$$

- Let $\phi_t(l_t, A_t, i)$ be such that $\phi_t(l_t, A_t, i) = 1$ denotes an event when the LRs are big:

$$\phi_t(k, a, i) = \mathbb{I} \left(\frac{\pi_i(a|C_t)}{\pi_k(a|C_t)} > \frac{p_{k,t}}{p_{i,t}} \right).$$

- Modified LR-based estimate:

$$g'_t(i, Y_t) = \frac{(1 - \phi_t(l_t, A_t, i))}{\sum_{j=1}^N p_{j,t}(1 - \phi_t(j, A_t, i))} \frac{\pi_i(A_t|C_t)}{\pi_{l_t}(A_t|C_t)} g(l_t, Y_t).$$

LExp: LR-based Payoff Estimation # 2

- Simple LR-based estimate:

$$g'_t(i, Y_t) = \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t).$$

- Let $\phi_t(I_t, A_t, i)$ be such that $\phi_t(I_t, A_t, i) = 1$ denotes an event when the LR's are big:

$$\phi_t(k, a, i) = \mathbb{I} \left(\frac{\pi_i(a|C_t)}{\pi_k(a|C_t)} > \frac{p_{k,t}}{p_{i,t}} \right).$$

- Modified LR-based estimate:

$$g'_t(i, Y_t) = \frac{(1 - \phi_t(I_t, A_t, i))}{\sum_{j=1}^N p_{j,t}(1 - \phi_t(j, A_t, i))} \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t).$$

LExp: LR-based Payoff Estimation # 2

- Simple LR-based estimate:

$$g'_t(i, Y_t) = \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t).$$

- Let $\phi_t(I_t, A_t, i)$ be such that $\phi_t(I_t, A_t, i) = 1$ denotes an event when the LRs are big:

$$\phi_t(k, a, i) = \mathbb{I} \left(\frac{\pi_i(a|C_t)}{\pi_k(a|C_t)} > \frac{p_{k,t}}{p_{i,t}} \right).$$

- Modified LR-based estimate:

$$g'_t(i, Y_t) = \frac{(1 - \phi_t(I_t, A_t, i))}{\sum_{j=1}^N p_{j,t}(1 - \phi_t(j, A_t, i))} \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t).$$

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - **Control-variates**
- 3 Experimental Results
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

CExp3: Control-variables # 1

- **Motivation:** In dynamic pricing, the product C_t controls to a large extent the distribution of the actual payoffs $g(i, Y_t)$ – hence also the variance.
- **Idea:** consider the payoffs compensated for C_t instead of the original payoffs:

$$\begin{aligned}g_c(i, Y_t) &= g(i, Y_t) - r(C_t), \\g'_{c,t}(i, Y_t) &= g'_t(i, Y_t) - r(C_t).\end{aligned}$$

Here $r(C_t)$ is the mean payoff when seeing C_t .

CExp3: Control-variables # 1

- **Motivation:** In dynamic pricing, the product C_t controls to a large extent the distribution of the actual payoffs $g(i, Y_t)$ – hence also the variance.
- **Idea:** consider the payoffs compensated for C_t instead of the original payoffs:

$$\begin{aligned}g_c(i, Y_t) &= g(i, Y_t) - r(C_t), \\g'_{c,t}(i, Y_t) &= g'_t(i, Y_t) - r(C_t).\end{aligned}$$

Here $r(C_t)$ is the mean payoff when seeing C_t .

CExp3: Control-variables # 2

- **Effect:**

$\text{Var}[g'_t(i, Y_t) | Y^{t-1}, I^{t-1}]$ is reduced. Previous analysis can be repeated to show that this is beneficial – regret bounds for compressed range payoffs carry over to the regret defined with the unmodified payoffs.

- **Intuitive explanation:**

Compresses the range of payoffs.

CExp3: Control-variables # 2

- **Effect:**

$\text{Var}[g'_t(i, Y_t) | Y^{t-1}, I^{t-1}]$ is reduced. Previous analysis can be repeated to show that this is beneficial – regret bounds for compressed range payoffs carry over to the regret defined with the unmodified payoffs.

- **Intuitive explanation:**

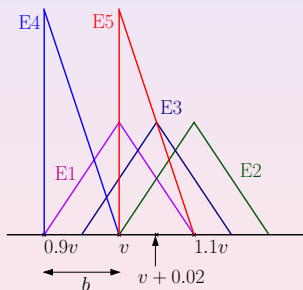
Compresses the range of payoffs.

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results**
 - Dynamic Pricing**
 - Dynamic Pricing – Tracking Experiments
 - Experiments with Poker
- 4 Conclusions

Dynamic pricing

Experts:

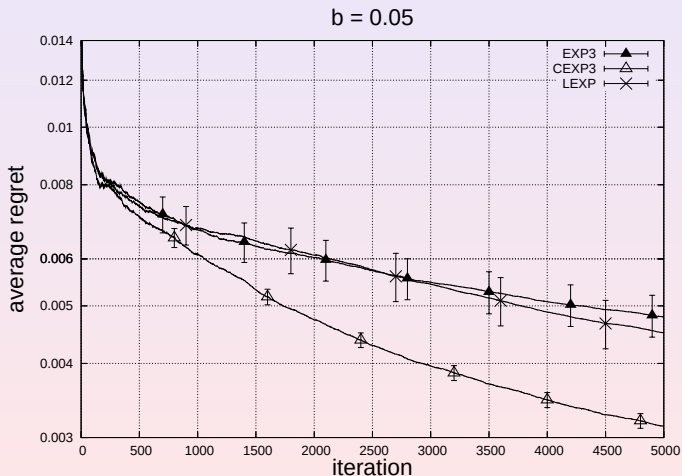


Customers:

$$p_2 = 1.1v + \frac{\beta - 50}{500},$$

where $\beta \sim B(100, 0.5)$.

Results: Almost deterministic experts



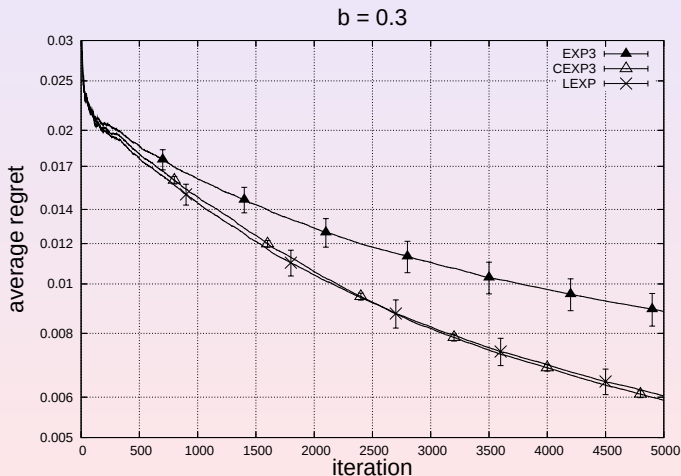
Dynamic pricing: Statistics # 1

$$b = 0.05$$

$\mathbb{E}[g'_t(i, Y_t)]$	i=1	i=2	i=3	i=4	i=5
Exp3	0.388	0.399	0.401	0.371	0.396
CExp3	0.390	0.399	0.398	0.371	0.398
LExp	0.390	0.402	0.400	0.368	0.399
$\mathbb{E}[g(i, Y_t)]$	0.390	0.400	0.399	0.371	0.399

$\sqrt{\text{Var}[g'_t(i, Y_t)]}$	i=1	i=2	i=3	i=4	i=5
Exp3	1.782	1.435	1.427	2.097	1.573
CExp3	0.467	0.476	0.473	0.332	0.472
LExp	0.739	0.788	0.500	1.671	0.688
$\sqrt{\text{Var}[g(i, Y_t)]}$	0.143	0.148	0.145	0.129	0.144

Results: Heavily randomized experts



Dynamic pricing: Statistics # 2

 $b = 0.3$

$\mathbb{E}[g'_t(i, Y_t)]$	i=1	i=2	i=3	i=4	i=5
Exp3	0.338	0.351	0.347	0.385	0.383
CExp3	0.343	0.354	0.348	0.381	0.382
LExp	0.343	0.356	0.351	0.383	0.384
$\mathbb{E}[g(i, Y_t)]$	0.343	0.356	0.350	0.383	0.384

$\sqrt{\text{Var}[g'_t(i; Y_t)]}$	i=1	i=2	i=3	i=4	i=5
Exp3	2.107	1.929	2.014	1.046	1.169
CExp3	0.735	0.724	0.726	0.744	0.745
LExp	0.856	0.573	0.651	0.475	0.412
$\sqrt{\text{Var}[g(i, Y_t)]}$	0.153	0.151	0.150	0.141	0.143

Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 **Experimental Results**
 - Dynamic Pricing
 - **Dynamic Pricing – Tracking Experiments**
 - Experiments with Poker
- 4 Conclusions

Experiments with Tracking Algorithms

- **Goal:** minimize regret over the sequence of best experts
where the frequency of expert changes is upper bounded
- **Problem:** Exp3 and variants can run into problems (weights converging too fast – too slow response at changepoints)
- Warmuth & Herbster: Tracking the Best Expert (ML, 1998);
Fixed-Share Algorithm:

$$w_{i,t} = \alpha \frac{\sum_{j=1}^N w_{j,t-1} e^{\eta g'_t(j, Y_t)}}{N} + (1 - \alpha) w_{i,t-1} e^{\eta g'_t(i, Y_t)}.$$

Experiments with Tracking Algorithms

- **Goal:** minimize regret over the sequence of best experts where the frequency of expert changes is upper bounded
- **Problem:** Exp3 and variants can run into problems (weights converging too fast – too slow response at changepoints)
- Warmuth & Herbster: Tracking the Best Expert (ML, 1998); Fixed-Share Algorithm:

$$w_{i,t} = \alpha \frac{\sum_{j=1}^N w_{j,t-1} e^{\eta g'_t(j, Y_t)}}{N} + (1 - \alpha) w_{i,t-1} e^{\eta g'_t(i, Y_t)}.$$

Experiments with Tracking Algorithms

- **Goal:** minimize regret over the sequence of best experts where the frequency of expert changes is upper bounded
- **Problem:** Exp3 and variants can run into problems (weights converging too fast – too slow response at changepoints)
- Warmuth & Herbster: Tracking the Best Expert (ML, 1998); Fixed-Share Algorithm:

$$w_{i,t} = \alpha \frac{\sum_{j=1}^N w_{j,t-1} e^{\eta g'_t(j, Y_t)}}{N} + (1 - \alpha) w_{i,t-1} e^{\eta g'_t(i, Y_t)}.$$

Experiments with Tracking Algorithms

- **Goal:** minimize regret over the sequence of best experts where the frequency of expert changes is upper bounded
- **Problem:** Exp3 and variants can run into problems (weights converging too fast – too slow response at changepoints)
- Warmuth & Herbster: Tracking the Best Expert (ML, 1998); Fixed-Share Algorithm:

$$w_{i,t} = \alpha \frac{\sum_{j=1}^N w_{j,t-1} e^{\eta g'_t(j, Y_t)}}{N} + (1 - \alpha) w_{i,t-1} e^{\eta g'_t(i, Y_t)}.$$

Experiments with Tracking Algorithms

- **Goal:** minimize regret over the sequence of best experts where the frequency of expert changes is upper bounded
- **Problem:** Exp3 and variants can run into problems (weights converging too fast – too slow response at changepoints)
- Warmuth & Herbster: Tracking the Best Expert (ML, 1998); Fixed-Share Algorithm:

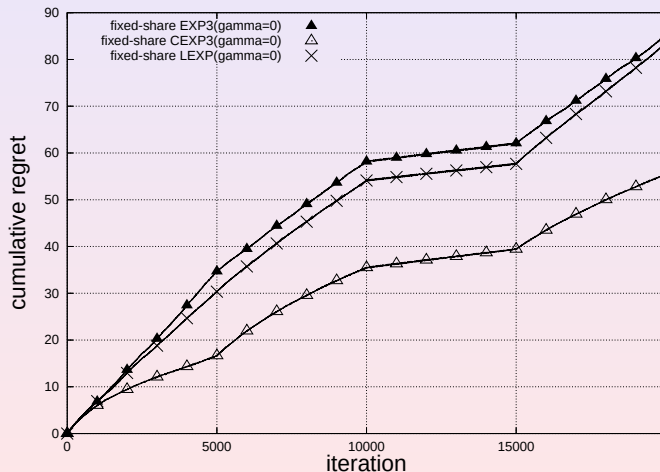
$$w_{i,t} = \alpha \frac{\sum_{j=1}^N w_{j,t-1} e^{\eta g'_t(j, Y_t)}}{N} + (1 - \alpha) w_{i,t-1} e^{\eta g'_t(i, Y_t)}.$$

Parameters of Experiments

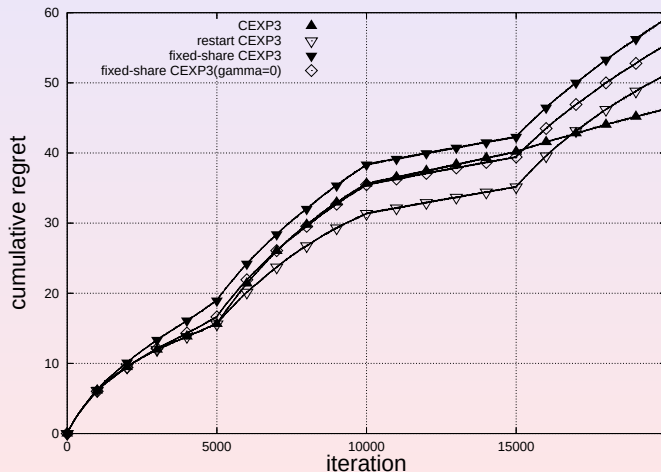
round	m	k	$\mathbb{E}[g(1, Y_t)]$	$\mathbb{E}[g(2, Y_t)]$	$\mathbb{E}[g(3, Y_t)]$	$\mathbb{E}[g(4, Y_t)]$	$\mathbb{E}[g(5, Y_t)]$
0	100	1.1	0.3906	0.4000	0.3992	0.3714	0.3989
5000	10	0.1	0.3767	0.3641	0.3716	0.371	0.3760
10000	20	0.9	0.3607	0.3604	0.3605	0.3615	0.3606
15000	100	1.0	0.3785	0.3766	0.3806	0.3687	0.3822

- $b = 0.05$
- $p_2 = kv + \frac{\beta - m/2}{m}, \beta \sim B(m, 0.5)$

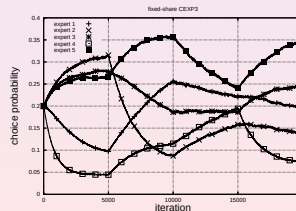
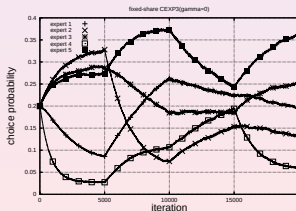
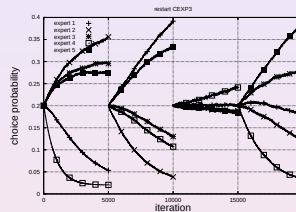
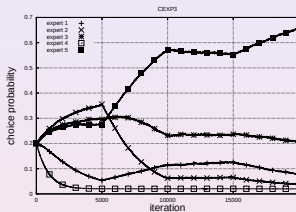
No exploration, fixed-share update-rule



Results for CExp3 variants



Expert-selection Frequencies (CExp3)



Outline

- 1 Universal Prediction with Expert Advice
 - Prediction with Expert Advice
 - Some Previous Results
 - Issues
- 2 Improved Payoff-Estimation
 - Generalized Exp3
 - Theoretical Results
 - Likelihood-ratio Based Payoff Estimation
 - Control-variates
- 3 Experimental Results**
 - Dynamic Pricing
 - Dynamic Pricing – Tracking Experiments
 - **Experiments with Poker**
- 4 Conclusions

Opponent Modelling in Poker

- **Omaha Hi-Lo**
- McRaise: Monte-Carlo Sampling to Approximate Action-Values
- Opponent Model: Weights card configurations in face of opponents' previous actions
- Five models: random, greedy, smooth, mcr, spsa, humanoid
- Algorithms: Exp3, aCExp3, cCExp3, LExp

Opponent Modelling in Poker

- Omaha Hi-Lo
- **McRaise: Monte-Carlo Sampling to Approximate Action-Values**
- Opponent Model: Weights card configurations in face of opponents' previous actions
- Five models: random, greedy, smooth, mcr, spsa, humanoid
- Algorithms: Exp3, aCExp3, cCExp3, LExp

Opponent Modelling in Poker

- Omaha Hi-Lo
- McRaise: Monte-Carlo Sampling to Approximate Action-Values
- Opponent Model: Weights card configurations in face of opponents' previous actions
- Five models: random, greedy, smooth, mcr, spsa, humanoid
- Algorithms: Exp3, aCExp3, cCExp3, LExp

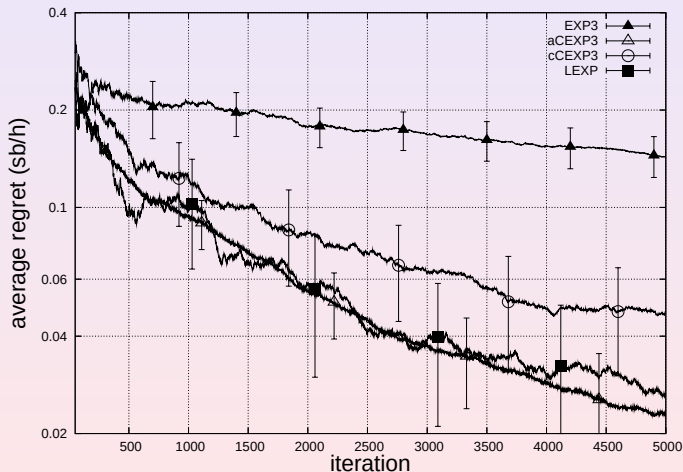
Opponent Modelling in Poker

- Omaha Hi-Lo
- McRaise: Monte-Carlo Sampling to Approximate Action-Values
- Opponent Model: Weights card configurations in face of opponents' previous actions
- Five models: random, greedy, smooth, mcr, spsa, humanoid
- Algorithms: Exp3, aCExp3, cCExp3, LExp

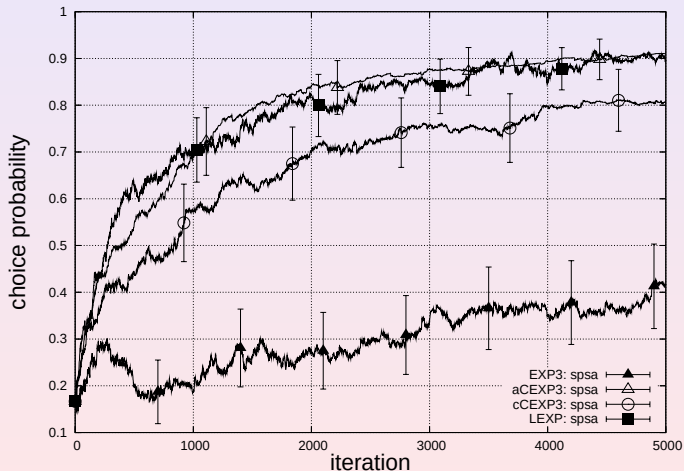
Opponent Modelling in Poker

- Omaha Hi-Lo
- McRaise: Monte-Carlo Sampling to Approximate Action-Values
- Opponent Model: Weights card configurations in face of opponents' previous actions
- Five models: random, greedy, smooth, mcr, spsa, humanoid
- Algorithms: Exp3, aCExp3, cCExp3, LExp

Opponent Modelling in Poker: Regret



Poker: Choosing the best expert



Conclusions

- Regret-minimization framework $\sim$ non-stationary environments
- Bandit algorithms:
 - Role of payoff-estimate variance
 - Algorithms for reducing the variance
- Results: Substantial improvements
.. though in poker far from being useful (so-far)

Conclusions

- Regret-minimization framework $\sim$ non-stationary environments
- **Bandit algorithms:**
 - Role of payoff-estimate variance
 - Algorithms for reducing the variance
- Results: Substantial improvements
.. though in poker far from being useful (so-far)

Conclusions

- Regret-minimization framework $\sim$ non-stationary environments
- Bandit algorithms:
 - **Role of payoff-estimate variance**
 - Algorithms for reducing the variance
- Results: Substantial improvements
.. though in poker far from being useful (so-far)

Conclusions

- Regret-minimization framework $\sim$ non-stationary environments
- Bandit algorithms:
 - Role of payoff-estimate variance
 - Algorithms for reducing the variance
- Results: Substantial improvements
.. though in poker far from being useful (so-far)

Conclusions

- Regret-minimization framework $\sim$ non-stationary environments
- Bandit algorithms:
 - Role of payoff-estimate variance
 - Algorithms for reducing the variance
- **Results: Substantial improvements**

.. though in poker far from being useful (so-far)

Conclusions

- Regret-minimization framework $\sim$ non-stationary environments
- Bandit algorithms:
 - Role of payoff-estimate variance
 - Algorithms for reducing the variance
- Results: Substantial improvements
.. though in poker far from being useful (so-far)

Future Work

- **Non-oblivious (adapting) opponents**
- Use C_t in expert selection
- External-regret vs. internal-regret

Future Work

- Non-oblivious (adapting) opponents
- Use C_t in expert selection
- External-regret vs. internal-regret

Future Work

- Non-oblivious (adapting) opponents
- Use C_t in expert selection
- External-regret vs. internal-regret

Questions?

More information: www.sztaki.hu/~szcsaba