Sequential Decision

Session 6: Pure exploration and learning from preferences

Igor Colin

October 2026

ESIEE Paris

Today

Part	Content	Time
Lecture	best-arm identification, Bradley-Terry, Elo	1 h
Tutorial 6	exercises 1 and 2: sequential halving, Bradley-Terry and Elo	40 min
Lab 6	sequential halving, ranking models from duels	1 h 20

Two changes of objective

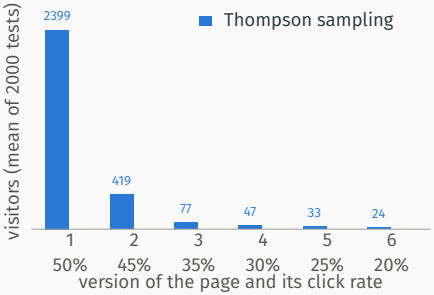
- Only the final decision counts, not the rewards collected on the way
- The feedback is a comparison, not a reward

Where it is used

- A/B/n tests, hyperparameter search
- Chatbot arenas, reward models of RLHF

Pure exploration

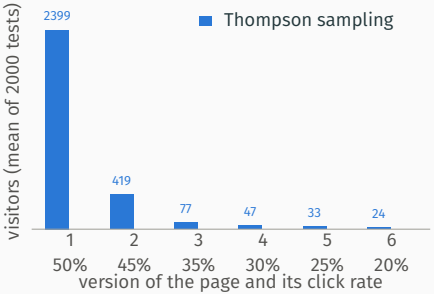
Identifying the best arm



A company tests 6 versions of a page (click rates unknown) on $n = 3000$ visitors, then keeps one for good.

- The clicks during the test do not matter: the criterion is the probability of keeping a wrong version

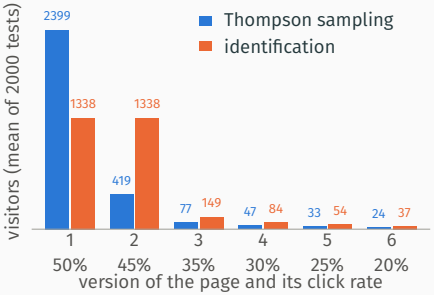
Identifying the best arm



A company tests 6 versions of a page (click rates unknown) on $n = 3000$ visitors, then keeps one for good.

- The clicks during the test do not matter: the criterion is the probability of keeping a wrong version
- Thompson sampling (regret): few visitors on 2, so 1 and 2 are badly separated. Wrong version: 1.6% of the tests

Identifying the best arm



A company tests 6 versions of a page (click rates unknown) on $n = 3000$ visitors, then keeps one for good.

- The clicks during the test do not matter: the criterion is the probability of keeping a wrong version
- Thompson sampling (regret): few visitors on 2, so 1 and 2 are badly separated. Wrong version: 1.6% of the tests
- Identification: as many visitors on 2 as on 1, few on the bad versions. Wrong version: 0.5% (uniform, 500 each: 5.3%)

Fixed budget and fixed confidence

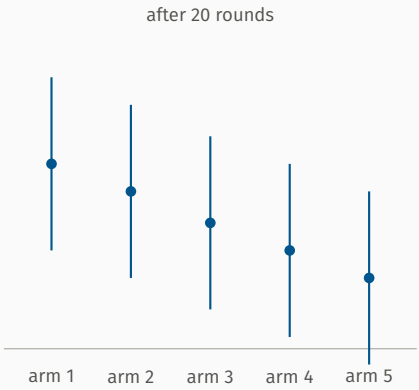
	Fixed budget	Fixed confidence
Given	a number of pulls n	an error level δ
Goal	minimize $\mathbb{P}$(wrong arm)	stop early, with $\mathbb{P}$(wrong arm) $\leq \delta$
Example	A/B test on n visitors	clinical trial with a stopping rule
Algorithm	sequential halving	successive elimination

Two arms with gap Δ, each pulled m times (tutorial, exercise 1)

$$\mathbb{P}(\text{the worse arm has the larger empirical mean}) \leq e^{-m\Delta^2/2}$$

For $\Delta = 0.1$ and an error of 5%: $m = 600$ pulls of each arm.

Successive elimination

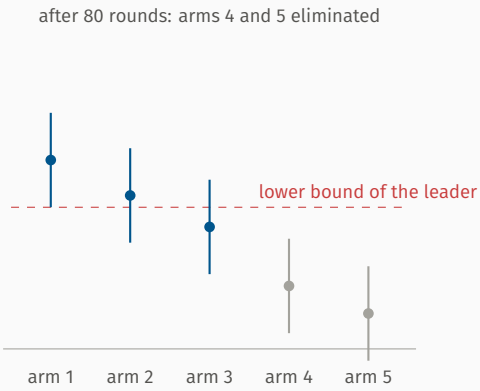


Successive elimination

At each round, pull every remaining arm once. Eliminate an arm when its upper confidence bound falls below the lower confidence bound of the leader. Stop when one arm remains.

- Confidence intervals from Hoeffding, with a level that accounts for all arms and all rounds
- Bad arms leave early; the budget goes to the close ones
- Number of pulls of order $\sum_a \log(K/\delta)/\Delta_a^2$

Successive elimination

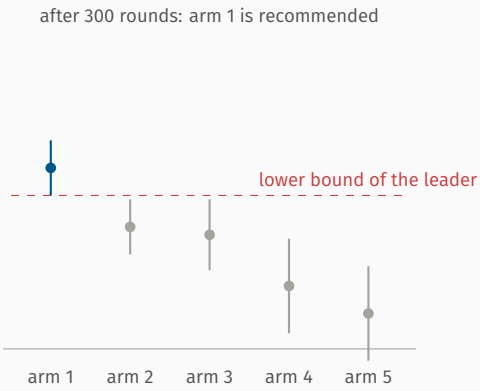


Successive elimination

At each round, pull every remaining arm once. Eliminate an arm when its upper confidence bound falls below the lower confidence bound of the leader. Stop when one arm remains.

- Confidence intervals from Hoeffding, with a level that accounts for all arms and all rounds
- Bad arms leave early; the budget goes to the close ones
- Number of pulls of order $\sum_a \log(K/\delta)/\Delta_a^2$

Successive elimination

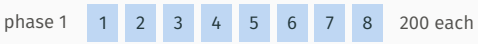


Successive elimination

At each round, pull every remaining arm once. Eliminate an arm when its upper confidence bound falls below the lower confidence bound of the leader. Stop when one arm remains.

- Confidence intervals from Hoeffding, with a level that accounts for all arms and all rounds
- Bad arms leave early; the budget goes to the close ones
- Number of pulls of order $\sum_a \log(K/\delta)/\Delta_a^2$

Sequential halving

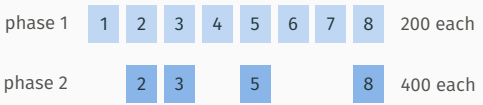


Sequential halving (Karnin et al., 2013)

Budget n, $L = \lceil \log_2 K \rceil$ phases. In each phase, pull every remaining arm $\lfloor n/(|S|L) \rfloor$ times, and keep the better half, using the rewards of this phase only.

- Each phase costs n/L pulls
- The last comparison uses $n/(2L)$ pulls per arm: 800 for $K = 8$, $n = 4800$, against 600 with uniform allocation
- No parameter to tune

Sequential halving

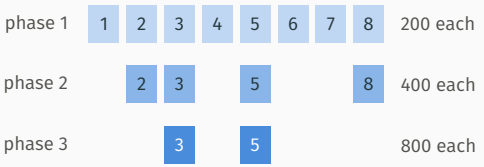


Sequential halving (Karnin et al., 2013)

Budget n, $L = \lceil \log_2 K \rceil$ phases. In each phase, pull every remaining arm $\lfloor n/(|S|L) \rfloor$ times, and keep the better half, using the rewards of this phase only.

- Each phase costs n/L pulls
- The last comparison uses $n/(2L)$ pulls per arm: 800 for $K = 8$, $n = 4800$, against 600 with uniform allocation
- No parameter to tune

Sequential halving

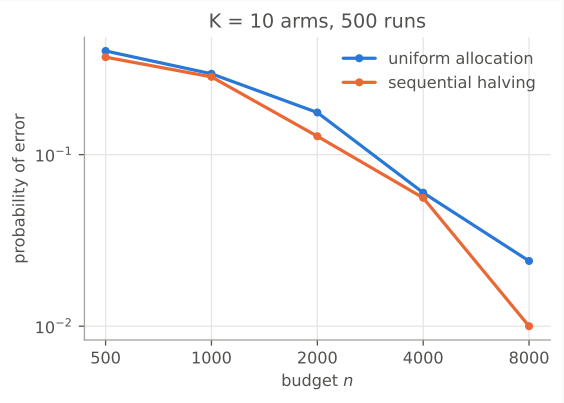


Sequential halving (Karnin et al., 2013)

Budget n, $L = \lceil \log_2 K \rceil$ phases. In each phase, pull every remaining arm $\lfloor n/(|S|L) \rfloor$ times, and keep the better half, using the rewards of this phase only.

- Each phase costs n/L pulls
- The last comparison uses $n/(2L)$ pulls per arm: 800 for $K = 8$, $n = 4800$, against 600 with uniform allocation
- No parameter to tune

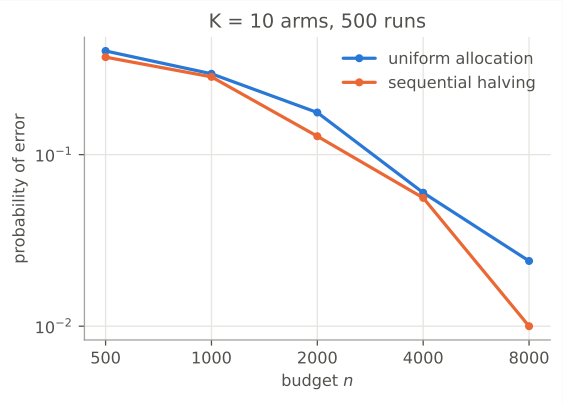
Uniform allocation and sequential halving



10 Bernoulli arms, best mean 0.5, second 0.45, worst 0.2.

- The error decreases exponentially with the budget, for both
- At $n = 8000$: 1% for sequential halving, 2.4% for uniform allocation
- The gain grows with the number of arms that are easy to eliminate

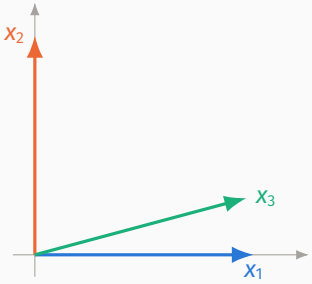
Uniform allocation and sequential halving



10 Bernoulli arms, best mean 0.5, second 0.45, worst 0.2.

- The error decreases exponentially with the budget, for both
- At $n = 8000$: 1% for sequential halving, 2.4% for uniform allocation
- The gain grows with the number of arms that are easy to eliminate
- Regret algorithms, recommending their most pulled arm, can also do well; but they have no stopping rule and no guarantee on the error

Pure exploration in linear bandits

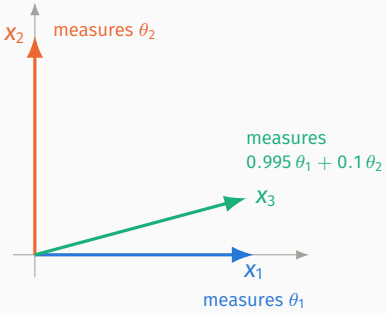


angle of x_3 exaggerated

Example of the $\mathcal{XY}$ design: $x_1 = (1, 0)$, $x_2 = (0, 1)$, $x_3 = (\cos 0.1, \sin 0.1)$. If $\theta^\star = (1, 0)$, the means are 1, 0 and 0.995.

- Hard part: telling x_1 from x_3 (gap 0.005). Regret: pull x_1 and x_3, never x_2

Pure exploration in linear bandits

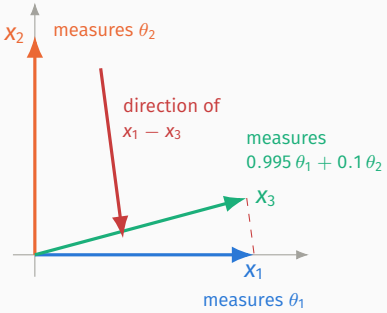


angle of x_3 exaggerated

Example of the $\mathcal{XY}$ design: $x_1 = (1, 0)$, $x_2 = (0, 1)$, $x_3 = (\cos 0.1, \sin 0.1)$. If $\theta^\star = (1, 0)$, the means are 1, 0 and 0.995.

- Hard part: telling x_1 from x_3 (gap 0.005). Regret: pull x_1 and x_3, never x_2
- A pull of x is a noisy measurement of $\langle \theta^\star, x \rangle$, with $\theta^\star = (\theta_1, \theta_2)$ unknown

Pure exploration in linear bandits

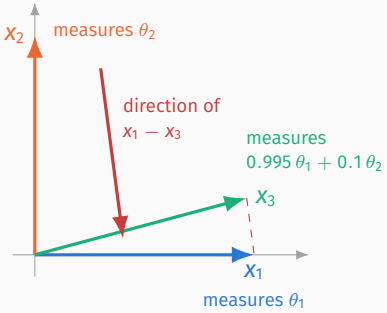


angle of x_3 exaggerated

Example of the $\mathcal{XY}$ design: $x_1 = (1, 0)$, $x_2 = (0, 1)$, $x_3 = (\cos 0.1, \sin 0.1)$. If $\theta^\star = (1, 0)$, the means are 1, 0 and 0.995.

- Hard part: telling x_1 from x_3 (gap 0.005). Regret: pull x_1 and x_3, never x_2
- A pull of x is a noisy measurement of $\langle \theta^\star, x \rangle$, with $\theta^\star = (\theta_1, \theta_2)$ unknown
- The gap is $\langle \theta^\star, x_1 - x_3 \rangle = 0.005\,\theta_1 - 0.1\,\theta_2$: its sign depends on θ_2, which x_3 sees with a factor 0.1 only

Pure exploration in linear bandits

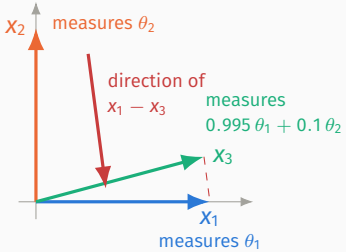


angle of x_3 exaggerated

Example of the $\mathcal{XY}$ design: $x_1 = (1, 0)$, $x_2 = (0, 1)$, $x_3 = (\cos 0.1, \sin 0.1)$. If $\theta^\star = (1, 0)$, the means are 1, 0 and 0.995.

- Hard part: telling x_1 from x_3 (gap 0.005). Regret: pull x_1 and x_3, never x_2
- A pull of x is a noisy measurement of $\langle \theta^\star, x \rangle$, with $\theta^\star = (\theta_1, \theta_2)$ unknown
- The gap is $\langle \theta^\star, x_1 - x_3 \rangle = 0.005\,\theta_1 - 0.1\,\theta_2$: its sign depends on θ_2, which x_3 sees with a factor 0.1 only
- x_2 measures θ_2 directly: the worst arm is the most informative on the gap between the two best

Designing the pulls for the gaps



angle of x_3 exaggerated

Allocation of the pulls	Pulls needed
half on x_1, half on x_3 (regret)	$\times 360$
uniform on the three arms	$\times 2.7$
half on x_1, half on x_2	$\times 1.8$
95% on x_2, 5% on x_1	$\times 1$

pulls needed to know the sign of the gap ($\times 1$: best allocation)

$\mathcal{XY}$ design (Soare, Lazaric and Munos, 2014)

Choose how often to pull each arm so that every *difference* of means is estimated precisely: the differences decide which arm is the best.

Adaptive designs

The directions that matter depend on which arms are close to the best, unknown at the start.

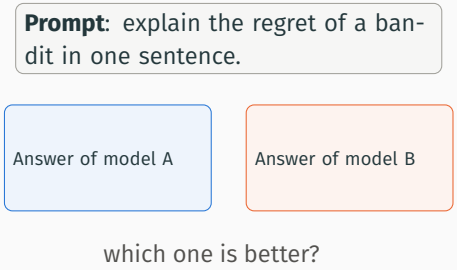
- $\mathcal{XY}$**-adaptive** (Soare, Lazaric and Munos, 2014): in phases, recompute the design on the differences between the arms that are not eliminated yet
- **LinGapE** (Xu, Honda and Sugiyama, 2018): at each round, take the current leader and its most dangerous competitor, and pull the arm that most reduces the uncertainty on *their* gap; stop when the leader is better with high confidence
- **RAGE** (Fiez, Jain, Jamieson and Nowak, 2019): successive elimination, where each round plays an $\mathcal{XY}$ design on the remaining arms, with a precision doubled from one round to the next

Informative is not promising

Minimizing the regret pulls the arms that look good; identifying the best arm pulls the arms that tell the good ones apart.

Learning from comparisons

Feedback by comparisons



- Humans grade answers inconsistently, but compare them reliably
- Chatbot Arena (LMSYS): millions of duels between anonymous models, voted by users
- RLHF: annotators compare two answers of the same model, and a reward model is trained on these comparisons (session 9)

Feedback by comparisons

Prompt: explain the regret of a bandit in one sentence.

Answer of model A

Answer of model B

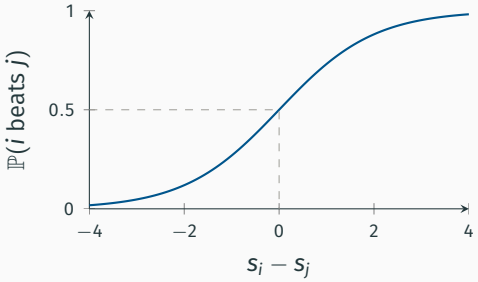
which one is better?

A wins

- Humans grade answers inconsistently, but compare them reliably
- Chatbot Arena (LMSYS): millions of duels between anonymous models, voted by users
- RLHF: annotators compare two answers of the same model, and a reward model is trained on these comparisons (session 9)
- Each duel is one bit: which model is better, and by how much?

The Bradley-Terry model

One score per model instead of one probability per pair: K numbers instead of $K(K-1)/2$, and every duel informs on all the scores.



Bradley-Terry (1952)

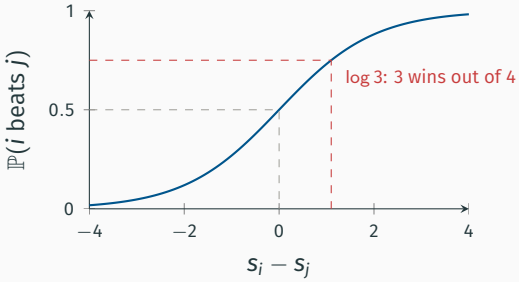
Each model i has a score s_i, and

$$\mathbb{P}(i \text{ beats } j) = \sigma(s_i - s_j) = \frac{1}{1 + e^{-(s_i - s_j)}} \ .$$

- Equal scores: a coin flip. Only the differences of scores matter
- Model i answers with quality s_i + noise, and the better answer wins: with a Gumbel noise, this gives exactly $\sigma(s_i - s_j)$

The Bradley-Terry model

One score per model instead of one probability per pair: K numbers instead of $K(K-1)/2$, and every duel informs on all the scores.



Bradley-Terry (1952)

Each model i has a score s_i, and

$$\mathbb{P}(i \text{ beats } j) = \sigma(s_i - s_j) = \frac{1}{1 + e^{-(s_i - s_j)}} \ .$$

- Equal scores: a coin flip. Only the differences of scores matter
- Model i answers with quality s_i + noise, and the better answer wins: with a Gumbel noise, this gives exactly $\sigma(s_i - s_j)$

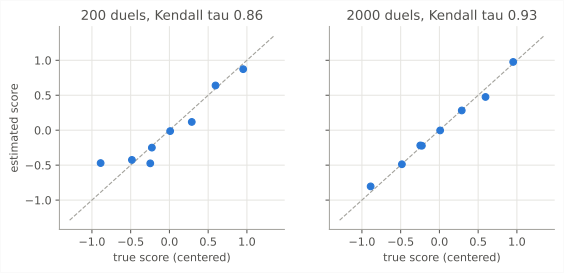
Maximum likelihood

With W_{ij} the number of wins of i against j:

$$\ell(s) = \sum_{i \neq j} W_{ij} \log \sigma(s_i - s_j) \ ,$$

$$\frac{\partial \ell}{\partial s_i} = \underbrace{\sum_j W_{ij}}_{\text{wins}} - \underbrace{\sum_j N_{ij}\, \sigma(s_i - s_j)}_{\text{expected wins}} \ .$$

- Concave: gradient ascent finds the maximum
- A small penalty $\frac{\lambda}{2}\|s\|^2$ fixes the constant and handles undefeated models



8 models, duels between random pairs (lab)

A logistic regression

The duel i vs j is a sample with features $x = e_i - e_j$ ($+1$ for i, -1 for j) and label 1 if i wins, since $\sigma(\langle s, x \rangle) = \sigma(s_i - s_j)$.

Updating the scores after each duel

Chess: thousands of games a day, players whose level changes. Instead of refitting all the scores, update the ratings after each game, from that game only.

Elo update (chess, from the 1960s)

Player i has a rating r_i. After a duel i vs j, with $y = 1$ if i wins, 0 otherwise:

$$r_i \leftarrow r_i + k\big(y - \sigma(r_i - r_j)\big) \ , \qquad r_j \leftarrow r_j - k\big(y - \sigma(r_i - r_j)\big) \ ,$$

Updating the scores after each duel

Chess: thousands of games a day, players whose level changes. Instead of refitting all the scores, update the ratings after each game, from that game only.

Elo update (chess, from the 1960s)

Player i has a rating r_i. After a duel i vs j, with $y = 1$ if i wins, 0 otherwise:

$$r_i \leftarrow r_i + k\big(y - \sigma(r_i - r_j)\big) \ , \qquad r_j \leftarrow r_j - k\big(y - \sigma(r_i - r_j)\big) \ ,$$

- Result minus prediction: if $\sigma(r_i - r_j) = 0.9$, a win gives $+0.1\,k$, a loss $-0.9\,k$

Updating the scores after each duel

Chess: thousands of games a day, players whose level changes. Instead of refitting all the scores, update the ratings after each game, from that game only.

Elo update (chess, from the 1960s)

Player i has a rating r_i. After a duel i vs j, with $y = 1$ if i wins, 0 otherwise:

$$r_i \leftarrow r_i + k\big(y - \sigma(r_i - r_j)\big) \quad , \qquad r_j \leftarrow r_j - k\big(y - \sigma(r_i - r_j)\big) \quad ,$$

- Result minus prediction: if $\sigma(r_i - r_j) = 0.9$, a win gives $+0.1\,k$, a loss $-0.9\,k$
- A stochastic gradient step, with step k, on the log-likelihood of this duel only (tutorial, exercise 2)

Updating the scores after each duel

Chess: thousands of games a day, players whose level changes. Instead of refitting all the scores, update the ratings after each game, from that game only.

Elo update (chess, from the 1960s)

Player i has a rating r_i. After a duel i vs j, with $y = 1$ if i wins, 0 otherwise:

$$r_i \leftarrow r_i + k\big(y - \sigma(r_i - r_j)\big) \quad , \qquad r_j \leftarrow r_j - k\big(y - \sigma(r_i - r_j)\big) \quad ,$$

- Result minus prediction: if $\sigma(r_i - r_j) = 0.9$, a win gives $+0.1\,k$, a loss $-0.9\,k$
- A stochastic gradient step, with step k, on the log-likelihood of this duel only (tutorial, exercise 2)
- Chess scale: $R = 1500 + 400\,r/\log 10$ and $K = 400\,k/\log 10$; 200 points of difference mean 76% of wins

Elo by hand

Tutorial, exercise 2: three players, $k = 1$

Duel	A	B	C
start	0	0	0

Elo by hand

Tutorial, exercise 2: three players, $k = 1$

Duel	A	B	C
start	0	0	0
A beats B	0.50	-0.50	0

- Equal ratings: $\sigma(0) = 1/2$, the winner gets $+0.5$

Elo by hand

Tutorial, exercise 2: three players, $k = 1$

Duel	A	B	C
start	0	0	0
A beats B	0.50	-0.50	0
B beats C	0.50	0.12	-0.62

- Equal ratings: $\sigma(0) = 1/2$, the winner gets $+0.5$
- B beats C with ratings -0.50 and 0: a surprise, $+0.62$

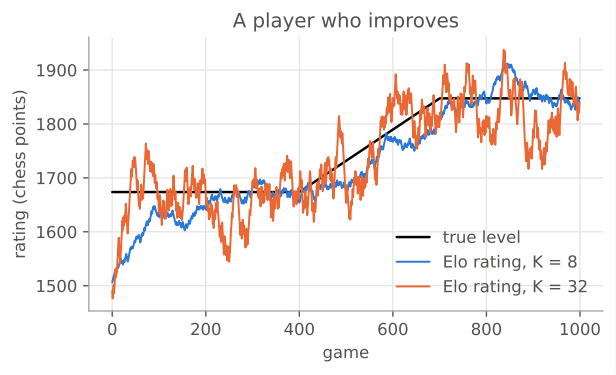
Elo by hand

Tutorial, exercise 2: three players, $k = 1$

Duel	A	B	C
start	0	0	0
A beats B	0.50	-0.50	0
B beats C	0.50	0.12	-0.62
A beats C	0.75	0.12	-0.87

- Equal ratings: $\sigma(0) = 1/2$, the winner gets $+0.5$
- B beats C with ratings -0.50 and 0: a surprise, $+0.62$
- A beats C, expected: only $+0.25$
- The ratings depend on the order of the duels: the maximum likelihood on the same three duels does not (lab)

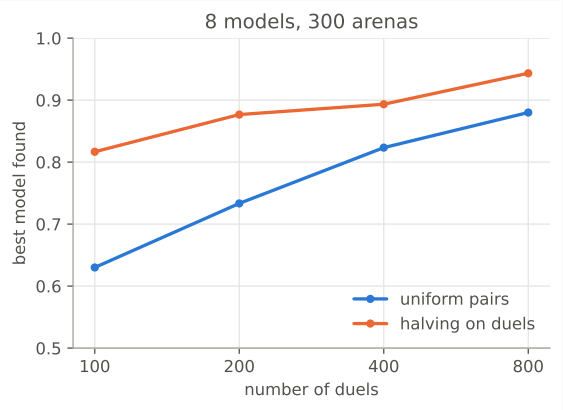
Choosing the step



A new player starts at 1500, plays opponents of their rating level, and improves after game 400.

- Small K: a stable rating, but 200 games to reach the true level, and a delay when the player improves
- Large K: follows quickly, but about ± 100 points of noise
- In practice: a large K for the first games, smaller afterwards (FIDE: 40, then 20, then 10)

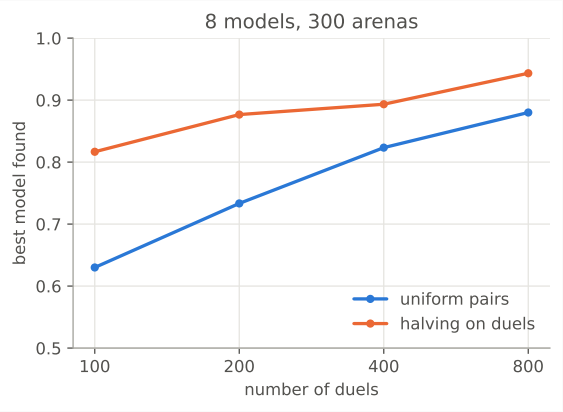
Choosing the duels



Each duel costs one annotation. To find the best of 8 models:

- **Uniform pairs**: the weakest models keep being compared
- **Halving on duels**: sequential halving on the models, with a Bradley-Terry fit after each phase
- 100 duels with halving do as well as 400 uniform ones

Choosing the duels



Each duel costs one annotation. To find the best of 8 models:

- **Uniform pairs**: the weakest models keep being compared
- **Halving on duels**: sequential halving on the models, with a Bradley-Terry fit after each phase
- 100 duels with halving do as well as 400 uniform ones
- Cyclic preferences (A beats B beats C beats A) have no scores, and no best model (lab)

Summary

Problem	Method	Key point
Best arm, fixed budget	sequential halving	no tuning, L phases
Best arm, fixed confidence	successive elimination	stops when sure
Best arm, linear	$\mathcal{XY}$ design, LinGapE, RAGE	pull what informs on the gaps
Scores from duels	Bradley-Terry	a logistic regression
Online ratings	Elo	one gradient step per duel

Tutorial 6

- Sequential halving
- Bradley-Terry and Elo

Lab 6

- Uniform allocation against sequential halving
- Bradley-Terry, Elo, choice of duels

Reading

- Lattimore and Szepesvári, chapter 33
- Chiang et al., *Chatbot Arena*, 2024