Ilya Kuzovkin

Review of the article

Multi-armed Bandit Algorithms and Empirical Evaluation

Joannès Vermorel[1], Mehryar Mohri[2]

Machine Learning: ECML 2005

[1]École Normale Supérieure, Paris

[2]Courant Institute of Mathematical Sciences, New York

The Problem

Limited number
of trials

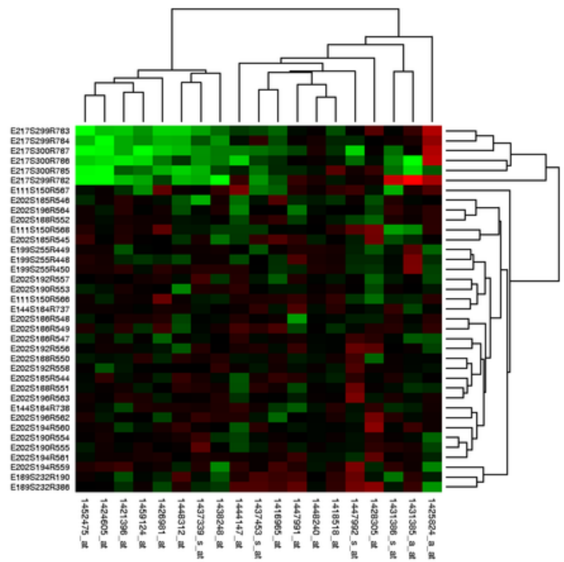
- Money
- Time
- …

The Problem

Limited number of trials

- Money
- Time
- …

Valuable knowledge to gain



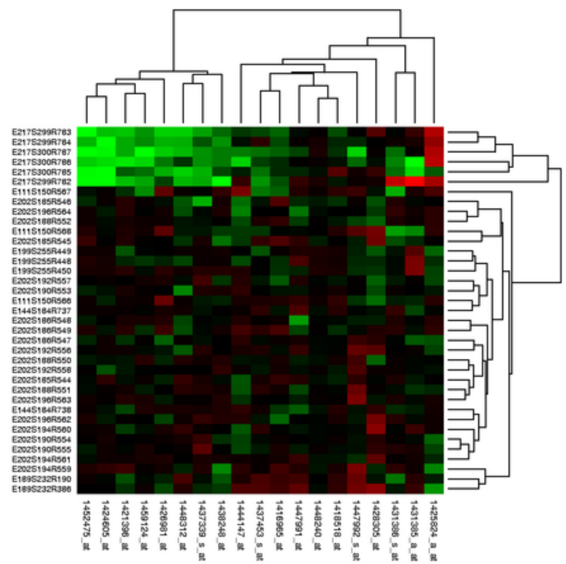
Reward

The Problem

Limited number of trials

- Money
- Time
- …

Valuable knowledge to gain



Reward

The Problem

Each time you have to choose …

Waste resources and hope
to find something valuable

The Problem

Each time you have to choose …

or

Waste resources and hope to find something valuable

Exploit knowledge you already have

The Problem

Each time you have to choose …

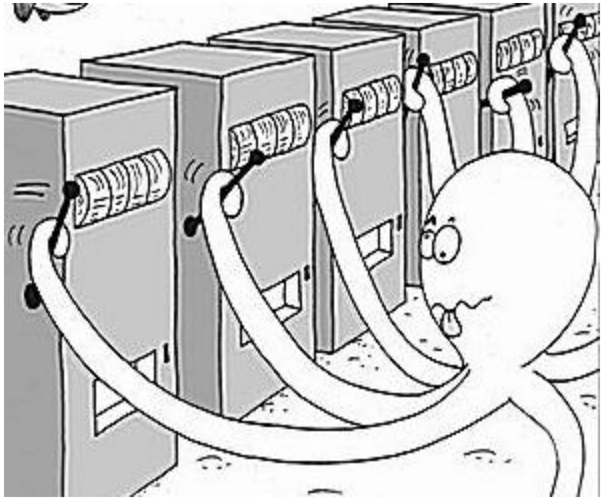
or

Waste resources and hope to find something valuable

Exploit knowledge you already have

Exploitation-Exploration Trade-off

Formalization



Multi-armed bandit:
- Each arm has independent distribution behind it
- Player can explore new levers
- Or draw the ones he already knows to be profitable
- Each time he gets some numerical reward

Glossary:
- Horizon – remaining number of times you can pull a lever
- Reward – function you want to maximize
- Regret – difference between optimal and collected reward
- Zero-Regret Strategy – asymptotically gives regret = 0

Types:
- Opaque – only one reward is observed at each round
- Transparent – all rewards are observed

Multi-Armed Bandits: Strategies

Approximate

a) ε- greedy
   - ε- first
   - ε- decreasing

a) SoftMax
   - Decreasing
   - Exp3

b) Interval Estimation

a) Price of Knowledge and Estimated Reward

Optimal

There exists number of strategies theoretically proven to be optimal for certain distributions or other conditions

a)ε- greedy

ε – first

1. Explore ε* T (#rounds)
2. Exploit the rest

Find a-optimal arm with
probability at least 1−δwith

$$\mathcal{O}\left(\frac{K}{\alpha^2}\log\left(\frac{K}{\delta}\right)\right)$$

pulls, where K is number of arms

Not a zero-regret strategy.

a)ε- greedy

ε – first

1. Explore ε* T (#rounds)
2. Exploit the rest

Find a-optimal arm with probability at least 1−δwith

$$\mathcal{O}\left(\frac{K}{\alpha^2}\log\left(\frac{K}{\delta}\right)\right)$$

pulls, where K is number of arms

Not a zero-regret strategy.

ε – decreasing

At each round there is ε_t probability to pull random lever

ε_t is smaller with each round

With carefully chosen parameters regret is

$$\mathcal{O}(\log(T))$$

Zero-regret strategy.

b) SoftMax

Chooses lever according to probability distribution (family of methods is called *probability matching strategies*)

b) SoftMax

Chooses lever according to probability distribution (family of methods is called *probability matching strategies*)

$$p_k = e^{\widehat{\mu}_k/\tau} / \sum_{i=1}^{n} e^{\widehat{\mu}_i/\tau}$$

$\boldsymbol{k}$ is lever index
$\widehat{\mu}_k$ is estimated distribution mean for the lever
τ is temperature (constant or decreasing)

b) SoftMax

Chooses lever according to probability distribution (family of methods is called *probability matching strategies*)

$$p_k = e^{\widehat{\mu}_k/\tau} / \sum_{i=1}^{n} e^{\widehat{\mu}_i/\tau}$$

k is lever index
$\widehat{\mu}_k$ is estimated distribution mean for the lever
τ is temperature (constant or decreasing)

Regret guarantee is same as for ε- decreasing

$$\mathcal{O}(\log(T))$$

b) SoftMax : Exp3

"Exponential Weight Algorithm for Exploration and Exploitation"

b) SoftMax : Exp3

"Exponential Weight Algorithm for Exploration and Exploitation"

$$p_k(t) = (1-\gamma)\frac{w_k(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$$

b) SoftMax : Exp3

"Exponential Weight Algorithm for Exploration and Exploitation"

$$p_k(t) = (1-\gamma)\frac{w_k(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$$

If lever *j* was just pulled $\quad w_k(t) = w_k(t-1)e^{\gamma\frac{r_k(t-1)}{p_k(t-1)K}}$

b) SoftMax : Exp3

"Exponential Weight Algorithm for Exploration and Exploitation"

$$p_k(t) = (1-\gamma)\frac{w_k(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$$

If lever *j* was just pulled $\quad w_k(t) = w_k(t-1)e^{\gamma \frac{r_k(t-1)}{p_k(t-1)K}}$

else $\quad w_k(t) = w_k(t-1)$

b) SoftMax : Exp3

"Exponential Weight Algorithm for Exploration and Exploitation"

$$p_k(t) = (1-\gamma)\frac{w_k(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$$

If lever *j* was just pulled $$w_k(t) = w_k(t-1)e^{\gamma \frac{r_k(t-1)}{p_k(t-1)K}}$$

else $$w_k(t) = w_k(t-1)$$

The main idea is to divide lever reward by lever probability.
In such way if discover unexpectedly good lever we will pull it again.

b) SoftMax : Exp3

"Exponential Weight Algorithm for Exploration and Exploitation"

$$p_k(t) = (1-\gamma)\frac{w_k(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$$

If lever *j* was just pulled $\quad w_k(t) = w_k(t-1)e^{\gamma \frac{r_k(t-1)}{p_k(t-1)K}}$

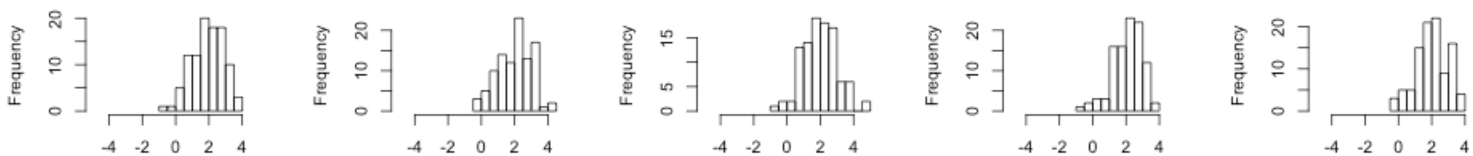
else $\quad w_k(t) = w_k(t-1)$

The main idea is to divide lever reward by lever probability.
In such way if discover unexpectedly good lever we will pull it again.

Estimated regret is $\mathcal{O}(\sqrt{KT\log(K)})$

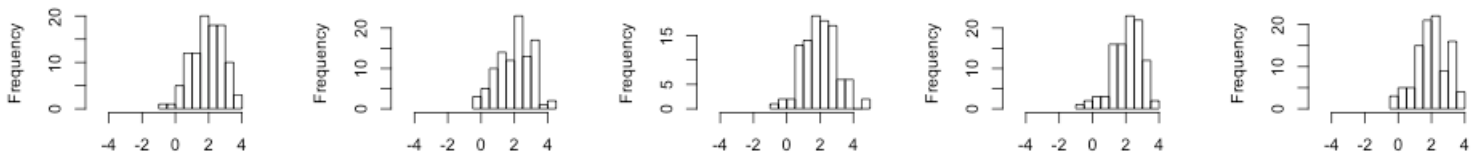
c) Interval Estimation

1. Each lever is given *optimistic reward estimate* within certain confidence interval



c) Interval Estimation

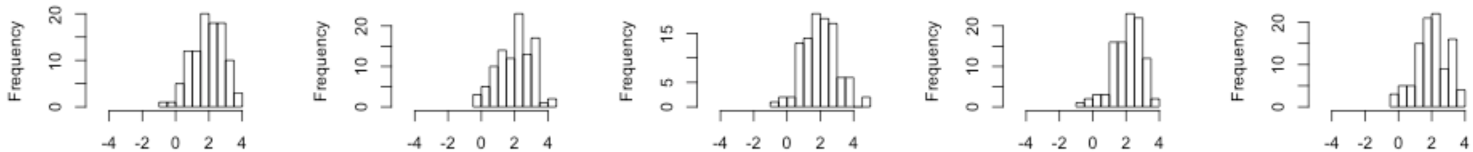
1. Each lever is given *optimistic reward estimate* within certain confidence interval



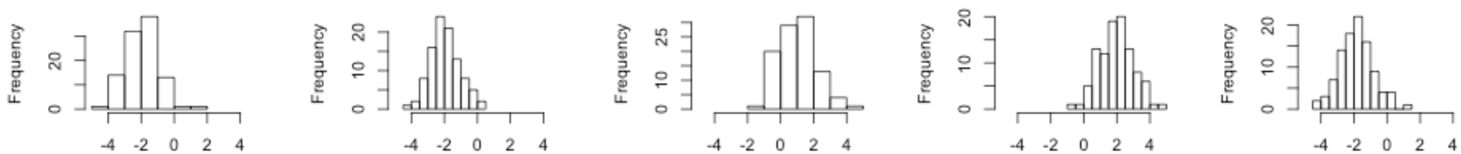
1. Infrequently observed levers will have over-estimated mean, which will lead to further exploration
2. Lever with highest reward mean upper bound is chosen

c) Interval Estimation

1. Each lever is given *optimistic reward estimate* within certain confidence interval



1. Infrequently observed levers will have over-estimated mean, which will lead to further exploration
2. Lever with highest reward mean upper bound is chosen
3. With each pull *optimistic mean* comes closer to true mean



No theoretical results on regret estimation known.
Zero-Regret with careful choice of parameters.

d) Price of Knowledge and Estimated Reward

- Price: Quantify uncertainty in same units as reward

What is better	Reward A
or	Reward B + Information gain C

- Estimate unobserved lever's distributions from observed ones
- Take horizon into account

Is a zero-regret strategy.

Evaluation: Datasets

a) Randomly generated

- 10,000 trials
- 1000 levers
- Rewards have normal distribution with random mean and standard deviation. Both in range (0,1)

- Task is to maximize reward

b) ULR Retrieval Latency

- Data retrieval with redundant sources
- One page = one lever
- Latency = (negative) reward

- Task is to minimize latency

Evaluation: Results

Strategies	R-100	R-1k	R-10k	N-130	N-1.3k
Poker	0.787	0.885	0.942	203	132
ϵ-greedy, 0.05	0.712	0.855	0.936	733	431
ϵ-greedy, 0.10	0.740	0.858	0.916	731	453
ϵ-greedy, 0.15	0.746	0.842	0.891	715	474
ϵ-first, 0.05	0.732	0.906	0.951	735	414
ϵ-first, 0.10	0.802	0.893	0.926	733	421
ϵ-first, 0.15	0.809	0.869	0.901	725	411
ϵ-decreasing, 1.0	0.755	0.805	0.851	738	411
ϵ-decreasing, 5.0	0.785	0.895	0.934	715	413
ϵ-decreasing, 10.0	0.736	0.901	0.949	733	417
LeastTaken, 0.05	0.750	0.782	0.932	747	420
LeastTaken, 0.1	0.750	0.791	0.912	738	432
LeastTaken, 0.15	0.757	0.784	0.892	734	441
SoftMax, 0.05	0.747	0.801	0.855	728	410
SoftMax, 0.10	0.791	0.853	0.887	729	409
SoftMax, 0.15	0.691	0.761	0.821	727	410
Exp3, 0.2	0.506	0.501	0.566	726	541
Exp3, 0.3	0.506	0.504	0.585	725	570
Exp3, 0.4	0.506	0.506	0.594	728	599
GaussMatch	0.559	0.618	0.750	327	194
IntEstim, 0.01	0.725	0.806	0.844	305	200
IntEstim, 0.05	0.736	0.814	0.851	287	189
IntEstim, 0.10	0.734	0.791	0.814	276	190

Evaluation: Conclusions

- ε- greedy can be very good if parameters are chosen correctly.
- SoftMax is performing well, however it's variation Exp3 shows worst results over all. This can be explained by the fact that Exp3 was designed to optimize asymptotic behavior.
- On the real dataset POKER works best, which seem to justify the decisions authors made when designing it. And it is non-parametric.
- Dynamic estimation of the level of exploration seems to perform better.
- Empirical results from random data are not transferrable to real-world data.

Applications

- Clinical trials (William, 2009)
- Adaptive routing in networks
- Document ranking based on user response (Radlinski, 2008)
- Task assignment to UAV (Unmanned Aerial Vehicles) (Le Ny, 2006)
- Budget allocation between projects (Gittins, 1989)
- Bioinformatics ?

Thanks!

More on this topic

- Markov Decision Process – more general approach, which includes bandit formalization
- Feynman's Restaurant Problem – illustrative example with optimal strategy and proofhttp://www.feynmanlectures.info/exercises/Feynmans_restaurant_problem.html
- Burnetas, AN; Katehakis, MN "Optimal adaptive policies for Markov decision processes" 1997