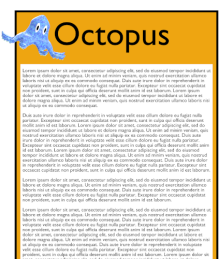
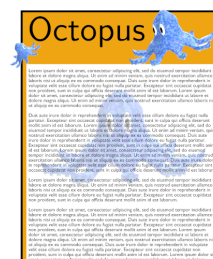
Machine Learning 2.12: Multi-armed Bandits

Tom S. F. Haines
T.S.F.Haines@bath.ac.uk

Design

- Scenario: Changing design of website, multiple proposals
- Anything from tiny tweak to complete redesign





UNIVERSITY OF BATH

Decision

- How to choose specific design?
- Ask people? (user study)
  (designer, yourself, users, random people on street. . .)
- . . . but how reliable is that?
- . . . expensive!

Measurement

- Objectives:
  - User time on website?
  - Pages visited?
  - Adverts clicked?
  - Sales made?

(all can be tracked using cookies/Javascript)

UNIVERSITY OF BATH

Measurement

- Objectives:
  - User time on website?
  - Pages visited?
  - Adverts clicked?
  - Sales made?

  (all can be tracked using cookies/Javascript)

- These are noisy measurements!
- Highly biased, e.g. click rate of adverts $\approx \frac{1}{1000}$
  (usually you measure zero)

UNIVERSITY OF BATH

A/B testing

- Show version A to half of users, version B to other half (use cookies)
- After w weeks pick version with best score

UNIVERSITY OF BATH

A/B testing

- Show version A to half of users, version B to other half (use cookies)
- After w weeks pick version with best score

- Weaknesses:
  - One change at a time
  - w may be too long / too short
  - Combinatorics (many changes) a problem

UNIVERSITY OF BATH

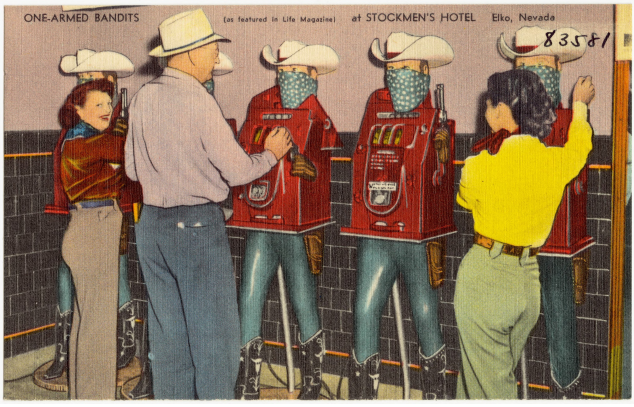
Goal

- Automate this!

- Many changes at a time
- Can add new variants at any time
- Dynamically adapt w (weeks)

Multi-armed bandits

- Called a **multi-armed bandit** problem
- Row of *one-armed bandits*
- Differing levels of reward (unknown)
- Which lever to pull?
- Machines correspond to designs



UNIVERSITY OF BATH

Exploration vs exploitation

- Two competing goals:
  - **Exploration** – learn reward of pulling each lever
  - Only at start
  - **Exploitation** – pull best level to maximise reward
  - After exploration "finished"
- This model can be mapped to much of life!
- Key to **reinforcement learning**

Three levers

Red lever:

$P(\text{reward} = 100) = 0.05$

$P(\text{reward} = 0) = 0.95$

$\therefore$

$\mathbb{E}[\text{reward}] = 5$

$(0.05 \times 100 + 0.95 \times 0)$

Green lever:

$P(\text{reward} = 1200) = 0.01$

$P(\text{reward} = 0) = 0.99$

$\therefore$

$\mathbb{E}[\text{reward}] = 12$

$(0.01 \times 1200 + 0.99 \times 0)$

Blue lever:

$P(\text{reward} = 100) = 0.1$

$P(\text{reward} = 0) = 0.9$

$\therefore$

$\mathbb{E}[\text{reward}] = 10$

$(0.1 \times 100 + 0.9 \times 0)$

UNIVERSITY OF BATH

Stupid strategies

- Pull levers at **random**
- Pure **exploration**
- Expected reward:

$$\mathbb{E}\left[\text{reward}\right] = \frac{5 + 12 + 10}{3} = 9$$

- Often much worse
  (many dud levers)
- Missing the point!

Stupid strategies

- Pull levers at **random**
- Pure **exploration**

- Expected reward:

$$\mathbb{E}\left[\text{reward}\right] = \frac{5 + 12 + 10}{3} = 9$$

- Often much worse
  (many dud levers)

- Missing the point!

- Pull lever with **maximum expected reward**
- Pure **exploitation**

- Don't know expectation ∴
  1. Pull at random until a reward occurs...
  2. Pull that one lever forever (**stuck!**)

- Expected reward:

$$\begin{aligned}\mathbb{E}\left[\text{reward}\right] &= \frac{0.05 \times 5 + 0.01 \times 12 + 0.1 \times 10}{0.05 + 0.01 + 0.1} \\ &= 8.5625\end{aligned}$$

(over many runs...)

- Fails!

Epsilon-greedy

- Combine them = Epsilon-greedy:
  - With probability ϵ select at random (exploration)
  - With probability $1 - \epsilon$ select maximum expected reward (exploitation)

$$= \underset{m \in \mathbf{M}}{\operatorname{argmax}} \left(\frac{1}{N_m} \sum_{i=1}^{N_m} r_{i,m} \right)$$

where

- m = a machine/lever
- N_m = how many times it has been queried in the past
- $r_{i,m}$ = the reward from the i^{th} time machine m was queried

UNIVERSITY OF BATH

Epsilon-greedy

- Combine them = Epsilon-greedy:
  - With probability ϵ select at random (exploration)
  - With probability $1 - \epsilon$ select maximum expected reward (exploitation)

$$= \underset{m \in \mathbf{M}}{\operatorname{argmax}} \left(\frac{1}{N_m} \sum_{i=1}^{N_m} r_{i,m} \right)$$

where

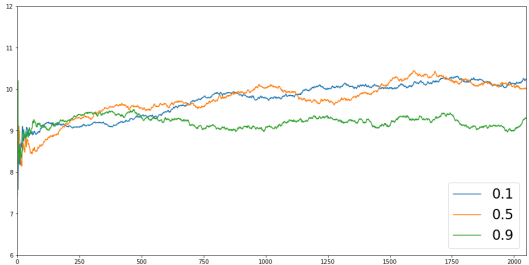
- m = a machine/lever
- N_m = how many times it has been queried in the past
- $r_{i,m}$ = the reward from the i^{th} time machine m was queried

- Need schedule: $\epsilon = 1$ at start, $\epsilon = 0$ when converged
- Back to guessing weeks (w) required...
- Best schedule for one bandit $\neq$ best schedule for another
  (continues to sample bad machines when still deciding between good machines)

Epsilon-greedy results

- Constant epsilon:

Mean of last 256 queries: (12 is ideal)

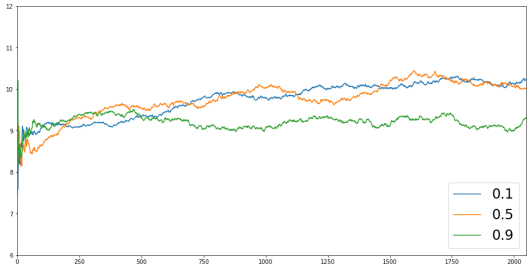


- Higher $\epsilon \rightarrow$ Lower steady state
- Lower $\epsilon \rightarrow$ Slower start

Epsilon-greedy results

- Constant epsilon:

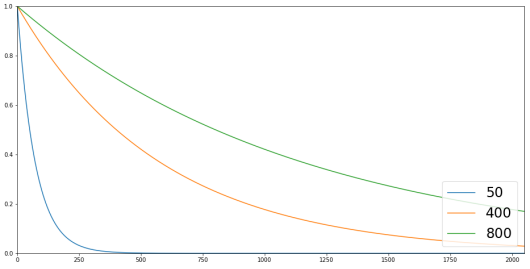
Mean of last 256 queries: (12 is ideal)



- Higher $\epsilon \rightarrow$ Lower steady state
- Lower $\epsilon \rightarrow$ Slower start

- Exponential epsilon, $\epsilon = \frac{1}{2}^{\frac{t}{\lambda}}$
  where t = query number; λ = half life

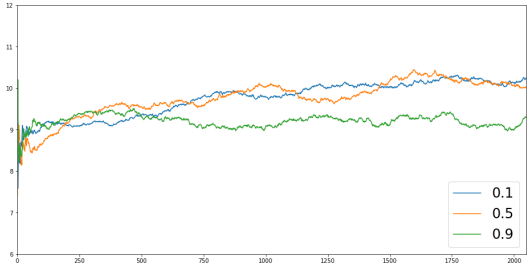
Schedule:



Epsilon-greedy results

- Constant epsilon:

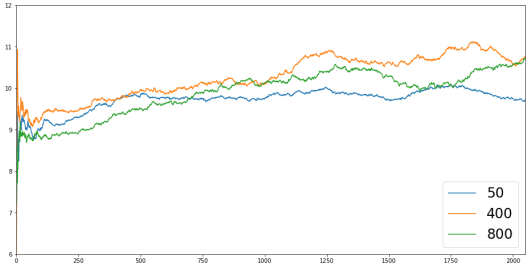
Mean of last 256 queries: (12 is ideal)



- Higher $\epsilon \rightarrow$ Lower steady state
- Lower $\epsilon \rightarrow$ Slower start

- Exponential epsilon, $\epsilon = \frac{1}{2}^{\frac{t}{\lambda}}$
  where t = query number; λ = half life

Mean of last 256 queries: (12 is ideal)



- Too low: Doesn't explore enough
- Too high: Wastes exploitation time

UNIVERSITY OF BATH

Upper confidence bound

- Satisfies all goals
- "Optimistic" – pulls lever with highest upper confidence bound
  - Confidence warranted: Keeps going
  - Confidence misplaced: Gets evidence to reduce confidence

UNIVERSITY OF BATH

Upper confidence bound

- Satisfies all goals
- "Optimistic" – pulls lever with highest upper confidence bound
  - Confidence warranted: Keeps going
  - Confidence misplaced: Gets evidence to reduce confidence
- Upper confidence bound?

$$= \underset{m \in \mathbf{M}}{\operatorname{argmax}} \left(\frac{1}{N_m} \sum_{i=1}^{N_m} r_{i,m} + U(N_m, t) \right)$$

  - Expected value + confidence bound
  - Confidence bound
    - High with few samples
    - Low with lots of samples

Hoeffdings Inequality

- For any random variable **bounded** by $0 < X < 1$:

$$P\left(\mathbb{E}[X] > \frac{1}{N}\sum_{i=1}^{N} x_i + \mu\right) \leq e^{-2N\mu^2}$$

Hoeffdings Inequality

- For any random variable **bounded** by $0 < X < 1$:

$$P\left(\mathbb{E}[X] > \frac{1}{N}\sum_{i=1}^{N} x_i + \mu\right) \leq e^{-2N\mu^2}$$

- Rearrange

$$U(N) = \sqrt{\frac{-\log(p)}{2N}}$$

where $p = e^{-2N\mu^2}$ = some small probability bound

Hoeffdings Inequality

- For any random variable **bounded** by $0 < X < 1$:

$$P\left(\mathbb{E}[X] > \frac{1}{N}\sum_{i=1}^{N} x_i + \mu\right) \leq e^{-2N\mu^2}$$

- Rearrange

$$U(N) = \sqrt{\frac{-\log(p)}{2N}}$$

where $p = e^{-2N\mu^2}$ = some small probability bound

- What to set p to?
- UCB1: $p = t^{-4}$ (an adaptive schedule – always works)

$$U(N, t) = \sqrt{\frac{2\log(t)}{N}}$$

Hoeffdings Inequality

- For any random variable **bounded** by $0 < X < 1$:

$$P\left(\mathbb{E}[X] > \frac{1}{N}\sum_{i=1}^{N} x_i + \mu\right) \leq e^{-2N\mu^2}$$

- Rearrange

$$U(N) = \sqrt{\frac{-\log(p)}{2N}}$$

where $p = e^{-2N\mu^2}$ = some small probability bound

- What to set p to?
- UCB1: $p = t^{-4}$ (an adaptive schedule – always works)

$$U(N, t) = \sqrt{\frac{2\log(t)}{N}}$$

- Bounded? We know rewards – hence bounds. Scale accordingly!
(at least for all scenarios suggested)

UCB1

- For each query $t = 1, 2, \ldots$ select machine

$$= \underset{m \in \mathbf{M}}{\operatorname{argmax}} \left(\frac{1}{N_{t,m}} \sum_{i=1}^{N_{t,m}} r_{i,m} + u \sqrt{\frac{2 \log(t)}{N_{t,m}}} \right)$$

where

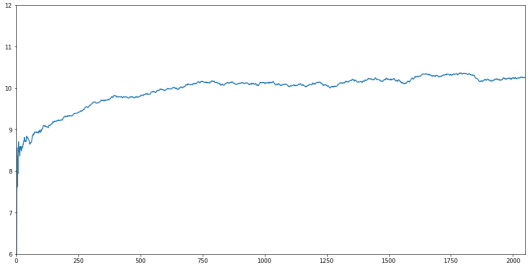
  - $N_{t,m}$ = number of times machine m has been selected at time t
  - $r_{i,m}$ = reward from the i^{th} query of machine m
  - u is the upper bound on the reward (assuming 0 is lower bound)

- Note: Start by evaluating each machine once!

UCB1 results

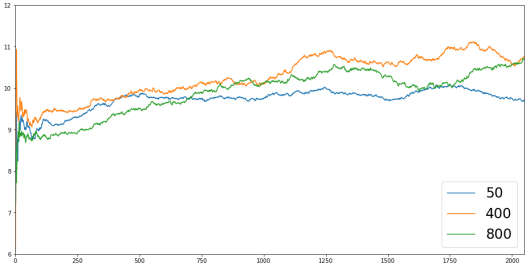
- UCB1:

Mean of last 256 queries: (12 is ideal)



- Epsilon greedy with exponential schedule:

Mean of last 256 queries: (12 is ideal)



UCB1 summary

- Epsilon greedy can win if well tuned
- Tuning not viable: UCB1

- Solves original problems:
  - Many changes at a time
  - Can add new variants at any time
  - Dynamically adapts w weeks

UCB1 summary

- Epsilon greedy can win if well tuned
- Tuning not viable: UCB1

- Solves original problems:
  - Many changes at a time
  - Can add new variants at any time
  - Dynamically adapts w weeks

- Now:
  - Incremental mean (+ others for completeness)
  - Further improvements

UNIVERSITY OF BATH

Incremental mean

- Mean can be calculated incrementally:

```
count = 0
mean = 0.0
for value in data:
  count += 1
  mean += (value - mean) / count
```

Incremental variance

- Can also do variance: (includes mean)
  (for completeness)

```
count = 0
mean = 0.0
scatter = 0.0 # variance * count
for value in data:
  count += 1
  delta = value - mean
  mean += delta / count
  scatter += delta * (value - mean)
variance = scatter / count
```

(Note how scatter uses the delta before then after the mean update!)

Incremental covariance

- Covariance as well: (generalisation of variance)
  (for completeness)

```
count = 0
mean_x = 0.0
mean_y = 0.0
coscatter = 0.0
for x, y in data:
  count += 1
  delta_x = x - mean_x
  mean_x += delta_x / count
  mean_y += (y - mean_y) / count
  coscatter += delta_x * (y - mean_y)
covar = coscatter / count
```

(One delta has to be from before mean update, other after!)

Incremental median

- Median is a little different: (converges in the limit)
  (for completeness)

```
delta = 0.01
median = 0.0
for value in data:
  if value<median:
    median -= delta
  else:
    median += delta
```

Delta: Smaller = Slower convergence but more accurate estimate.
Can adjust delta dynamically for better performance; a histogram may be a better choice.

Incremental percentile

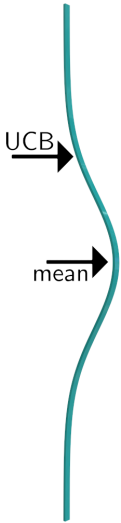
- Median generalises to percentile: (converges in the limit; p is the desired quantile) (for completeness)

```
delta = 0.01
quantile = 0.0
for value in data:
  if value<quantile:
    median -= delta * (2 - 2*p)
  else:
    median += delta * (2*p)
```

- Min/max obvious!

Bayesian upper confidence bound

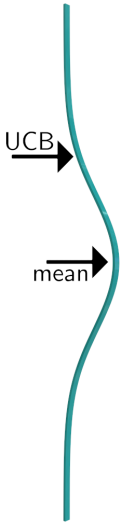
- Select model with
  best value within credible interval
  (Credible interval = Bayesian confidence interval)
- Typically 95% credible interval



Bayesian upper confidence bound

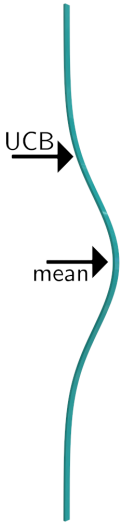
- Select model with
  best value within credible interval
  (Credible interval = Bayesian confidence interval)
- Typically 95% credible interval
- Gaussian distribution:

$$x \sim \mathcal{N}(\mu, \sigma^2)$$



Bayesian upper confidence bound

- Select model with
  best value within credible interval
  (Credible interval = Bayesian confidence interval)
- Typically 95% credible interval



- Gaussian distribution:

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

- Conjugate prior: (3 parameters)

$$\sigma^{-2} \sim \Gamma\left(\frac{n_0}{2}, \frac{1}{2V_0}\right)$$

$$\mu|\sigma^2 \sim \mathcal{N}\left(\mu_0, \frac{\sigma^2}{n_0}\right)$$

Bayesian upper confidence bound

- Select model with
  best value within credible interval
  (Credible interval = Bayesian confidence interval)
- Typically 95% credible interval

- Gaussian distribution:

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

- Conjugate prior: (3 parameters)

$$\sigma^{-2} \sim \Gamma\left(\frac{n_0}{2}, \frac{1}{2V_0}\right)$$

$$\mu|\sigma^2 \sim \mathcal{N}\left(\mu_0, \frac{\sigma^2}{n_0}\right)$$

- Update when x_t arrives:

$$n_t = n_{t-1} + 1$$

$$\mu_t = \frac{n_{t-1}\mu_{t-1} + x_t}{n_{t-1} + 1}$$

$$\sigma_t^2 = \sigma_{t-1}^2 + \frac{n_{t-1}(x_t - \mu_{t-1})^2}{n_{t-1} + 1}$$

Bayesian upper confidence bound

- Select model with
  best value within credible interval
  (Credible interval = Bayesian confidence interval)
- Typically 95% credible interval

- Gaussian distribution:

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

- Conjugate prior: (3 parameters)

$$\sigma^{-2} \sim \Gamma\left(\frac{n_0}{2}, \frac{1}{2V_0}\right)$$

$$\mu|\sigma^2 \sim \mathcal{N}\left(\mu_0, \frac{\sigma^2}{n_0}\right)$$

- Update when x_t arrives:

$$n_t = n_{t-1} + 1$$

$$\mu_t = \frac{n_{t-1}\mu_{t-1} + x_t}{n_{t-1} + 1}$$

$$\sigma_t^2 = \sigma_{t-1}^2 + \frac{n_{t-1}(x_t - \mu_{t-1})^2}{n_{t-1} + 1}$$

- Upper confidence bound is hence

$$\mu_t + A(n_t)\sqrt{\frac{n_t + 1}{n_t^2}}\sigma_t$$

where $A(n_t) = \mathcal{T}(0.975, n_t)$,
$\mathcal{T}$ = CDF of Student's t
(`scipy.stats.t.cdf(0.975, nt)`)

UNIVERSITY OF BATH

Thompson sampling

- Better Bayesian approach! (invented 1933)

- Goal: $P(\text{choose machine}) = P(\text{machine is best})$
- Procedure:
  1. Draw reward from posterior for each machine
  2. Select machine with maximum reward

UNIVERSITY OF BATH

Thompson sampling

- Better Bayesian approach! (invented 1933)

- Goal: $P(\text{choose machine}) = P(\text{machine is best})$
- Procedure:
  1. Draw reward from posterior for each machine
  2. Select machine with maximum reward

- Simple problems: UCB1 usually wins
- But: Can have far more complex models!

UNIVERSITY OF BATH

Contextual bandits

- Reward of pulling lever $= f(\text{side information})$
- Know side information before making choice
- e.g. advert system: side information $=$ age, gender etc.

UNIVERSITY OF BATH

Contextual bandits

- Reward of pulling lever $= f(\text{side information})$
- Know side information before making choice
- e.g. advert system: side information $=$ age, gender etc.

- Expected reward $\rightarrow$ ML model!
- Must be robust to noisy input,
  e.g. linear regression with stochastic gradient descent,
  density estimation

UNIVERSITY OF BATH

Contextual bandits

- Reward of pulling lever $= f(\text{side information})$
- Know side information before making choice
- e.g. advert system: side information = age, gender etc.

- Expected reward $\rightarrow$ ML model!
- Must be robust to noisy input,
  e.g. linear regression with stochastic gradient descent,
  density estimation

- Uses of contextual bandits:
  - Adverts on websites
  - YouTube "next video"
  - Cambridge Analytica (propaganda targetting)

- A double edged sword...

UNIVERSITY OF BATH

Summary

- Multi-armed bandits
- Contextual bandits

- Epsilon greedy
- (Bayesian) Upper confidence bound
- Thompson sampling

- There are many other variants!
  (primarily for different scenarios)

UNIVERSITY OF BATH

Further reading

- Gives UCB variants and their performance: (proof heavy)
  "*Finite-time Analysis of the Multiarmed Bandit Problem*",
  by Auer, Cesa-Bianchi & Fisher (2002)

- Original Thompson sampling paper: (old fashioned)
  "*On the likelihood that one unknown probability exceeds another in view of the evidence of two samples*",
  by Thompson (1933)

- Investigation of Google advertising biases:
  "*Automated Experiments on Ad Privacy Settings*
  A Tale of Opacity, Choice, and Discrimination",
  by Datta, Carl & Datta (2015)

Sources

- One armed bandit postcard, public domain: https://commons.wikimedia.org/wiki/File:One-Armed_Bandits_at_Stockmen%27s_Hotel,_Elko,_Nevada_(83581).jpg