Reinforcement Learning 1

Basic concepts, Bandit algorithm

Pekka Marttinen

Aalto University

Reinforcement learning study material

- Excellent online materials
  - Book: *Reinforcement learning: An introduction* by Richard S. Sutton and Andrew G. Barto
  - Online course: *Introduction to Reinforcement Learning* by David Silver

  Both available at: https://deepmind.com/learning-resources/-introduction-reinforcement-learning-david-silver
- Lectures 9 and 10 on this course are based on these materials[1].

[1] with permission

What is reinforcement learning (RL)

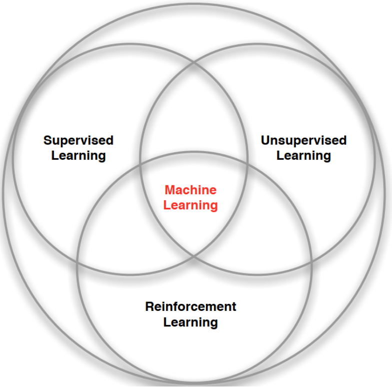
- In RL an agent learns to act in an environment so as to maximize its cumulative future reward.
- The agent is not told which actions to take, but instead needs to discover the best actions by trying them.
- Two important characteristics of RL:
    1. Trial-and-error search for good actions.
    2. Delayed reward.

Examples

- Learn to fly stunt manoeuvres in a helicopter.
- Defeat the world champion at Backgammon.
- Manage an investment portfolio.
- Control a power station.
- Make a humanoid robot walk.
- Play many different atari games better than humans.

What makes reinforcement learning different

- These is no supervisor, only a reward signal
- Feedback is delayed, not instantaneous
- Time matters (sequential, non i.i.d. data)
- Agent's actions affect subsequent data



Rewards

- A reward R_t is a scalar feedback signal
- Indicates how well agent is doing at step t.
- The agent's goal: *select actions that maximize total future reward*.
- Rewards may de delayed:
  - A financial investment (may take months to mature)
  - Refuelling a helicopter (might prevent a crash in several hours)
  - Blocking opponent moves (improves winning chances later in the game)

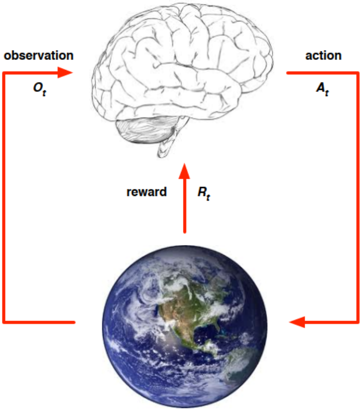
Examples of rewards

- Fly stunt manoeuvres in a helicopter
    - +ve reward for following desired trajectory
    - -ve reward for crashing
- Defeat the world champion at Backgammon
    - +ve/-ve reward for winning/losing a game.
- Manage an investment portfolio
    - +ve reward for each $ in bank
- Control a power station
    - +ve reward for producing power
    - -ve reward for exceeding safety thresholds
- Make a humanoid robot walk
    - +ve reward for forward motion
    - -ve reward fo falling over
- Play many different Atari games better than humans
    - +ve/-ve reward for increasing/decreasing score.

Learning and planning

- These is no supervisor, only a reward signal
- Feedback is delayed, not instantaneous

Agent and environment



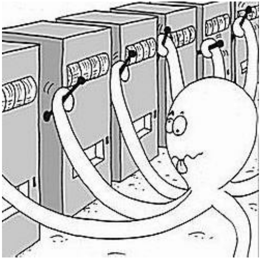
- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

Exploration vs. Exploitation

- Online decision making involves a fundamental choice:
  - *Exploitation:* Select the best option given current information
  - *Exploration:* Gather more information, and maybe find a better choice
- Examples:
  - Selecting a dish in a restaurant
    - Select your favourite dish
    - Try something new
  - Oil drilling
    - Drill at the best known location
    - Drill at a new location
  - Online advertising
    - Show the most successful advert
    - Show a different advert

The Multi-Armed Bandit (1/2)

- A multi-armed bandit[a] consists of:
  - A set $\mathcal{A}$ of m actions ("arms")
  - $\mathcal{R}^a(r) = \mathbb{P}[r|a]$ is an unknown distribution of rewards
- At step t the agent
  - selects action $a_t \in \mathcal{A}$
  - gets reward $r_t \sim \mathcal{R}^{a_t}$
- **Goal**: maximize the *cumulative reward* $\sum_{t=1}^{T} r_t$.



[a] Image: Microsoft Research

The Multi-Armed Bandit (2/2)

- A simplified example of *sequential decision making* under uncertainty
  - Only a single state, the reward doesn't depend on t.
- The value of an action a is its expected value:

$$q_*(a) = \mathbb{E}[R_t | A_t = a].$$

- The optimal action maximizes Q:

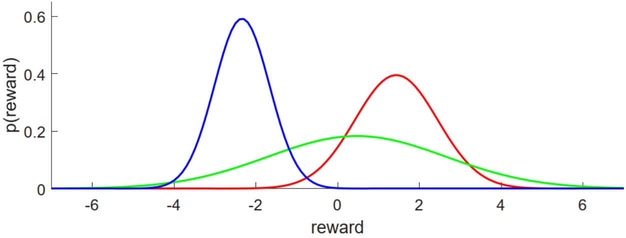
$$a^* = \arg\max_{a \in \mathcal{A}} q_*(a).$$

- At step t, $Q_t(a)$ is the estimate of q_*:

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i \mathbf{1}_{A_i = a}}{\sum_{i=1}^{t-1} \mathbf{1}_{A_i = a}}$$

Example

- Example: 3 actions (arms), initialized with 10 random picks:
  - Arm 1: 0.02, 2.41, 1.61, 1.69
  - Arm 2: 2.01, -1.07
  - Arm 3: -1.66, -2.90, -1.87, -2.94
- **Q**: Which arm would you select next?

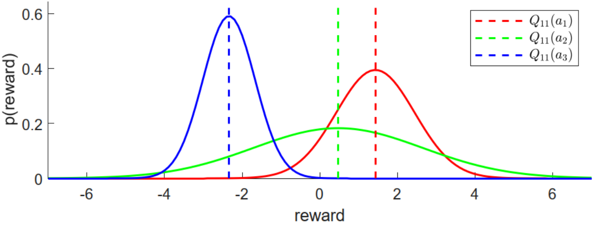


Greedy action selection

- Select the action that looks best, i.e., *greedily*:

$$A_t = \arg\max_a Q_t(a).$$

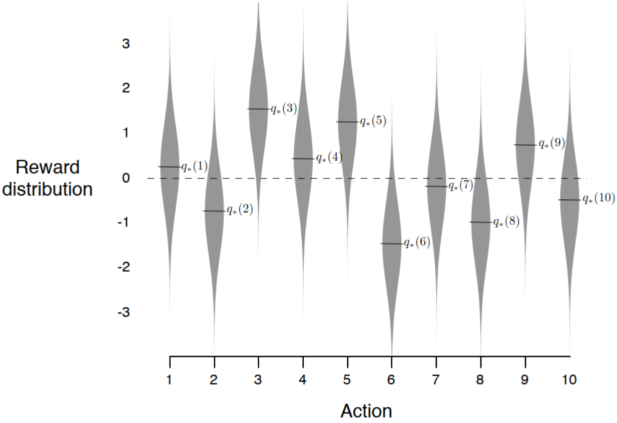
- Exploitation only.



Epsilon-greedy action selection

- ϵ-greedy method, with $\epsilon \in [0, 1]$:
    - With probability ϵ pick the action uniformly at random
    - With probability $1 - \epsilon$, pick the action greedily
- A compromise between exploitation and exploration
    - $\epsilon = 0$, greedy method, only exploitation
    - $\epsilon = 1$, completely random, only exploration
- **Q**: In case of two actions and $\epsilon = 0.5$, what is the probability of selecting the greedy action?

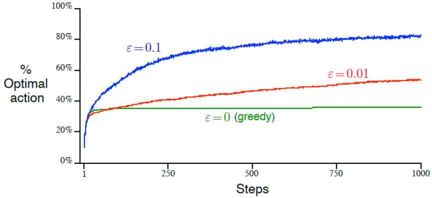
Example: the 10-armed testbed[2]

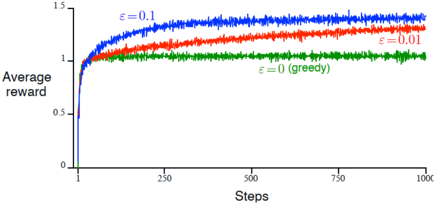


[2] Sutton & Barto, Fig. 2.1.

Example: the 10-armed testbed

- Average results across 2000 bandit problems.[a]
- Greedy approach gets stuck to sub-optimal solutions
- $\epsilon = 0.1$ finds the best action fastest. However, eventually $\epsilon = 0.01$ will yield a highest average reward. (Why?)





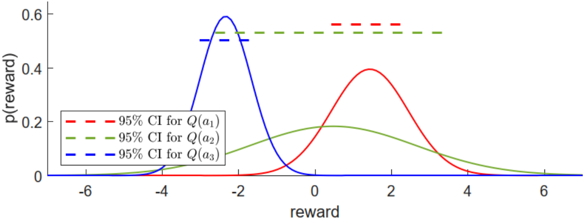
[a] Sutton & Barto, Fig. 2.2.

Upper confidence bound (1/3)

- ϵ-greedy samples either greedily or completely randomly.
  - It would be better to focus exploration on actions that seem somehow promising.
- The 95% confidence interval for $q_*(a)$ is approximately

$$\left[Q_t(a) - 2\frac{\sigma_{t,a}}{\sqrt{N(a)}}, Q(a) + 2\frac{\sigma_{t,a}}{\sqrt{N(a)}} \right],$$

where $Q_t(a)$ is the average reward so far, $N_t(a)$ is the number of times a has been picked, and $\sigma_{t,a}$ is the estimated standard deviation for reward.



Upper confidence bound (2/3)

- Estimate an upper confidence $U_t(a)$ for each action
  - such that $q_*(a) \leq Q_t(a) + U_t(a)$ with a large probability.
- Select action maximizing Upper Confidence Bound (UCB)

$$a_t = \arg\max_{a \in \mathcal{A}} Q_t(a) + U_t(a).$$

- Action is likely to be picked if
  1. its current estimated reward $Q_t(a)$ is large (exploitation)
  2. its uncertainty U_t is high (exploration)

Upper confidence bound (3/3)

- Bayesian UCB:

$$a_t = \arg\max_{a \in \mathcal{A}} Q_t(a) + c\sigma_{t,a} / \sqrt{N_t(a)}.$$

  For example $c = 2$ gives the upper bound of the 95% CI.

- General UCB[3]:

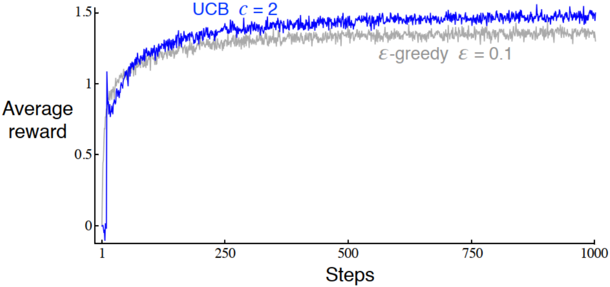
$$a_t = \arg\max_{a \in \mathcal{A}} \left[Q_t(a) + c\sqrt{\frac{\log t}{N_t(a)}} \right]$$

  - $N_t(a)$ increases every time a is picked → uncertainty decreases and the action is selected less often in the future
  - $\log t$ increases at each step → all actions will be selected at some point

[3] For a derivation using Hoeffding's inequality, see David Silver's Lecture 9.

UCB vs. epsilon-greedy

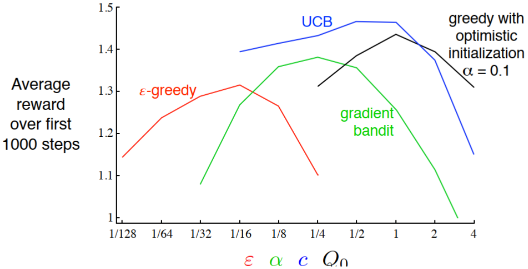
- Average performance on the 10-armed testbed[4].
- In the very beginning UCB is worse than epsilon-greedy as it explores more, but soon catches up after the best action is found.



[4] Sutton & Barto, Fig. 2.4.

Impact of hyperparameters

- Each method has a hyperparameter that affects its performance[5].
- A good method
  - has high average reward
  - is relatively insensitive to the hyperparameter value.



[5] Sutton & Barto, Fig. 2.6.

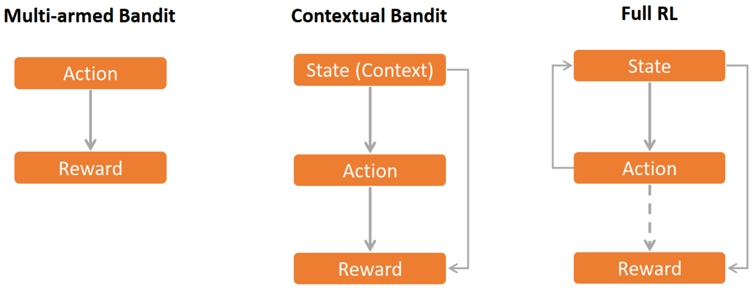
Contextual bandit definition

- A contextual bandit consists of:
  - A set $\mathcal{A}$ of m actions ("arms")
  - $\mathcal{S} = \mathbb{P}[s]$, an unknown distribution over states (or "contexts")
  - $\mathcal{R}_s^a(r) = \mathbb{P}[r|s, a]$, an unknown distribution over rewards
- At each step t
  - Environment generates state $s_t \sim \mathcal{S}$
  - Agent selects action $a_t \in \mathcal{A}$
  - Environment generates reward $r_t \sim \mathcal{R}_{s_t}^{a_t}$
- **Goal**: maximize the *cumulative reward* $\sum_{t=1}^{T} r_t$.

Examples

- Problem: play multiple different bandits (i.e., slot machines), and the machine you confront is selected at random
    - State: properties of the slot machine (color, size,...)
- Problem: recommed a movie to the user
    - State: user profile (user's age, previously liked movies,...)
- Problem: select a treatment for a patient
    - State: characteristics of the patient (age, gender, treatment history,...)

Contextual bandit vs. full RL



- Contextual bandit is simplified from the full RL
  - In both, the action can depend on the current state.
  - But in contextual bandits the action doesn't affect next state and the reward is always immediate.

Contextual bandit estimation

- The expected reward for state s and action a

$$q_*(s, a) = \mathbb{E}[r|s, a].$$

- Estimating this after t steps is a supervised problem
  - Data: $(s_1, a_1, r_1), (s_2, a_2, r_2), \dots, (s_t, a_t, r_t)$.
  - Estimate value function with a linear function approximator:

$$q_*(s, a) \approx Q_t(s, a) = \phi(s, a)^\top \theta_t.$$

- Collect feature vectors and rewards until t:

$$\Phi_t = \begin{bmatrix} \phi(s_1, a_1)^\top \\ \phi(s_2, a_2)^\top \\ \vdots \\ \phi(s_t, a_t)^\top \end{bmatrix}, \quad \mathbf{r}_t = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_t \end{bmatrix} \tag{1}$$

Contextual bandit estimation

- Estimate parameter θ_t by least-squares:

$$\begin{aligned} \theta_t &= \left(\Phi_t^\top \Phi_t\right)^{-1} \Phi_t^\top \mathbf{r}_t && (2) \\ \mathrm{Var}\left[\theta_t\right] &= \gamma^2 \left(\Phi_t^\top \Phi_t\right)^{-1}, && (3) \end{aligned}$$

- γ^2 is the error variance, i.e, $r = \phi(s,a)^\top \theta + \epsilon$, with $\epsilon \sim N(0, \gamma^2)$, whose estimator is

$$\overline{\gamma}_t = \sqrt{\frac{1}{n_{df}} \sum_{\tau=1}^{t} (r_\tau - \phi(s_\tau, a_\tau)^\top \theta_t)^2} \quad \text{(RMSE)}, \qquad (4)$$

where $n_{\text{df}} = t - \dim(\phi(s,a))$ is the degrees of freedom.

UCB criterion for contextual bandit

- Variance of the action value estimator:

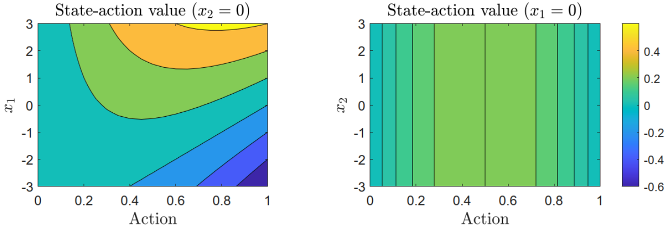
$$\begin{aligned}\sigma_t^2 &= \mathrm{Var}\left[\phi(s,a)^T\theta_t\right] \\ &= \phi(s,a)^T\mathrm{Var}\left[\theta_t\right]\phi(s,a) \\ &= \gamma^2\phi(s,a)^T\left(\Phi_t^\top\Phi_t\right)^{-1}\phi(s,a). \qquad (5)\end{aligned}$$

- The UCB criterion can be used for action selection

$$\begin{aligned}a_{t+1} &= \arg\max_{a\in\mathcal{A}} Q_t(s,a) + c\sigma_t \\ &= \arg\max_{a\in\mathcal{A}} Q_t(s,a) + c\sqrt{\phi(s_t,a)^\top\mathrm{Var}\left[\theta_t\right]\phi(s_t,a)}.\end{aligned}$$

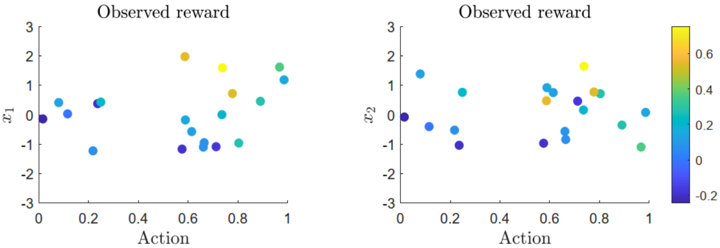
Toy example: setup

- Context $\mathbf{x} = (x_1, x_2)^\top \sim N_2(0, I)$
- A continuous-valued action $a \in [0, 1]$.
- Reward $r = -1.4a^2 + 1.3a + 0.3ax_1 + \epsilon$, $\epsilon \sim N(0, 0.15^2)$.
- For example: $\mathbf{x}$ can be user features, a the proportion of certain type of content on the web-page, and r whether the user clicked a link or not.



Toy example: initial observations

- Initial data are triplets (state, action, reward):
  - state $\mathbf{x} = (x_1, x_2)^\top$ generated by environment
  - action a selected by an initial policy, $a \sim \text{Unif}(0, 1)$
  - reward $r(\mathbf{x}, a)$ generated by the environment
- Data after 20 iterations:



Toy example: state-action value approximation

- Linear state value approximator:

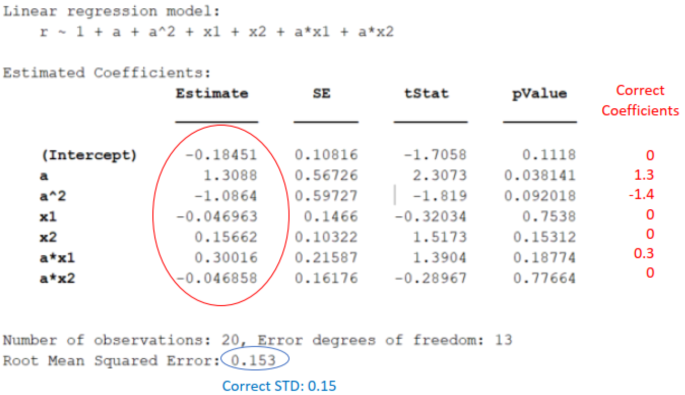
$$\begin{aligned} r_i &= \theta_0 + \theta_1 a + \theta_2 a^2 + \theta_3 x_1 + \theta_4 x_2 + \theta_5 a x_1 + \theta_6 a x_2 \\ &= \theta^\top \phi(\mathbf{x}, a), \end{aligned}$$

- The feature vector:

$$\phi(\mathbf{x}, a) = (1, a, a^2, x_1, x_2, ax_1, ax_2)^\top.$$

Toy example: estimation

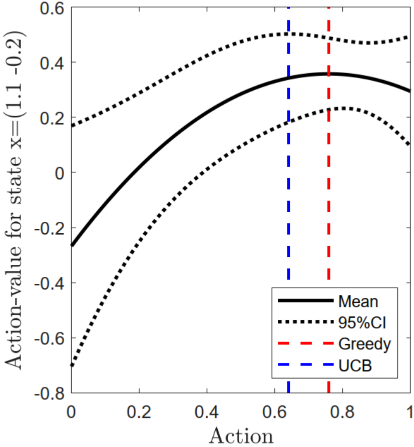
- Collect data into Φ_{20}, $\mathbf{r}_{20}$ (Equation 1).
- Estimate the linear model (Equations 2,3,4 or some statistics package):



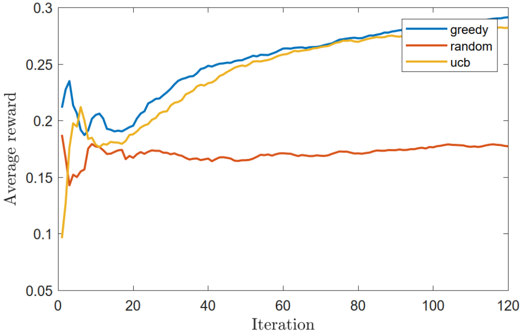
```
Linear regression model:
    r ~ 1 + a + a^2 + x1 + x2 + a*x1 + a*x2

Estimated Coefficients:
                    Estimate        SE         tStat       pValue        Correct
                                                                       Coefficients
                   _________    _______    ________    ________

    (Intercept)     -0.18451    0.10816     -1.7058      0.1118             0
    a                 1.3088    0.56726      2.3073    0.038141           1.3
    a^2              -1.0864    0.59727      -1.819    0.092018          -1.4
    x1             -0.046963     0.1466    -0.32034      0.7538             0
    x2               0.15662    0.10322      1.5173     0.15312             0
    a*x1             0.30016    0.21587      1.3904     0.18774           0.3
    a*x2           -0.046858    0.16176    -0.28967     0.77664             0

Number of observations: 20, Error degrees of freedom: 13
Root Mean Squared Error: 0.153
                         Correct STD: 0.15
```

Toy example: UCB for action selection

- Assume the next state generated by the environment is $\mathbf{x} = (x_1, x_2)^\top = (1.1, -0.2)^\top$.
- Calculate uncertainty σ_t^2 of different actions $a \in [0, 1]$ using Equation (5).
- The figure shows $Q_t(s, a) \pm c\sigma_t$ for $c = 2$ corresponding to the 95% confidence interval.

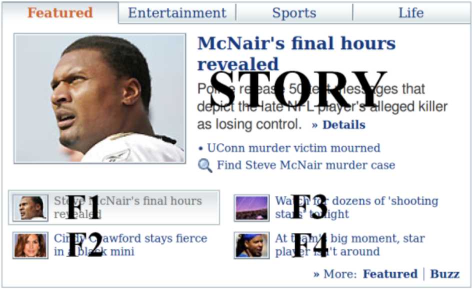


Toy example: methods' comparison



- In this toy problem the greedy method works well
  - For a given state, the action value is unimodal.
  - The linear model approximates to the true reward well.

Example: Personalized news article recommendation[6]



McNair's final hours revealed

Police release 50 text messages that depict the late NFL player's alleged killer as losing control. » Details

- UConn murder victim mourned

Find Steve McNair murder case

Steve McNair's final hours revealed

Cindy Crawford stays fierce in a black mini

Watch for dozens of 'shooting stars' tonight

At team's big moment, star player isn't around

» More: Featured | Buzz

- Task: Select a news story to feature on the *Yahoo! Front Page Today* webpage (one of the most visited pages at the time).

[6] Li, et al., (2010) A Contextual-Bandit Approach to Personalized News Article Recommendation. WWW 2010.

Example: Personalized news article recommendation

- A contextual bandit where the actions (arms) correspond to different news stories.
- Reward: $+1$ if the user clicks the article, 0 otherwise.
- The context vector $\mathbf{x}_{t,a}$ summarizes both features of the current user u_t and actions a (news items), for all $a \in \mathcal{A}_t$.
- LinUCB:
  - Assume: $\mathbb{E}[r_t | \mathbf{x}_{t,a}] = \mathbf{x}_{t,a}^\top \theta_a^*$.
  - Select: $a_t = \arg\max_{a \in \mathcal{A}_t} \left(\mathbf{x}_{t,a}^\top \widehat{\theta}_a + \alpha \sqrt{\mathbf{x}_{t,a}^\top A_a^{-1} \mathbf{x}_{t,a}} \right)$.

Example: Personalized news article recommendation

- Using contextual bandits improved the *click-through rate* (CTR) by 12.5% compared to the context-free bandit algorith.
- UCB outperformed ϵ-greedy approaches.

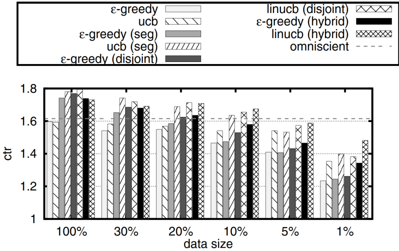
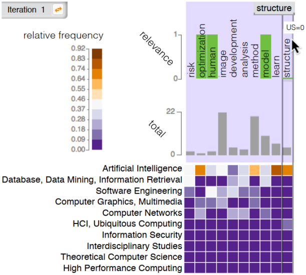


Fig. 4a in Li et al.

- Nevertheless, bandit algorithms are still greedy
  - Don't maximize the number of clicks in repeated visits.
  - Extensions have been developed since, e.g, based on the Markov-decision process (discussed later).

Research at Aalto: Expert-in-the-loop AI

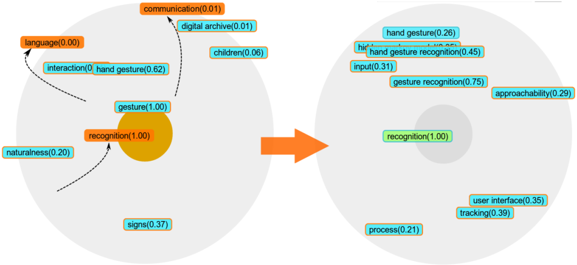
- Goal: predict citations based on article's keywords and abstract.
- Improve the prediction model by asking feedback from an expert about relevant words.
- The UCB criterion was used to select the word for which feedback was asked.



Micallef et al. (2017). Interactive Elicitation of Knowledge on Feature Relevance Improves Predictions in Small Data Sets. IUI'17.

Research at Aalto: Exploratory search of documents

- The UI shows keywords that are likely relevant for the user
- The user gives feedback about the relevance of the keywords by interacting with the UI.
- The keywords to show are selected with the UCB criterion.
- Another window shows the documents corresponding to the keywords.



Glowacka et al. (2013). Directing exploratory search: reinforcement learning from user interactions with keywords.