The Problem

Set-up: K ‘arms’, with different rewards. The reward from each arm follows an unknown probability distribution. Initially, we don’t know which arm is best.

Our aim: Find an algorithm minimising the ‘cumulative regret’ after n runs,

\[ R_n = \bar{\mu}^* - \mathbb{E}\left[ \sum_{t=1}^n X_t \right] = \sum_{k \in [K]} \Delta_k \mathbb{E}[T_k(n)]. \]

- \( \bar{\mu}^* \) - mean of the optimum arm
- \( X_t \) - reward at time t
- \( [K] = 1, 2, ... K \) - set of arm indices
- \( \Delta_k \) - difference in mean of arm k and mean of the optimum arm
- \( T_k(n) \) - number of times arm k has been played after n runs

This sort of problem arises in e.g. targeted advertising, clinical trials.

To find an algorithm achieving ‘sub-linear regret’, we need to test different arms.

Exploration: To better estimate the expected reward of each arm, we need to test different arms.

Exploitation: We should select arms that give us a higher reward more often.

The Exploration/Exploitation Dilemma

Investigating Optimism in the Exploration/Exploitation Dilemma

Matthew Gorton\textsuperscript{1}  Supervisor: Alan Wise\textsuperscript{2}

\textsuperscript{1}Physics, Lancaster University  \textsuperscript{2}Maths & Stats, Lancaster University

Our aim: Find an algorithm minimising the ‘cumulative regret’ after n runs.

E.g. Choosing a cafe

- Choose a cafe you know well (A), or a new cafe (B) you haven’t tried before?
- Optimism principle → try the new cafe several times, and update your information

Bayesian Approach: Thompson Sampling

\textbf{Approach:}
- Assume a prior probability distribution for each arm (e.g. Beta(\( \alpha_k \), \( \beta_k \))
- Sample a value \( \theta_k \) from the prior for each arm, \( \theta_k \sim \text{Beta}(\alpha_k, \beta_k) \)
- Select arm with maximum \( \theta_k \)
- Update \( \alpha_k, \beta_k \)

\textbf{Advantages}
- Optimal performance (for Bernoulli bandits)
- Not sensitive to parameters in prior (tuning not required)

\textbf{Disadvantages}
- Only works well if prior pdf is conjugate to the true pdf

Upper Confidence Bound (UCB) algorithms

Select arm at time t, \( A_t \), maximising the ‘upper confidence bound’, i.e.

\[ A_t \in \arg \max_k [\hat{\mu}_k(t-1) + f(T_k(t-1))]. \]

where \( f(T_k(t-1)) \) is a decreasing function of \( T_k(t-1) \)

- \( \hat{\mu}_k(t-1) \): encourages exploitation
- \( f(T_k(t-1)) \): encourages exploration

Can achieve theoretical asymptotic lower regret bound [Lai and Robbins, 1985], using the KL-UCB method.

Lower confidence bound

\[ \bar{\mu}^* - \mathbb{E}\left[ \sum_{t=1}^n X_t \right] = \sum_{k \in [K]} \Delta_k \mathbb{E}[T_k(n)]. \]

\textbf{Advantages}
- Can achieve theoretical asymptotic lower regret bound
- Only works well if prior pdf is conjugate to the true pdf

Disadvantages
- Not sensitive to parameters in prior (tuning not required)

LinUCB algorithm

Uses information collected from observations to update a ‘feature vector’, guiding arm selection.

LinUCB generally performs better than a simple UCB algorithm. Plot by Alan Wise.

Contextual Bandits

Sometimes, we can use information about one arm to make predictions about other arms, improving algorithmic performance. e.g. someone who buys red jumpers is likely to buy blue jumpers as well.

- Arms are now normalised vectors.
- The extent to which arms ‘point’ in the same direction shows their similarity.

References


Future Work

- Investigate whether different methods perform better for particular bandit probability distributions.
- Fix LinUCB and look into extensions.
- Consider more complicated examples, e.g. bandits in fraud detection.