CS6046: Multi-Armed Bandits
Final Project

The goal of the project is to use multi-armed bandit (MAB) algorithms to come up with effective batting strategies in the game of cricket. We now describe the rules and game dynamics. There are 6 actions namely $\mathcal{A} = \{0, 1, 2, 3, 4, 6\}$. The current ball is denoted by t, and the MAB algorithm will have to play for $t = 1, \ldots, T$ balls, and at ball t needs to pick an action $a_t \in \mathcal{A}$. At each ball the following happen:

1. A wicket can fall with a probability $p_{\text{out}}(a_t)$.

2. In case there is no wicket, then a run is scored with probability $p_{\text{run}}(a_t)$.

In **Problems 1, 2, 3, 4** there is only one batter, and every time the batter gets out, the same batter plays again.

• **Problem 1:** Let $w(a) = p_{\text{out}}(a)$ be the expected wicket per ball for playing action a. Let $w_* = \arg\min_{a \in \mathcal{A}} w(a)$. The goal is to minimise the regret $R_n = \sum_{t=1}^{n}(w(a_t) - w_*)$. Hint: KL-UCB might be useful.

• **Problem 2:** Let $s(a) = (1 - p_{\text{out}}(a)) \times p_{\text{run}}(a) \times a$ be the expected run per ball for playing action a. Let $s_* = \arg\max_{a \in \mathcal{A}} s(a)$. The goal is to minimise the regret $R_n = \sum_{t=1}^{n}(s_* - s(a_t))$.

• **Problem 3:** Let $\rho(a) = \frac{(1-p_{\text{out}}(a)) \times p_{\text{run}}(a) \times a}{p_{\text{out}}(a)}$ be the runs gained per wicket for action a. Let $\rho_* = \arg\max_{a \in \mathcal{A}} \rho(a)$. The goal is to minimise the regret $R_n = \sum_{t=1}^{n}(\rho_* - \rho(a_t))$. Hint:

• **Problem 4:** In this problem, there will be a total of $m = 1, \ldots, M$ matches. Each match will consist of $t = 1, \ldots, 60$ balls, and there are 4 wickets, i.e., if 4 wickets fall before 60 balls then the match stops. Let s_m be the score in match m. MAB algorithm is supposed to maximise $\sum_{m=1}^{M} s_m$.

• **Problem 5:** This is same as **Problem 4**, with a slight change. There are 4 kinds of batters, $i = 1, \ldots, 4$. This means we have $p_{\text{out}}(a, i)$ and $p_{\text{out}}(a, i)$ to be functions of the batter i. Once a batter gets out, the MAB needs to decide which batter will bat next, and for the current batter the MAB has to choose a_t. Here also the MAB algorithm is supposed to maximise $\sum_{m=1}^{M} s_m$.