SoC: EconML

Presentation

Yash Virani
210050170

DESCRIPTION

Compared Stochastic, Adverserial and Contextual bandits and discussed favourable scenarios for each.

Multi-armed bandit problem

- The multi-armed bandit problem is a classic framework in machine learning and decision theory that models a sequential decision-making scenario.
- In the multi-armed bandit problem, an agent faces a set of available actions or arms, and each arm is associated with an unknown reward distribution.
- The agent's goal is to maximize its cumulative reward over a series of time steps by strategically selecting which arms to pull.

Stochastic Bandits

- The term 'stochastic' here refers to the assumption that rewards associated with each arm are drawn from **fixed probability distributions.**
- The agent does not have prior knowledge of these distributions and must learn them through exploration and exploitation.
- At each time step, the agent selects an arm to pull from the set of available arms. The goal is to maximize the cumulative reward obtained over time.

Stochastic Bandits

- **Exploration vs Exploitation:** The agent faces a trade-off between exploring arms to learn more about their reward distributions and exploiting arms that seem to provide higher rewards based on the current knowledge.
- A variety of learning algorithms may be applied to estimate the reward probabilities associated with each arm based on observed outcomes. Some commonly used are **epsilon-greedy, Thompson sampling and Upper Confidence Bound**.

Stochastic Bandits

- One common approach of exploration is epsilon-greedy, where the agent chooses the arm with the highest estimated reward with probability (1-epsilon) and selects a random arm (explore) with probability epsilon.

- UCB balances exploration and exploitation by selecting arms based on an upper confidence bound that takes into account the uncertainty of estimated rewards.

Stochastic Bandits

- Thompson Sampling employs a probabilistic approach to solve the stochastic bandit problem. The algorithm maintains a distribution (usually a Bayesian distribution) over the unknown reward probabilities of each bandit. It starts with some prior belief about the reward distribution and updates it based on the observed outcomes.
- Regret is a key metric used to evaluate the performance of stochastic bandit algorithms. It measures the difference between the cumulative reward the agent achieves and the maximum cumulative reward that could have been obtained by always selecting the best arm. A s**ublinear regret** is a must for an efficient algorithm.

Adverserial Bandits

- They are a variation of the multi-armed bandit problem where rewards are determined by an adversary instead of being generated probabilistically.
- Adversarial bandits often assume a non-stationary environment where the adversary can change the reward generation strategy over time. This adds an additional challenge for the agent as it needs to adapt its decision-making strategy to the changing reward dynamics.

Adverserial Bandits - EXP3

- The algorithm most commonly used to tackle adverserial bandits is **EXP3** **(**Exponential-Weight algorithm for Exploration and Exploitation).
- At the beginning, the algorithm initializes a probability distribution over the available actions (bandits). Each action is assigned an initial probability weight. Usually, all the weights are initialized equally to ensure exploration in the early rounds.
- At each round, the algorithm samples an action to play based on the probability distribution. The selection is done by using the probabilities as weights in a weighted random selection process.

Adverserial Bandits - EXP3

- After selecting an action, the agent receives a reward from the adversary. The agent does not know the rewards of the unselected actions, only the observed reward for the chosen action.

- Based on the observed reward, the algorithm updates the probability distribution. The goal is to assign higher probabilities to actions that have yielded higher rewards in the past, while also promoting exploration.

Adverserial Bandits - EXP3

- Probability update formula: The probability update in EXP3 is done using an exponential-weight update scheme. For each action, the probability weight is updated as follows:
- w_t(a) = w_{t-1}(a) * exp(η * r_t(a) / p_t(a))
- The regret is calculated as the difference between the cumulative reward obtained by the EXP3 algorithm and the cumulative reward obtained by an optimal strategy.

Contextual Bandits

- They are a variation of the multi-armed bandit problem where the agent receives additional information, known as context, alongside the bandit arms. In contextual bandits, the rewards associated with each action depend not only on the action taken but also on the provided context.
- The agent receives a context vector x at each round. The context vector captures additional information or features that describe the current state of the environment.

Contextual Bandits

- For each context-action pair (x, a), there is an associated reward distribution or function that provides the expected reward when action 'a' is taken given context 'x'. The agent's objective is to learn the best action to take for each context based on the available information.
- Some algorithms used in case of contextual bandits are Follow the Leader Algorithm, Hedge Algorithm and epoch-greedy algorithm

Contextual Bandits - FTL

- In case of Follow the leader (as is evident from the name) the agent selects action with highest expected reward based on observed historical rewards associated with each action and the current context. Then the parameters are updated following a gradient descent approach (in direction that maximizes expected reward.

Contextual Bandits - Hedge

- Hedge algorithm maintains a probability distribution over a set of policies or actions and updates the distribution based on observed rewards. This allows the algorithm to explore different actions while assigning higher weights to actions that have yielded higher rewards in the past. The Hedge algorithm aims to minimize regret (it achieves sublinear regret).

Contextual Bandits - Epoch Greedy

- It initializes action value estimates and selects actions greedily based on these estimates. After a fixed number of rounds (epochs), it switches to exploration mode where actions are chosen uniformly at random. The algorithm updates action value estimates based on observed rewards using a simple average update rule. The Epoch Greedy algorithm achieves a trade-off between exploring different actions and exploiting actions with higher estimated values.

CHOICE...

1. **Stochastic Bandits:** It is suitable when the environment is relatively stable, and the reward distributions do not change over time. Stochastic bandits are often used in scenarios where exploration is necessary to gather information about the reward probabilities of different arms.

2. **Adverserial Bandits:** Adversarial bandits are commonly used in scenarios where the environment is non-stationary, and the rewards can change dynamically based on the agent's actions.

CHOICE...

3. **Contextual Bandits:** It is suitable when there is additional contextual information available that can influence the rewards. Contextual bandits are used in scenarios where the agent needs to adapt its decision-making based on the current state of the environment or other relevant factors.

In practice, it is also possible to encounter problem domains that combine elements from multiple variations. For example, there may be contextual bandits with adversarial elements or stochastic bandits with contextual information. In such cases, we can adapt and design algorithms that suit the specific requirements of your problem.

THANKS FOR READING :)