Unit-3

Lab experiments

Implement an e-commerce platform that adjusts product prices dynamically based on demand and competition using the TRPO algorithm to optimize pricing strategies.

Aim

To implement an e-commerce dynamic pricing system that adjusts product prices based on customer demand and competitor pricing, using the TRPO reinforcement learning algorithm to learn optimal pricing strategies that maximize revenue while remaining competitive.

Algorithm:

- Initialize the pricing environment and policy network
- Observe the current state (demand, competitor price, current price)
- Select a pricing action using the current policy
- Apply the price change and observe reward (profit)
- Store trajectory data (state, action, reward)
- Estimate policy gradients using collected trajectories
- Update the policy using TRPO with KL-divergence constraint
- Repeat until pricing policy converges
- Output optimal pricing strategy

Code Github Link:

https://github.com/syekumar/MLA0316-Reinforcement-learning-

Output:

```
======= RESTART: C:/Users/Sye Kumar/OneDrive/Documents/unit 03 lab 04.py ======
Training Dynamic Pricing using TRPO (Simplified)

Episode 1: Demand=1.0, Price=100, Reward=95.0
Episode 2: Demand=0.8, Price=100, Reward=76.0
Episode 3: Demand=0.6, Price=100, Reward=57.0
Episode 4: Demand=0.6, Price=105, Reward=57.0
Episode 5: Demand=1.0, Price=105, Reward=95.0
Episode 6: Demand=1.2, Price=105, Reward=114.0
Episode 7: Demand=0.6, Price=105, Reward=57.0
Episode 8: Demand=1.0, Price=100, Reward=95.0
Episode 9: Demand=1.0, Price=95, Reward=95.0
Episode 10: Demand=1.2, Price=95, Reward=114.0

Optimized Pricing Policy Probabilities: [0.33333321 0.33333357 0.33333321]
```

Result:

The TRPO-based pricing agent learns to adjust prices dynamically by considering market demand and competitor prices. Over training episodes, the policy converges toward price adjustments that maximize expected profit, while maintaining stability through constrained policy updates.