Trip Purchases with Double Q-learning

Matt DiNardo - Data Scientist Interview

Introduction

For this problem, an agent faces a sequential decision-making process in deciding whether to buy a ticket for a trip today or wait for a lower price later. The goal is to learn an optimal policy for the task. One approach proposed by Groves & Gini (2011) could be to create a supervised learning system that uses features of the current state to predict the lowest future price of the trip, recommend `BUY` if the current best price is within some acceptable range of the forecasted minimum price, and recommend `WAIT` otherwise.[1] However, a supervised learning approach like this neglects the sequential structure of the problem. Sutton (1988) describes how temporal-difference learning methods make more efficient use of their experience, converge faster, and produce better policies for problems with sequential structures.[2]

Methodology

The TD/reinforcement learning method used in this experiment is a variation of Q-learning called Double Q-learning proposed by Silver et al. (2015).[3] Its main differences from normal Q-learning are the replacement of a tabular Q function with a function approximator, its use of experience replay to de-correlate samples during training, and use of a double estimator.

Training a DQ-Network on <*state, action, reward*> representations aggregated from the `boscun` data set results in learning a policy with an average out-of-sample cumulative reward of about $73 per trip.

Rewards

The agent faces a Markov Decision Process: given state s, select action $a \in \{BUY, WAIT\}$ that maximizes the expected sum of *discounted* future rewards. Here, the reward is constructed as the daily return on `WAIT`, calculated as $minprice_t - minprice_{t+1}$. If `BUY` is selected, the episode terminates with reward 0.

There are important considerations that can be addressed by this reinforcement learning approach:

- *discount factor*: Future rewards (savings) should be discounted based on risk and time-value preference. The level of the discount rate may significantly alter the behavior of the policy learned by the agent, so the discount rate should be somewhat representative of the preferences of users/customers.

- *transitions*: In the real world, giving a `BUY` recommendation is not a guarantee the user will follow the policy. These concerns with stochastic transitions and rewards are automatically handled by Q-learning.

Results

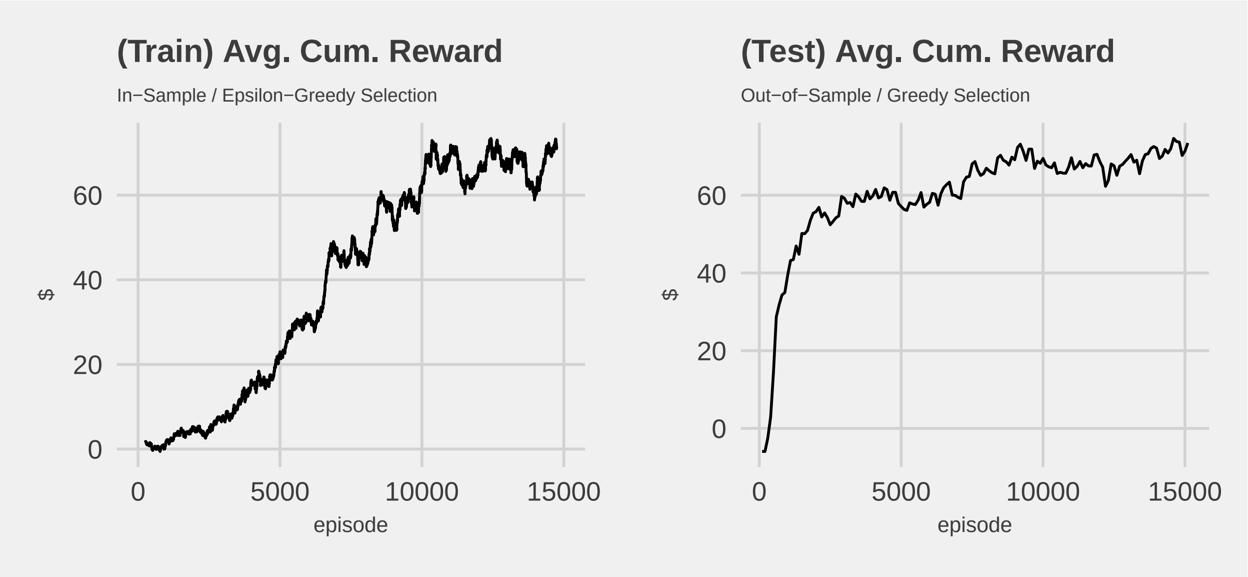
The DQN was trained using tensorflow with 864 unique trips, where each trip was a sequence of daily statistics describing itinerary prices for a trip on a given day. These features included *minimum price*, *mean price*, *days out*, *seats*, etc. and represented the *state* of the environment. The agent performance was monitored across 15,000 training episodes.

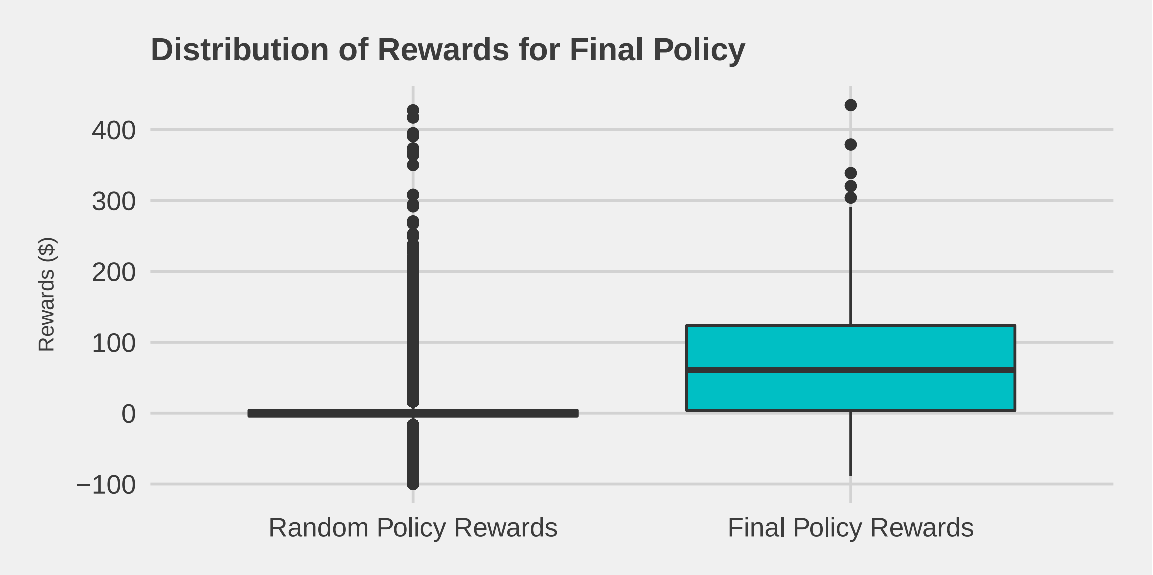
[1] [1] Gini, Groves. A regression model for predicting optimal purchase timing for airline tickets. 2011

[2] [2] Sutton. Learning to Predict by the Methods of Temporal Differences. 1988

[3] [3] van Hasselt, Guez, Silver. Deep Reinforcement Learning with Double Q-learning. 2015

Training & Test Performance





Conclusion and Remaining Issues

The agent successfully learns to time trip purchase with an average savings of about $73, more than enough to buy some extra jumbo margaritas in Cancun. Reinforcement Learning provides promising methods for solving this problem, but potential issues remain:

- Testing generalization across different origins/destinations
- Non-stationarity of rewards / Responsiveness to changes in market conditions
- Training stability of function approximator (neural network) in online setting
- **Improving state representation with better features & aggregation methods**