Reinforcement Learning for Flight Ticket Prices

Niki Agrawal, Ramnik Arora
nikhar@stanford.edu, rarora@stanford.edu

Models

Baseline: Customer always selects “buy.” Reflects a common trend for flight prices to increase close to purchase point, avoids ticket selling out.

Q-Learning:
- Tabular representation of (state, action) → Q-value developed during training
- Each state belongs to an equivalence class: same flight_uid (carrier + flight number), same hours before departure, but different departure dates
- Single iteration over training data in each equivalence class
- Reward Function:
  - For action = wait:
    - If flight sells out, Reward = -300,000
    - If flight does not sell out, Reward = 0
  - For action = buy: Reward = -current sale price

DQN
- State → neural network → Q-value
- 2 hidden layers: sigmoid activation followed by ReLU, Output layer: linear
- State = [current sale price, hours before departure, booking_code_count]
- Reward Function: If flight sells out, Reward = -previous flight price - $300

Results

Baseline:
- Customer always selects “buy.” Reflects a common trend for flight prices to increase close to purchase point, avoids ticket selling out.

Q-Learning:
- Tabular representation of (state, action) → Q-value developed during training
- Each state belongs to an equivalence class: same flight_uid (carrier + flight number), same hours before departure, but different departure dates
- Single iteration over training data in each equivalence class
- Reward Function:
  - For action = wait:
    - If flight sells out, Reward = -300,000
    - If flight does not sell out, Reward = 0
  - For action = buy: Reward = -current sale price

DQN
- State → neural network → Q-value
- 2 hidden layers: sigmoid activation followed by ReLU, Output layer: linear
- State = [current sale price, hours before departure, booking_code_count]
- Reward Function: If flight sells out, Reward = -previous flight price - $300

Conclusions & Future Work

- Access to high quality flight data for multiple routes and over long time-periods is incredibly important to be able to train the data. It'll be interesting to extend the work to more routes, with more data.
- Deep Q-Networks can capture more nuanced states, but they are difficult to train and needs more parameter tuning.
- Users generally purchase a ticket on a route, not an individual flight. The current setup doesn’t capture the majority use case.
- Currently, the agents interaction doesn’t change the environment. However, in actual ticket purchase problems, the agents behavior can lead to tickets being sold out etc.

References

1. To Buy or Not to Buy: Mining Airfare Data to Minimize Ticket Purchase Price
   https://www.stat.berkeley.edu/~aldous/157/Papers/etzioni.pdf
2. On Optimizing Airline Ticket Purchase Timing
   https://dl.acm.org/citation.cfm?id=2733384
3. Computational Complexity of Air Travel Planning