The end of Airline Revenue Management as we know it?

(Deep) Reinforcement Learning for Revenue Management

AGIFORS Symp 2017, London
October 2017
Speakers: Rodrigo ACUNA AGOST and Thomas FIIG
Credits

Amadeus Airlines R&D + Innovation and Research

Nicolas BONDOUX
Quan NGUYEN
Thomas FIIG
Rodrigo ACUNA-AGOST

RM Expertise + AI Expertise = (Deep) Reinforcement Learning for Revenue Management
Motivation – limitations of RMS

RMS assumptions

- RMS assumes that the future is accurately described by the past:
  - Issue with change in business environment (new competitors)
  - Issue with shift in demand and willingness to pay
  - Issue with change in customer behavior (for example: arrival pattern)

- RMS assumes that customers are rational:
  - However, customers are irrational, influenced by psychological factors (framing, etc.).
  - There is no model for irrationality.

- RMS assumes monopoly:
  - Competitors offers are accounted for implicitly by how they affect customers behavior. This corresponds to a monopoly seen from RMS

- RMS assumes that a model exist that describes “world”. For general offers this is impossible:
  - Increased complexity of offers (seat + ancillaries)
  - Complex products (flexibility, time to think, etc.), bundles of ancillaries; are difficult to price.
  - Interactions between the prices of ancillaries, bundles, fare families, etc.
**Scientific worldview**

**Reinforcement Learning**
- Balancing exploitation and exploration
- Ability to model customers' behavior?
  - No

**Classical Revenue Management**
- Collect observations
- Experimentation
- Low dimensions (degrees of freedom)
  - Build forecasting model
    - All future customers
    - Optimization - Multiple sequential decisions
      - Dynamic Programming (Bellman 1950s)
    - Complete solution
  - Proof in 1990s

**Supervised Machine Learning**
- Build forecasting model
  - One given customer
- High dimensions (degrees of freedom)
- Classification
- Prediction
Application of RL
How it works?

(1) Actions
(2) State
(3) Reward
Actions: \{ Left, no-change, Right \}

State:  \{ Information of Sensors \}

Reward = stay alive as long as possible  
(Alive = no crash)
So, what’s new?
... it is very disruptive

- No demand forecast
- No modeling passenger behavior (WTP)
- No RM optimization model
Reinforcement Learning

Mathematical details*)

\[ V(t, x) = \text{Max}_f \left[ (1 - P(f))V(t + 1, x) + P(f)\left( f + V(t + 1, x - 1) \right) \right] \]

\[ V(s) = \text{Max}_a \sum_{s'} p_{ss'}^a [R_{ss'}^a + V(s')] \]

\[ Q(s, a) = \sum_{s'} p_{ss'}^a [R_{ss'}^a + V(s')] \]

\[ V(s) = \text{Max}_a Q(s, a) \]

*) Reinforcement Learning, Sutton, Brato, 1998

Bellman (1950s)

Q-learning Watkins (1989)
Experimentation our research journey ...

1. Base Reinforcement Learning
   Scenario: Monopoly

2. + Deep Learning

3. + GPUs
   + Scenario: with 1 competitor

High Complexity / Realism

Low Complexity / Realism
Experiments

Base Reinforcement Learning in a Monopoly

**Simulation set-up**
Cap = 10
DCP = 20
Fare classes = 3
Fenceless fare structure

**RMS basecase**
- AL1: Dynamic Programming
- Two customer segments with different frat5
- Forecaster = Q-forecasting

**Reinforcement Learning**
- No Forecaster or Optimizer
- AL1: Q–learning
- State (t,x)
- Action: f1,f2,f3, closed
Experiments

1

Base Reinforcement Learning in a Monopoly

- Theoretical optimum => perfect demand forecast, perfect WTP models, no changes in the market

- RL can solve the problem.
- RL converges to the optimal solution
- Poor performance (RL needs a lot of data and computation.)
Experiments

Base Reinforcement Learning in a Monopoly

Lack of observations (cabin full far from departure)

Lack of observations (cabin empty at departure)

Good policies where we have lots of observations

RL Policies vs. Optimal Policies
Deep Reinforcement Learning

Deep Neural Network

Classical Artificial Neural Networks

Input

Output

Accuracy: worse than other ML methods

Deep Neural Network

Input

Layer1

Layer2

Layer3

Output

Accuracy: better than other ML methods
Deep Reinforcement Learning

Deep Neural Network as function approximation

What is function approximation?

\[(s, a) \rightarrow Q(s, a)\]

\[\theta(x, y) = \alpha x + \beta y\]

\[(s, a) \rightarrow Q(s, a, \theta) = \alpha s + \beta a\]
Experiments + Deep Learning

Normal RL

135kY

of flight data

~ 97%

of opt. rev

Deep RL

5-15Y

of flight data

~ 99%

of opt. rev

Revenue

Data (1000 year)

Revenue

Data (Year)
Experiments

Reinforcement Learning in Duopoly

**Simulation set-up**
Cap = 50
DCP = 20
Fare classes = 10
Fenceless fare structure

**RMS basecase**
- AL1: Dynamic Programming
- A2: AT80
- Two customer segments with different frat5
- Estimated frat5 (optimal revenue)
- Forecaster = Q-forecasting

**Reinforcement Learning**
- No Forecaster or Optimizer
- AL1: Deep RL
- State (t,x)
- Action: f1, f2, f3, ..., f10, closed
Experiments 3
One Competitor + GRUs

RMS vs Competitor

Revenue

100%
RMS

89%
AT80

DRL vs Competitor

Revenue

102.5%
DRL

80%
AT80
Why is RL better?

- Remember RMS were optimal
- DRL produces higher revenue by understanding the competitive game and swamping the competitors with low yield passengers.
Conclusion

- Classical RMS techniques are no longer sufficient.
- RL opens the door to a radical new approach:
  - Model free
  - No forecasting
  - No optimization
  - Learns by direct price testing
- Shown that RL = RMS for monopoly
- We discover the richness of RL
- Beats RMS against competition

- More complex scenarios:
  - Full Networks
  - Many competitors
  - Pricing of complex product
  - Pricing of psychological factors - irrational customers.
  - Shift in demands/WTP
- Improve learning performance
- Add more information to the state (eg., competitors and market)