Mining Airfare Data to Minimize Ticket Purchase Price

Oren Etzioni (UW)  Craig Knoblock (USC)
Alex Yates (UW)  Rattapoom Tuchinda (USC)
Price change over time for American Airlines flight #192:223, LAX-BOS, departing on Jan. 2.
Consumers’ Dilemma

To Buy or Not to Buy... that is the question..

Hamlet

Data mining $\rightarrow$ Price drops
1. Consumer wants to buy a ticket.
2. Hamlet: ‘buy’ (this is a good price).
3. Or: ‘wait’ (a better price will emerge).
Arbitrage Model

1. “going price” is $900.
2. Hamlet anticipates a price of $400.
3. Hamlet offers a $600 fare.
4. Hamlet buys when the price drops to $400.
5. Consumer saves $300; Hamlet earns $200.
   (of course, Hamlet could lose money!)
Will Flights sell out?

1. Watch the number of empty seats.
2. Upgrade to business class.
3. Place on another flight and give a free ticket.

In our experiment: upgrades were sufficient.
Is Airfare Prediction Possible???

Complex “yield management” algorithms.
- airlines have tons of historical data.
Exogenous events create randomness.

How about the stock market?
True markets are unpredictable.
For Hamlet, prices are set by the airlines!
Surprising Experimental Result

Savings: buy immediately versus Hamlet.
Optimal: buy at the best possible time.

HAMLET’s savings were 61.8% of optimal!

Though it be madness, yet there be method in it.
Data Set

- Used Fetch.com’s data collection infrastructure.
- Collected over 12,000 price observations:
  - Lowest available fare for a one-week roundtrip.
  - LAX-BOS and SEA-IAD.
  - 6 airlines including American, United, etc.
  - 21 days before each flight, every 3 hours.
Learning Task Formulation

**Input:** price observation data.

**Algorithm:** label observations (decision point); run learner.

**Output:** Classify each decision point → buy versus wait.
Formulation Fine Points

- Want to learn from the latest data.
- Run learner nightly to produce a new model.
  - Learner is trained on data gathered to date.
- Learned policy is a sequence of 21 models.

- Test set: 8 * 21 decision points for the last 1/3 of the flights.
Labeling Training Data

IF price drops between now and takeoff THEN label(O)=wait
ELSE label(O) \rightarrow Pr(price will drop between now and takeoff)

We estimate Pr based on behavior of past flights.
**Candidate Approaches**

- **Fixed:** “asap”, 14 days prior, 7 days,…
- **By hand:** an expert looks at the data.
- **Time series:** \( P_t = F(P_{t-1}, P_{t-2}, ... P_1) \).
  - Not effective at price jumps!
- **Reinforcement learning:** Q-learning.
  - Used in computational finance.
- **Rule learning:** Ripper, …
• Features include price, airline, route, hours-before-takeoff, etc.

• Learned 20-30 rules…

IF hours-before-takeoff ≥ 252 AND price ≥ 2223
AND route = LAX-BOS THEN wait.
Simple Time Series

- Predict price using a fixed window of \( k \) price observations weighted by \( \alpha \).
- We used a linearly increasing function for \( \alpha \)

\[
P_{t+1} = \frac{\sum_{i=1}^{k} \alpha(i) p_{t-k+i}}{\sum_{i=1}^{k} \alpha(i)}
\]
Q-learning

Natural fit to problem

\[ Q(a, s) = R(a, s) + \gamma \cdot \max_{a'} (Q(a', s')) \]

\[ Q(b, s) = -\text{price}(s) \]

\[ Q(w, s) = \begin{cases} -300000 & \text{if flight sells out after } s. \\ \max(Q(b, s'), Q(w, s')) & \text{otherwise.} \end{cases} \]
Stacking with three base learners:

1. Ripper (e.g., $R=\text{wait}$)
2. Time series
3. Q-learning (e.g., $Q=\text{buy}$)

Ripper used as the meta-level learner.

Output: classifies each decision point as ‘buy’ or ‘wait’.
Experimental Results

- Real price data; Simulated passengers.
  - Uniform distribution over decision points. (sensitivity)
  - Requesting specific flights (also 3hr interval).
- Learner run once per day on “past data”.
- Execution: label each purchase point until buy (or sell out).
- Compute savings (or loss).
• **Net** savings = cost now – cost at purchase point.
• Penalty for sell out = upgrade cost. 0.42% of the time.
• Total ticket cost is $4,579,600.
Passenger requests any nonstop flight in a 3 hour interval:

Sensitivity Analysis

Legend:
- Time Series
- Q-Learning
- By Hand
- Ripper
- Hamlet
- Optimal
## Upgrade Penalty

<table>
<thead>
<tr>
<th>Method</th>
<th>Upgrade Cost</th>
<th>% Upgrades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>$0</td>
<td>0%</td>
</tr>
<tr>
<td>By hand</td>
<td>$22,472</td>
<td>0.36%</td>
</tr>
<tr>
<td>Ripper</td>
<td>$33,340</td>
<td>0.45%</td>
</tr>
<tr>
<td>Time Series</td>
<td>$693,105</td>
<td>33.00%</td>
</tr>
<tr>
<td>Q-learning</td>
<td>$29,444</td>
<td>0.49%</td>
</tr>
<tr>
<td><strong>Hamlet</strong></td>
<td><strong>$38,743</strong></td>
<td><strong>0.42%</strong></td>
</tr>
</tbody>
</table>
Discussion

- 76% of the time --- no savings possible.
- Uniform distribution over 21 days.
- 33% of the passengers arrived in the last week.
- No passengers arrived >21 days before.

Simulation understates possible savings!
## Savings on “Feasible” Flights

<table>
<thead>
<tr>
<th>Method</th>
<th>Net Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>30.6%</td>
</tr>
<tr>
<td>By hand</td>
<td>21.8%</td>
</tr>
<tr>
<td>Ripper</td>
<td>20.1%</td>
</tr>
<tr>
<td>Time Series</td>
<td>25.8%</td>
</tr>
<tr>
<td>Q-learning</td>
<td>21.8%</td>
</tr>
<tr>
<td><strong>Hamlet</strong></td>
<td><strong>23.8%</strong></td>
</tr>
</tbody>
</table>

Comparison of Net Savings (as a percent of total ticket price) on Feasible Flights
Related Work

- Trading agent competition.
  - Auction strategies
- Temporal data mining.
- Time Series.
- Computational finance.
Future Work

- More tests: international, multi-leg, hotels, etc.
- Cost sensitive learning (tried MetaCost).
- Additional base learners
- Bagging/boosting
- Refined predictions
- Commercialization: patent, license.
Conclusions

1. Dynamic pricing is prevalent.
2. Price mining a-la-Hamlet is feasible.
3. Price drops can be surprisingly predictable.
4. Need additional studies and algorithms.
5. Great potential to help consumers!

All’s well that ends well.
Savings by Method

- Savings over “buy now”.
- Penalty for sell out = upgrade cost.
- Total ticket cost is $4,579,600.

<table>
<thead>
<tr>
<th>Method</th>
<th>Savings</th>
<th>Losses</th>
<th>Upgrade Cost</th>
<th>% Upgrades</th>
<th>Net Savings</th>
<th>% Savings</th>
<th>% of Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>$320,572</td>
<td>$0</td>
<td>$0</td>
<td>0%</td>
<td>$320,572</td>
<td>7.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>By hand</td>
<td>$228,318</td>
<td>$35,329</td>
<td>$22,472</td>
<td>0.36%</td>
<td>$170,517</td>
<td>3.8%</td>
<td>53.2%</td>
</tr>
<tr>
<td>Ripper</td>
<td>$211,031</td>
<td>$4,689</td>
<td>$33,340</td>
<td>0.45%</td>
<td>$173,002</td>
<td>3.8%</td>
<td>54.0%</td>
</tr>
<tr>
<td>Time Series</td>
<td>$269,879</td>
<td>$6,138</td>
<td>$693,105</td>
<td>33.00%</td>
<td>-$429,364</td>
<td>-9.5%</td>
<td>-134.0%</td>
</tr>
<tr>
<td>Q-learning</td>
<td>$228,663</td>
<td>$46,873</td>
<td>$29,444</td>
<td>0.49%</td>
<td>$152,364</td>
<td>3.4%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Hamlet</td>
<td>$244,868</td>
<td>$8,051</td>
<td>$38,743</td>
<td>0.42%</td>
<td>$198,074</td>
<td>4.4%</td>
<td>61.8%</td>
</tr>
</tbody>
</table>
Passenger requests any nonstop flight in a 3 hour interval:

<table>
<thead>
<tr>
<th>Method</th>
<th>Net Savings</th>
<th>% of Optimal</th>
<th>% upgrades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>$323,802</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>By hand</td>
<td>$163,523</td>
<td>55.5%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Ripper</td>
<td>$173,234</td>
<td>53.5%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Time Series</td>
<td>-$262,749</td>
<td>-81.1%</td>
<td>6.3%</td>
</tr>
<tr>
<td>Q-Learning</td>
<td>$149,587</td>
<td>46.2%</td>
<td>0.2%</td>
</tr>
<tr>
<td>Hamlet</td>
<td>$191,647</td>
<td>59.2%</td>
<td>0.1%</td>
</tr>
</tbody>
</table>