A Cost-Effective Mobile Recommender System for Taxi Drivers

Meng Qu
Rutgers Data Mining Group
Goal: to maximize the income of taxi drivers.

Three ways to develop a recommender system:
Motivation

- How to improve the income of taxi drivers?
- A real story of a taxi driver:

<table>
<thead>
<tr>
<th>Cost</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily Company Fee</td>
<td>• 61.3 dollars</td>
</tr>
<tr>
<td>Gas</td>
<td>• 33.87 dollars</td>
</tr>
<tr>
<td>Taxi Fare</td>
<td>• 12 hours per day</td>
</tr>
<tr>
<td></td>
<td>• 5.56 dollars</td>
</tr>
</tbody>
</table>

- 5.56 dollars per day
- 12 hours per day
- Cost
- Revenue

- 33.87 dollars
- 61.3 dollars
Task: The development of a cost-effective mobile recommender system

- Focusing on recommending an entire driving route for taxi drivers which yields maximum profit in the shortest time.

Objectives:

- Min: Average driving time spend on the route.
- Max: Taxi fare by following the route.
- Max: Pick-up probability along the route.
Preliminaries

• Route: A sequence of connected road segments

• Road Segments: A block with two terminal points r.s and r.e

• Road Segments Network $G=\langle V, E \rangle$
Problem Formulation

Given

- Taxi Drivers’ Trajectory Data
- Road Network Data

Find:

- The potential earnings at road segment \( r \).

\[
e(r) = \frac{\sum_{i=1}^{N_r} Fee(i; r)}{N_r} P(r)
\]

Where, \( Fee(i; r) \) is the earnings of the \( ith \) trip in road segment \( r \). \( N_r \) is the total pick up events. \( P(r) \) is the historical pick up probability.
Problem Formulation

Find:

- The potential cost of road segment $r$:

$$c(r) = (1 - P(r)) \cdot (L(r) \cdot Gas + T(r) \cdot CompanyFee)$$

Where

- $L(r)$ is the length of road segment $r$.
- $Gas$ is the average gas cost every mile.
- $T(r)$ is the average driving time in road segment $r$.
- $CompanyFee$ is the cost of renting cab in unit time.
Objective:

- **Calculate the net profit of road segment** \( r \):

  \[
g(r) = e(r) - c(r)
  \]

  where \( e(r) \) is the potential earning and \( c(r) \) is the potential cost of road segment \( r \).

- **Define the net profit for each route** \( R \) **starts at** \( r_1 \)

  \[
  G(R, r_1, M) = g(r_1) + \sum_{i=2}^{M} g(r_i) \prod_{j=1}^{i-1} (1 - P(r_j))
  \]

  where \( M \) represents the number of small road segments in a recommended driving route \( R \).
Fixed Cursing Length $M$

- Average increase rate of net profit

\[ \tau = \frac{<G(R, r_i, M + 1)> - <G(R, r_i, M)>}{<G(R, r_i, M + 1)>} \]

- The net profit increasing rate $< 10\%$ after 5 road segments.
Problem Formulation

- MNP (Maximum Net Profit) Recommendation Problem

Given the current location $L_{cab} \in r$ of a taxi driver, a fixed cruising length $M$, and a set of route candidates $R$, where $\forall R \in R$ satisfying $R$ starts from $r$. The MNP recommendation problem is to recommend a route $R^* \in R$, which has the maximum net profit, i.e.,

$$R^* = \arg \max_{R \in R} \{ G(R, r, M) \}$$
Building Road Network

Input: The street names of San Francisco

Output: The coordinates of each intersection.

Search for the 4 Nearest Coordinates in 4 different directions

Road Network of San Francisco Bay Area: 2250 road segments in total.
Candidate Route

A Graph Theory Approach.

Building an entire road network of San Francisco Bay Area.

Store the road segments’ information at each node.
Rewrite objective function

\[ G(R, r_1, M) = g(r_1) + (1 - P(r_1))G(R - r_1, r_2, M - 1) \]

Recursion Tree

- The recursion tree \( \gamma_{r_1} \) of a road segment \( r_1 \) is a tree, where each node represents a road segment and the root node is \( r_1 \).
- Each node \( r_i \) in the recursion tree has a children node set that equals to \( r_i . next[] \).
Recursive Recommendation

- An illustrating example of recursive recommendation

\[
G(A; 3) = P(A)g(A) + (1 - P(A)) \times \max(G(B; 2), G(C; 2), G(F; 2), G(E; 2))
\]

\[
G(B; 2) = P(B)g(B) + (1 - P(B)) \times \max(G(D; 1), G(I; 1))
\]

\[
G(p; 1) = P(p)g(p), \text{ } p = D, I, H, G, E
\]
Step 1

• Compare the net profit of each leaf and keep the leaf node with greatest profit within each branch.

Step 2

• Move to the previous level. Compare the net profit of those nodes and also the profit from the remaining children nodes. Keep the node with greatest profit within each branch.

Step 3

• Repeat step 2 until reach the root node. The remaining branch represents for the best driving route.

---

Algorithm 3 rMNP(r,K)

Input 1: recursion tree \( \mathcal{T}_r \);  
Input 2: the depth \( M \) of recursion tree;  
Output: the MNP value and route stating from \( r \);

1: \( \text{Depth} = M - K + 1 \);
2: if (\( \text{Depth} == M \)) do
3: \( \text{Profit} = \emptyset \);
4: \( \text{Route} = \emptyset \);
5: for each \((r_i \in \mathcal{T}_r, \text{level}[\text{Depth}])\) do
6: \( \text{Profit}[i] = g(r) \);
7: \( \text{Route}[i] = r_i \);
8: return \((\text{Max(Profit)}, \text{Max(Route)})\);
9: else
10: \( \text{Profit} = \emptyset \);
11: \( \text{Route} = \emptyset \);
12: for each \((r_i \in \mathcal{T}_r, \text{level}[\text{Depth}])\) do
13: \( (\text{Profit}^*, \text{Route}^*) = r\text{MNP}(r_i, K - 1) \);
14: \( \text{New}_r\text{route}[i] = r_i \cup \text{Route}^* \)
15: \( \text{Profit}[i] = g(r_i) + (1 - g(r_i)) \cdot \text{Profit}^* \);
16: return \((\text{Max(Profit)}, \text{Max(Route)})\);
Top-K Route Recommendation

- **Load Balancing Problem**
  - Recommend the same route to too many drivers at the same location and the same time will degrade the recommendation performance

- **Minimal Redundancy Principle**
  - Measuring Direction Correlation
Direction Correlation

- We need to transform each grid into a *direction vector* $g = (p_1, p_2, p_3, \ldots, p_8)$, where $p_i$ is the probability of moving towards direction $i$ within this grid.

Then, $p_i = f_i / \sum_{k=1}^{8} f_k$

- The probability vector for Route A:

\[
g(A) = (p_{11}, p_{12}, \ldots, p_{n7}, p_{n8})
\]

The Correlation between route A and B:

\[
\rho(A, B) = \frac{\text{Cov}(g(A), g(B))}{\sigma(g(A)) \sigma(g(B))}
\]
**Experimental Results**

**Taxi Data Set**

- GPS location traces of approximately 500 taxis collected around 30 days in the San Francisco Bay Area.

- Focus on two time periods for SF data: 1PM-3PM and 5PM-7PM.

- 87,688 pick-up and drop-off activities in total.

- Use taxi fare website to calculate the profits concerning each trip.
Experimental Results

San Francisco Bay Area

Pick up and drop off activities
Dead Segments: the road segments that the drivers can never find next passenger.

- Pick up event number: \( N^*_r = 0 \)
- Passing by event number: \( N^0_r \approx 0 \)
Data Preprocessing

• Total cost per hour per taxi driver: 5.56 dollar.

• If revenue is less than 5.56 dollar per hour, the driver is losing money.

• Define low profit Segments.

• Prune those low profit segments out of our road network.
Data Preprocessing

- Some route segments associated with traffic jam and the average travel time is long.
- If a taxi driver spent one hour on those road segment without a passenger, he is losing 5.56 dollars.
- Prune those heavy traffic road segments out of our road network.
Profit Improvement for Inexperienced Taxi Drivers

- Route recommendation for Inexperienced Drivers
  - Training set: The driving routes of experienced drivers.
  - Testing set: The driving routes of unexperienced drivers.
  - Driver’s Event: roam → pick up → drop off.

- Experimental results:

<table>
<thead>
<tr>
<th>Table 1: Net Profits per Unit Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommender System</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>SD</td>
</tr>
</tbody>
</table>
Figure 9: Net Profit Statistical Histogram. The blue bar represents our optimized routes potential profit and the red bar represents taxi drivers’ habitual route profits ranked below top 30%

Figure 10: Profit Difference. X axis is Net Profit Difference between our optimized routes and taxi drivers’ habitual route ranked below top 30%, Y axis is the number of events
A Case Study

- A comparison of MNP route recommended by our approach and the suggested route by Google map.

Case Study (a)

Case Study (b)
Top-K Recommendations

- Top-K driving routes starting from the same location, where K equals to 4.

Fig (3) The top 4 driving routes starting from the same location

- Each route has different driving directions and the correlations between those driving distances are very small.
Related Work

- **Recommender systems in mobile environments**
  - Cyber-guide: A mobile context-aware tour guide (G. Abowd and C. Atkeson et al., 1997)
  - Map-based interaction with a conversational mobile recommender system (O. Averjanova et al., 2008)

- **Traveling Salesman Problems**
  - The Traveling Salesman Problem: A Computational Study (Applegate et al., 2006)
  - Probabilistic Traveling Salesman Problems (Jaillet et al., 1985)

- **Efficient Driving Route Problems**
  - T-drive: driving directions based on taxi trajectories (J. Yuan et al., 2010)
  - An energy-Efficient Mobile Recommender System (Y. Ge et al., 2010)
Conclusion

- The development of a cost-effective mobile recommender system for taxi drivers
  - With a goal of maximizing the Net Profit
- A graph representation of road networks
- A recursion strategy based on the special form of net profit function for search optimal routes efficiently
Future Work

- Recommender systems in mobile environments
  - Development of real-time cost-effective mobile recommender systems by integrating the real-time traffic information
  - Development of smart mobile recommender systems with other business constraints, such as taxi-sharing service
Questions?

Thank You!