Discover LinkedIn Job Seeker’s Commute Preference

Dianxia Yang (dianxiay@Stanford.edu)

Background & Challenge

LinkedIn rolled out a feature to allow job seekers to provide us their job commute preference, including starting point, preferred way to commute and maximum commute time. And we will be using this data to provide more relevant job recommendations.

All commute preference data are currently organically input by the active members who discovered this feature.

LinkedIn data are rich for many reasons:

- 30 minute drive near home
-商学院 and industry
- North America
- Large dataset

Problem Statement

We believe there are some correlation between member’s job seeking activity and their implicit commute preference. The goal of the project is to find out:

- Member’s Willingness: Whether a member would love to provide commute preference information.
- Potential Commute Duration: What is member’s most possible commute duration if preference provided.

With this prediction, we could potentially: 1) Promote feature with precise target and personalized copy. 2) Leverage implicit inferred data for job recommendation.

Feature Generation

1. Selected member derived data: industry, job activity score directly as feature
2. With member’s standardized lat/long, generate distance vectors for view/search/apply/save/preference location data with following equation.

\[
d = 2r \arcsin \left( \sin \left( \frac{\theta - \theta'}{2} \right) \right),
\]

3. For each generated distance vectors for action X, we produce the following features: X_count, X_avg_distance and X_std.

Data Preprocess

1. For Problem #1, the binary problem, generate negative data label under the same environment for testing.
2. For Problem #2, the multi-class problem, rewrite the original 6 ranges of duration into 3 ranges: [15,30) -> 0, [30,60) -> 1, [60,120) -> 2
3. Deal with missing values for linear models. We removed samples without job view data, and then populate missing values of search/apply/save/preference data as a chain from business logic.

Models for Willingness Problem (Binary)

- Logistic Regression (binary configuration with L2 regularization) to optimize the following loss function:

\[
J(w) = \sum_{i=1}^{N} \left[ -y_i \log \left( \sigma (w^T x_i) \right) - \left( 1 - y_i \right) \log \left( 1 - \sigma (w^T x_i) \right) \right] + \frac{\lambda}{2} \| w \|^2
\]

- Neural Network

- Simple Decision Tree with Min-leaf pruning. We are trying to maximize the reduction of Gini loss in each step:

\[
\text{Gini} = \sum_i \left[ \text{leaf}_{i} \left( \text{leaf}_{i} \right) + \text{leaf}_{i} \left( \text{leaf}_{i} \right) \right] \cdot \left( 1 - \text{leaf}_{i} \left( \text{leaf}_{i} \right) \right)
\]

- XGBoost, which follows this objective function in each step:

\[
\sum_{t=1}^{T} \left( f_t (x_i) + \frac{1}{2} h_t (x_i) (g_t (x_i))^{2} \right) + \text{regularization}
\]

Models for Duration Problem (Multi Class)

Since we are using the same feature set for this problem, we are also leveraging similar models but with different settings.

- Logistic Regression with configuration of softmax with L2 regularization
- Neural network with softmax output layer
- Simple decision tree with tuned similar pruning
- XGBoost with tuned similar settings

Test Result

Willingness Problem Result: Note that we are reporting average precision score because this problem is subject to precision-recall analysis to make product decision.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision Score</th>
<th>Recall Score</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.7171</td>
<td>0.7171</td>
<td>0.7171</td>
</tr>
<tr>
<td>Neutral Network</td>
<td>0.7279</td>
<td>0.7279</td>
<td>0.7279</td>
</tr>
<tr>
<td>Simple Decision Tree</td>
<td>0.7258</td>
<td>0.7258</td>
<td>0.7258</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.7258</td>
<td>0.7258</td>
<td>0.7258</td>
</tr>
</tbody>
</table>

Duration Problem Result:

- Logistic Regression
- Neural Network
- Simple Decision Tree
- XGBoost

Discussion & Future Work

1. The binary willingness problem illustrated a good result, and we could explore productization.
2. The multi-class duration problem’s performance is poor. Potential reasons are: 1) Not enough data for minority classes. 2) The features we’ve chosen could not effectively distinguish commute duration in minutes granularity.
3. We should explore leveraging RNN for the vector features.
4. We should explore populating missing values with regression algorithms.