1 Match Score Calculation

1.1 Input Arguments

- *commute* - boolean indicating whether trip is one-time or commute.

- *roundtrip* - boolean indicating whether trip is one-way or roundtrip.

- *userType* - driver, rider (or both?).

- *dayOfWeek* - e.g. MonTueWedThuFri, indicating which days of week the trip happens, null if commute=false. (what's the best representation)

- *date* - trip date, null if commute=true.

- *originAdd* - address of the origin.(break into multiple fields?)

- *destinationAdd* - address of the destination.(break into multiple fields?)

- *detourFactor* - how much (at the maximum) user is willing to detour to pick someone up, default to 20%.

- *forwardTime* - start time of the trip going from origin to destination.

- *forwardFlexibility* - flexibility of *forward_time*. e.g. if *forward_time* is 9:00am and *forward_flexibility* is 30 minutes, then any start time between 8:30am-9:30am is acceptable.

- *backTime* - start time of trip going from destination to origin. null if roundtrip=false.

- *backFlexibility* - similar to *forward_flexibility*.

Use subscript 1 and 2 to distinguish between 2 users.

1.2 Type Match Score

Type match score=1 if and only if

$\text{commute}_1 == \text{commute} \ \&\& \ \text{roundtrip}_1 == \text{roundtrip}_2 \ \&\& \ (\text{userType}_1 == \text{driver} || \text{userType}_2 == \text{driver})$.

Otherwise type match score=0.

1.3 Destination Match Score

For simplicity we use the following notations:

- A_1, B_1 as origin and destination of user 1, A_2, B_2 as origin and destination of user 2.

- $d(A, B)$ as travel distance from A to B, $t(A, B)$ as time required to travel from A to B. Both should be obtained from Google Map API.

Assume that user 1 is driver and user 2 is rider, calculate the total distance travelled by driver (including picking up) as $d(A_1, A_2) + d(A_2, B_2) + d(B_2, B_1)$, see how much it increased relative to distance travelled without the rider $d(A_1, B_1)$. Then we compare the increased percentage to driver's *detourFactor*, and calculate a match score:

$$\max\{0, 1 - [(d(A_1, A_2) + d(A_2, B_2) + d(B_2, B_1))/d(A_1, B_1) - 1]/\text{detourFactor}_1\}.$$

If both users are drivers, we can calculate a destination match scores for each of:

- user 1 picks up user 2.

- user 2 picks up user 1.

Choose the better score and inform the users accordingly.

1.4 Time Match Score

For simplicity we use the following notations:

- t_1, tol_1 as forwardTime and forwardFlexibility for user 1.

- t_2, tol_2 as forwardTime and forwardFlexibility for user 2.

- s_1, s_2 indicates how satisfied user 1/2 is about the schedule.

Time t is picked for the trip, notice that s_1, s_2 are both functions of t. Define:

$$
\begin{aligned}
s_1(t) &= \max\{0, 1 - |t_1 - t|/tol_1\} \\
s_2(t) &= \max\{0, 1 - |t_2 - t|/tol_2\}
\end{aligned}
$$

Both drivers can opt-out of the ride share, therefore it is reasonable to choose $\min\{s_1(t), s_2(t)\}$ as the time match score. Therefore goal is to find t that maximizes the time match score. The optimization problem below yields both optimal trip time t and the best possible time match score.

$$\max_t \min\{s_1(t), s_2(t)\}$$

For roundtrips, do the same thing for the trip back and multiply the scores. (For example, if time only match for the trip from origin to destination but not the trip back, then score should be 0 because you can't drop off the rider at his/her destination without any means for him/her to come back)

1.5 Match Score

Each of the three sections above gives a score between [0,1], the aggregated score can be calculated as the product of the three.

Match Score = Type Match Score×Destination Match Score×Time Match Score

1.6 Return

Return results with match score above a certain threshold (0.8?), from high to low.

2 Pricing Scheme

2.1 Observation

A lot of cheap bids posted on zimride which does not make sense. Enforce a lower limit to the amount rider pays.

3 Details to add

- Should be able to match roundtrip with one-way, although in reality most are round trips.

- When both parties are drivers there are two ways of matching.

- Take into account time required for driver to arrive at rider's place, which is not negligible.

- Find closed-form solution for Section time match score.