Machine Learning – Researcher Practical Challenge

The proposed challenge aims to demonstrate the benefits of the construction of a tunnel to improve and existing traffic route and solve the congestion problem. Data of the current traffic is provided by the city council.

The current traffic route

The current traffic route can be approximated by the graph in Figure 1.

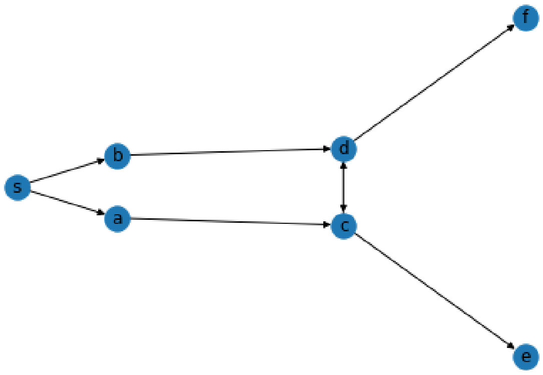

Figure 1: The current traffic route.

There is one starting point, named 's', and two destinations, 'f' and 'e'. We know that 35% of trips are $s \to e$, while 65% are $s \to f$.

Data interpretation

The data provided lacks of fundamental details, so it is necessary to make a few assumptions:

- the time unit provided is seconds. Any larger or smaller time unit would not be compatible with the spacial scale of an urban traffic route.
- The variable named *index* is a way to indicate a set of samples, collected at the same time. It is unclear which actual time of the day $index = 0, \cdots, 99$ correspond, as well as the time window in seconds between two consecutive indices. A possible interpretation of the meaning of the variable *index* is suggested by the plot in Figure 2, which represents the summation of the variable *count* on

all the edges at each index. A linear fit of the points shows almost no slope, which suggests that the different points oscillate around a constant value of the total *count* (or total amount of cars for all the edges), which is 4000. However, the variation of the total *count* between consecutive indices is up to 3000. These observations suggest that each index might refer to far away time samples (e.g. 24 hours apart), rather than consecutive time samples. In conclusion, the interpretation given is that the variable *index* represents the am traffic peak for different days on the same traffic route.

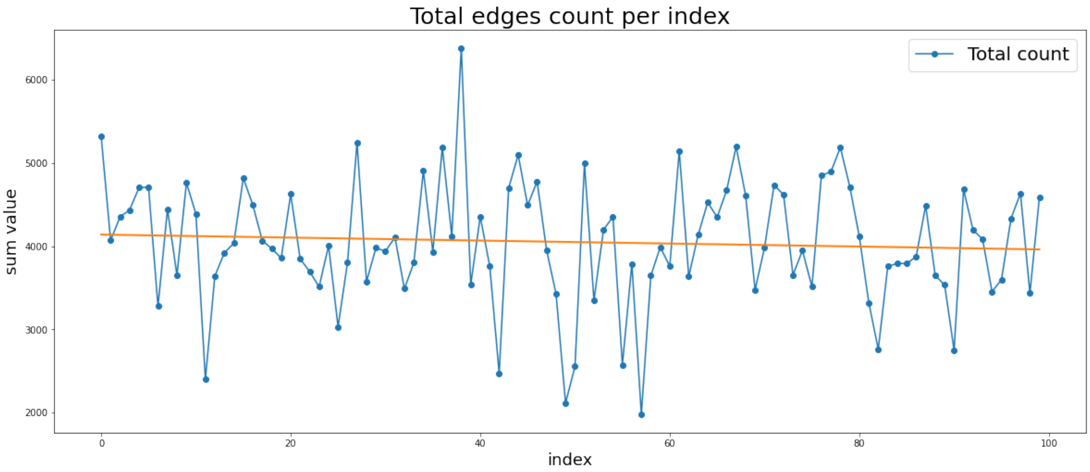

Figure 2: Total edges count per index.

Data visualization

It is useful to look for patterns in the data. Figure 3 contains the variable *journey_time* as a function of the variable *count*. Different colors are used to highlight the different edges, suggesting that the *journey_time* for each edge is directly proportional to the amount of traffic on the same edge. However, both edge 'ac' and 'sb' represent an exception, where the *journey_time* does not increase with the number of vehicles. This suggests that both 'ac' and 'sb' would be good candidates for public transportation, i.e. buses, which should have a limited impact on the *journey_time*, at the same time maximizing the amount of people transported.

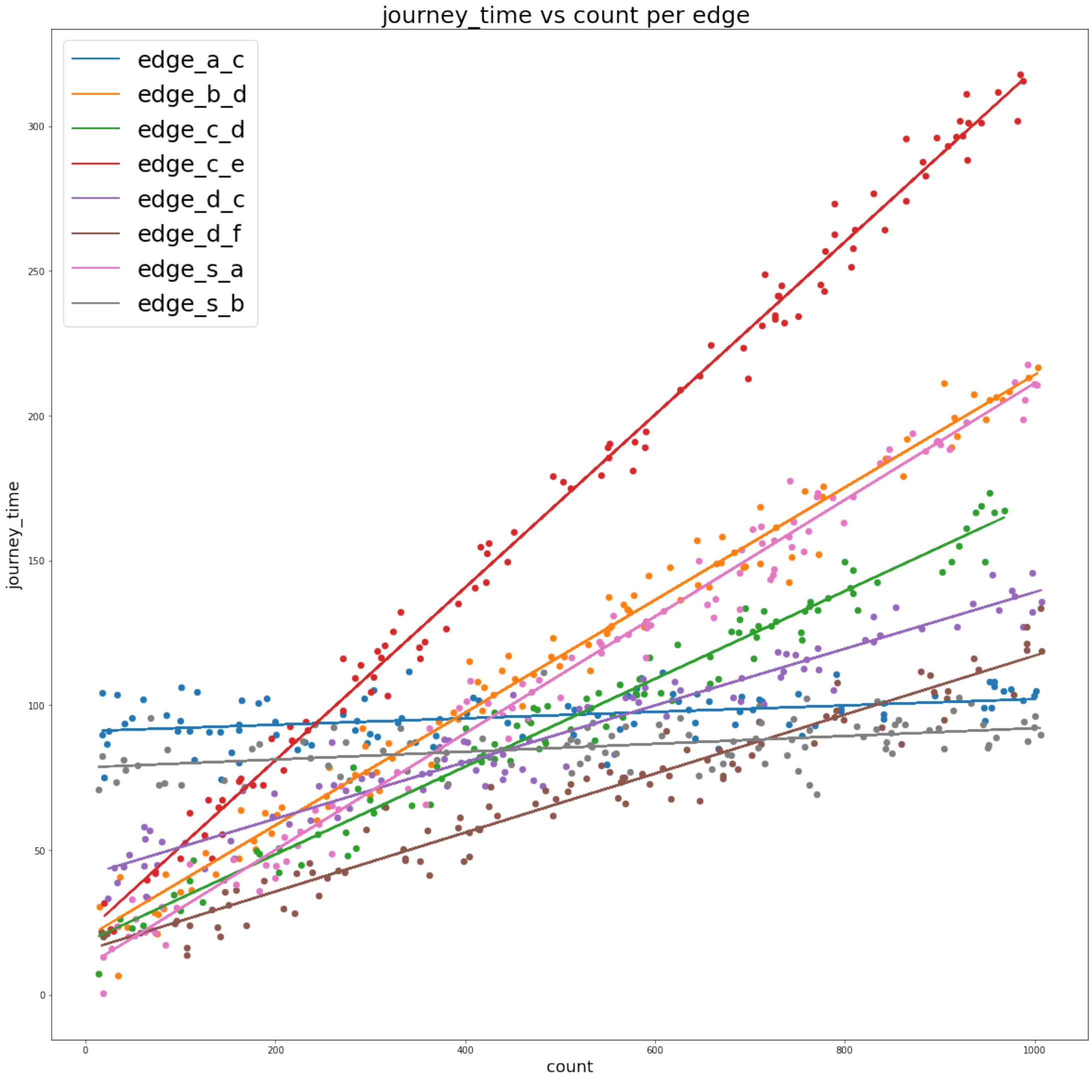

Figure 3: *journey_time* vs *count* per edge.

Assumptions in the proposed model

The proposed model is based on the natural behaviour of people, which is minimising the travelling time to get from a starting point to a given destination.

However, the proposed model is just an approximation of reality, thus it is necessary to make a few assumptions:

- The whole route is assumed to be initialized as completely clear of traffic for each different index. The first driver is likely to find the whole route clear of traffic, depending on his destination, while the next driver will have to take one existing driver into account, to calculate the *journey_time*.

- The model assumes that each driver is able to choose which path is the fastest route. This is generally not true in real life, unless some kind of very precise navigation aid is used to provide information about the length of the journey and the amount of traffic.
- The total amount of traffic used as a variable in the proposed model, corresponds to the variable total *count*, plotted in Figure 2. This same amount of traffic is redistributed when adding the tunnel. Alternatively, total *count* could have been sampled from a Gaussian distribution centred at 4000, but real data have been prioritized for the analysis.

The proposed model

The model works as follows:

1. a time index is selected.
2. The number of total drivers is computed by summing the variable *count* for all the edges at that time index.
3. A driver is selected and assigned to $s \rightarrow e$ or $s \rightarrow f$, respectively with 35% or 65% probability. The very first driver will find the whole route completely free of traffic.
4. In the current route, there are two possible paths of each destination:
   - $s \rightarrow e$: *sace*, *sbdce*
   - $s \rightarrow f$: *sbdf*, *sacdf*
5. Depending on the final destination, the driver computes the $journey\_time$ for each of the two possible paths. The computation of the $journey\_time$ for each path is performed by summing the $journey\_time$ for each edge along that path. More specifically, slopes $a(edge)$ and intercept $b(edge)$ for each edge can be obtained from the linear fits in Figure 3. These values can be used to extract the $journey\_time$ from the variable *count*:

$$journey\_time(edge) = a(edge) * count(edge) + b(edge) \tag{1}$$

   The very first driver, will only require

$$journey\_time(edge) = b(edge) \tag{2}$$

   in order to get through a given edge.

6. The driver selects the route with the shortest total *journey_time*, and is randomly placed on any of the edges along that route. This is equivalent of assuming that drivers depart from s at different time and uncorrelated times.
7. The model is looped, until all the drivers are inserted in the model.
8. The model is looped on all the values of *index*.

Including the tunnel

When the tunnel is added to the previous model, the whole algorithm is almost identical, with few exceptions.

Two additional edges, corresponding to the tunnel routes, should be included, named 'ab' and 'ba'. This enables a few additional paths, thus point 4. should be modified as follows:

- $s \rightarrow e$: *sace*, *sbdce*, *sbace*, *sabdce*.
- $s \rightarrow f$: *sbdf*, *sacdf*, *sabdf*, *sbacdf*

Both the edges require a constant *journey_time* of 4 minutes, corresponding to 240 seconds, thus equation (1) is not used when calculating the *journey_time* on 'ab' and 'ba'.

However, the maximum amount of drivers travelling through the tunnel is 400. This number takes into account both directions. This can be included in the model, simply by excluding the paths containing the edges 'ab' and 'ba' when the total *count* in the tunnel reaches 400.

Results and discussion

Tables 1 and 2 represent the average model prediction of the variables *count* and *journey_time* for each path over all the indices, for the current route and the one with the additional tunnel.

Path	Current route	With tunnel
sace	1418	1405
sbdce	0	14
sbace	-	0
sabdce	-	0
sbdf	2250	1685
sacdf	381	0
sabdf	-	945
sbacdf	-	0

Table 1: Predicted mean *count* for each path.

Path	Current route	With tunnel
sace	250.32	281.28
sbdce	NaN	252.52
sbace	-	NaN
sabdce	-	NaN
sbdf	232.75	436.34
sacdf	269.94	NaN
sabdf	-	360.74
sbacdf	-	NaN

Table 2: Predicted mean *journey_time* for each path.

Table 1 represents the Nash equilibrium, showing that the only strategy for $s \to e$ is 'sace' ('s'→'a'→'c' →'e') , for the model with and without the tunnel. For the case $s \to f$, the main strategy for both models is 'sbdf'. However, the current route has an alternative strategy 'sacdf', which is totally absent in the model with tunnel and replaced with 'sabdf'.

Equation (1) allows to calculate the *journey_time* in Table 2. These values are averaged over the number of drivers following that path, which are retrieved from Table 1. NaNs arise when dividing by the zeros in Table 1, but they can be neglected.

The results in Table 1 are very interesting and counter-intuitive. In fact, adding new possible paths by inserting the tunnel seems to increase the average time for each path, compared to the model without tunnel. This could be explained with the

Braess's paradox, where the drivers choose the most favourable path, rather than choosing the socially optimal traffic pattern.

In reality, several approximations were made, which have been previously mentioned. In a more realistic model, a car driver would assume a variety of approaches, alternative to the best *journey_time* approach here proposed.

- Most of the usual drivers, would just proceed by choosing the path they are mostly used to, simply because they know how it behaves during the am peak or maybe because it is the only one they know, or it is their lucky one or any other reason.
- Part of these drivers, might have a dynamic approach, by switching the path decision right before getting into the next edge. This involves changing their initial decision based on the traffic they see at each edge. This could potentially be included in the model, but requires some additional work.
- Part of the traffic would be due to drivers who have never been travelling on this route before, thus might be unsure about the most favourable path.
- A fraction of the drivers might be using a navigator, which could actually emulate the proposed model, especially if it can provide information about the traffic.

Edge	**Current route**	**With tunnel**
ac	565	466
bd	749	800
cd	95	0
ce	472	475
dc	0	3
df	843	798
sa	569	703
sb	754	565
ab	-	237
ba	-	0

Table 3: Predicted mean *count* for each edge.

Table 3 shows the mean *count* over all the indices for each edge. The effect of the tunnel on the traffic seems to be minimal, according to the proposed model. The tunnel is used only on the edge ‘ab’, while ’ba’ is never chosen. The total count on the tunnel is well below the 400 maximum capacity, which might be good, because the tunnel would be relatively safe. Moreover, the tunnel seem to partially decrease the overall traffic on the other edges, by accommodating 237 drivers which would alternatively increase the traffic on the other edges.

Conclusions

The proposed model does not provide enough evidence to support the construction of the tunnel. In particular, Table 2 shows that there is an overall increase in the $journey\_time$. This is likely to be due to the assumptions and the approximations made.