CHALMERS
UNIVERSITY OF TECHNOLOGY

Traffic Congestion in Urban Modelling

Master's thesis in Complex Adaptive Systems

ERIK ROOS

Department of Space, Earth and Environment
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis FRT 2018:01

Traffic Congestion in Urban Modelling

Erik Roos

Department of Space, Earth and Environment
Chalmers University of Technology
Gothenburg, Sweden 2018

Traffic Congestion in Urban Modelling
ERIK ROOS

Supervisor: Alexander Hellervik, Trafikverket
Examiner: Claes Andersson, Department of Space, Earth and Environment

Master's Thesis FRT 2018:01
Department of Space, Earth and Environment
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX

Abstract

In this thesis, a model for traffic congestion in urban areas is developed using betweenness centrality. The betweenness centrality is augmented with the activity [1] in the network, which corresponds to the assessed value of a property. In this way, shortest paths between two nodes with high activity contribute more to the overall betweenness centrality, and in turn traffic, than between two nodes with low activity. The traffic network is imported from Open Street Map.

In each iteration, the betweenness centrality in the network is calculated and the speeds on the edges are updated according to the VQ-relations [2][3]. A new activity is then calculated with the new speeds. Travel times in the network are finally compared to travel times given by Google Maps.

The model is tried out for a number of settings in the city of Gothenburg. The travel times obtained with the speed limits given from OSM show high linearity but the average travel time is too short. The model manages to improve the travel times to some extent but loses linearity when too much traffic is introduced. Updating the activity in each iteration shows little difference compared to calculating the activity once and keeping it static.

When comparing flows, the model does indeed capture some real-life behaviours but fails to capture others. For example, in the inner city and along the major highway going through Gothenburg the traffic is not slowed down at all in the model.

Future work includes studying different ways of defining the shortest path between nodes (such as topological or angular distance), introducing some through traffic or increasing the geographical calculation area and increasing the resolution of the traffic network (making it directed and have more precise road types).

Keywords: traffic congestion, betweenness centrality, urban modelling

Contents

1
Introduction

In the world of today, it is close to impossible to not be a part of the giant network that is our transport system. As soon as you step out of the door you have to interact with pedestrians, bicycles, cars, busses and whatever you come across. For the most part, the interaction is automatic, there are certain basic rules you have to follow, and for a single agent, this is quite easy to describe and understand. But when many agents have to interact complex features that are not described by these rules arise, for example, congestion.

While the people living in a city often know which areas are crowded and when to avoid them, city planners try to create traffic networks that have as little congestion as possible. An architect might attack this problem from one side, but as a physicist, it is natural to study traffic networks as complex networks. When looking at the problem from a mathematical point of view there are many tools one can use to obtain inherent properties of the network.

Alexander Hellervik is currently doing his Ph.D. on the interaction between urban traffic networks and urban activity and land use. In his model Hellervik takes in just a traffic network and tries to predict where there will be a lot of so-called 'activity', which roughly can be translated into the assessed value of a property [1]. The idea is to keep the number of parameters to a minimum and see what kind of information can be obtained from the traffic network, just the actual roads, of an urban area. This could then be used to, for example, see how the center of a city would change when removing or building a new road. It could also be used as a component of a larger model.

The 'activity' of the network is calculated as an iterated eigenvalue centrality depending on a node's position in the network (how well connected the node is) and travel time to other nodes. How long it takes to get from one node to another is calculated simply by the lengths of the roads divided by the maximum speed of each road. This is highly simplified, anyone who has ever been in a car knows that when driving in a city you often go a lot slower than the speeds on the traffic signs, which motivates the introduction of a congestion model to possibly get more accurate travel times. This is the starting point of this thesis.

While Hellervik wants an answer to the question 'Is it possible to improve the accuracy of the activity model by introducing a congestion model?', the opposite question can also be made: 'Is it possible to use the activity model to get an accurate congestion model?'. This is the problem formulation of this thesis and we will spend the next pages trying to find out.

1.1 Background

There are many models simulating traffic, both on a microscopic level (e.g. an intersection) and macroscopic levels (e.g. the traffic of a large city). Some models are more advanced than others but they often have a lot of parameters. One example is Trafikverket's macroscopical model called Sampers [4]. Sampers simulates traffic on a national level in Sweden. To get a traffic flow the model uses empirical data to describe what trips are being made and how often. Going out and asking people how and where they want to travel works for a national company like Trafikverket but is a bit out of the scope of this thesis. The model gives good results but takes a long time to run. Instead, in this thesis, the idea is to make a faster model using only the traffic network as a parameter.

This limits the number of ways to model traffic. One could use some kind of agent-based model, spreading out agents over the network and make them go to different places and interact with each other. This quickly becomes quite heavy calculation wise. We are looking for something faster and more abstract, preferably something inherent in the network. Using graph theory there are a number of centrality measures that can be calculated for the network [5][6]. The different centrality measures calculate how central nodes or edges in the network are from different aspects. One centrality measure that seems very much to describe some kind of flow in the network is betweenness centrality.

Betweenness centrality for an edge is a measurement of how much 'in between' other nodes a certain edge is. It is calculated by finding the shortest paths between each pair of nodes and then giving each edge a score depending on how many 'shortest paths' that pass through this edge. If an edge has a high value of betweenness centrality, it means that when taking the shortest path from one node to another, chances are high that you would pass through this edge. It is natural to believe that this would in some way reflect the actual traffic flow in a network and there have been several papers showing that there is indeed a correlation [6][7][8][9]. One thing these papers have in common is that they all augment the betweenness centrality in some way.

Altshuler et al. [8] augmented betweenness centrality in a number of ways using the distance between nodes and travel times among other things to achieve better results. They also introduced a mixing step where they mixed the betweenness centrality obtained for a free flow network and that of a congested network, which also improved the results. Furthermore, they also introduced a dependence on the road type for the betweenness centrality, as one, for example, would rather go on a highway when traveling a long distance.

Jayasinghe et al. [7] showed that using a BC focusing on topological or angular distance gave a better result than the actual shortest distance. Topological distance being defined as the number of turns one would have to take from one node to another and angular distance is defined as the total angles turned. Both introduced to describe a behaviour where you prefer to take not the actual shortest path but rather the one perceived as 'straightest'.

Scheurer et al. [6] also introduces a number of other centrality measures to combine with the betweenness centrality to describe a public transport network.

Kazerani et al. [9] also conclude that the betweenness centrality has to be augmented in some way to give meaningful results.

Porta et al. [10][11] also argue that betweenness centrality show important properties of the flow of a network but also look at information centrality and straightness centrality as good candidates to describe the flow.

In this paper, there will also be an attempt to describe traffic using BC. However, with a different augmentation than in previous attempts. In [1] Hellervik calculates activity levels in a city based only on the actual road network. A high activity level of a property would correspond to a high assessed value, which in turn corresponds to economic activity (such as an office or a shopping mall). An area with high activity would in this way generate a lot of traffic (this is not necessarily true, for example, location can have a big effect on the assessed value without actually affecting the generated traffic). If this hypothesis holds, the activity level could be used to augment the betweenness centrality to reflect the amount of traffic. The path between two nodes with high activity contributes more to the overall betweenness centrality than two nodes with low activity.

To summarize, the idea of this thesis is to try to use the betweenness centrality of a traffic network and augment it with the activity calculated by Hellervik to create a model that describes traffic congestion.

2

Theory

2.1 Betweenness centrality and activity

Betweenness centrality is one of many ways to measure how 'central' a node or edge in a network. The betweenness centrality of an edge v (or a node) is calculated as

$$C_{BC}(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{2.1}$$

where $\sigma_{st}(v)$ is the number of shortest paths from node s to node t passing through edge v and σ_{st} is the total number of shortest paths (with the same length) from node s to node t. In words, one calculates the shortest path(s) from all nodes to all other nodes. When a shortest path passes an edge, the edge is given the value +1. Since there can be several shortest paths between two nodes, not all traversing the same set of edges, the denominator makes sure that each set of two nodes only contribute 1 to the total betweenness centrality.

The edges with a high BC would then be very central edges with a high probability of traversing it when going between two nodes in the network. The idea in this thesis is that the betweenness centrality reflects and should be in proportion to the traffic flow.

However, it is easy to tell it will not work straight out of the box. In the betweenness centrality in eq. , all nodes are equal which should not be the case. A path between two nodes that generate a lot of traffic, i.e. two busy intersections in the center of any city should contribute more to the flow of traffic than between two houses in the suburbs. This suggests that the BC needs to be augmented in some way, this is where the activity comes in.

2.2 Activity

The activity is calculated by first dividing the map into cells. This is done by first simply putting a mesh grid over the traffic network. The cells that are close enough to any edge in the network are then connected to the network. The cells are connected by adding a virtual road from the center of each cell to the closest edge in the network. The activity for each cell is then basically calculated as an eigenvalue centrality. Note that cells that are deemed to be 'too far' from the road network are not used.

Using Dijktra's algorithm the travel times between cells, d_{ij}, are obtained. Dijktra's algorithm calculates the shortest path between two cells, here shortest means shortest travel time. The travel time for each edge is taken to be the length of the edge divided by the speed of the edge.

The interaction, D_{ij}, for the cells i and j are then calculated for all pairs in the network, it is defined in [1] as:

$$D_{ij} = (1 + ad_{ij})^{-b}, \tag{2.2}$$

where a and b are constants. As suggested by Hellervik, the constants are set to $a = 0.001$ and $b = 1.5$. D_{ij} is basically just the inverted distance and to avoid unreasonably large values for short distances 1 is added. The normalized activity a_i is then defined as

$$a_i = \sum_j \frac{D_{ij} a_j}{\sum_k D_{jk}}. \tag{2.3}$$

With $M_{ij} = D_{ij} / \sum_k D_{ik}$ we get the eigenvalue equation:

$$a_i = \sum_j M_{ij} a_j, \tag{2.4}$$

which can be solved for a_i. What is obtained when solving this equation is an activity in each cell that is equal to the sum of the activity in all other cells times their respective normalized interactions. A cell with high interactions with other cells with high activity would in turn also have a high activity.

2.3 Augmenting the BC

When performing the calculations for the BC, only the cells that are used when calculating the activity are seen as traffic generators. So the first augmentation to the BC is that we only take into account the shortest paths between the cells, and not between all nodes in the network. The second tweak to the BC is that it is weighted by the activity in the two nodes. In [1] Hellervik gives a formula for the activity flow, p_{ij}, in the network as:

$$p_{ij} = \frac{D_{ij} a_i}{\sum_k D_{ik}} = M_{ij} a_i. \tag{2.5}$$

Comparing with eq. 2.3, p_{ij} is just the contribution of activity from node i to node j. This value is then used in the betweenness centrality equation 2.1:

$$C_{BC}(v) = \sum_{s \neq v \neq t} \frac{p_{st}\sigma_{st}(v)}{\sigma_{st}}. \tag{2.6}$$

This takes into account both the activity in the origin and the destination (since the flow in from j to i is also added) but also the length of the shortest path. This is supposed to reflect that there should be a higher flow of traffic between two nodes with high activity on a close distance than between two nodes with low activity that are far away from each other.

2.4 Traffic flow

With the augmented BC calculated, a traffic flow is then spread out over the network. As there is no inherent amount of traffic in a network, a way to define a total number of trips in the network is needed. The total number of trips per hour is set to a factor of the population of the city in question as

$$F_{city} = \beta_{trips} \cdot P_{city}, \tag{2.7}$$

where F_{city} is the total number of trips per hour in the city, β_{trips} is the 'trip factor' and P_{city} is the population of the city. The trips are then spread out on the edges according to the betweenness centrality of the edges.

While this is a very easy way to get an amount of traffic there are some drawbacks to this way of defining it. The percentage of drivers in Gothenburg would (probably) be lower than an arbitrary city in the U.S. and in the same way larger than a poorer town, somewhere in the more undeveloped part of the world.

Another drawback is that the flow is static, in real life the traffic flow varies a lot during the day. To get good results one would have to use different amounts of flow during different parts of the day. Furthermore, the flow of traffic can in itself change the total amount of traffic. If taking the car would mean that you would get stuck in congestion on your way to work, people might choose different transport alternatives.

But it is, for now, of less importance since the goal of this project is to try to find congestion behaviours using BC, not to find a good way to estimate the number of cars on the road.

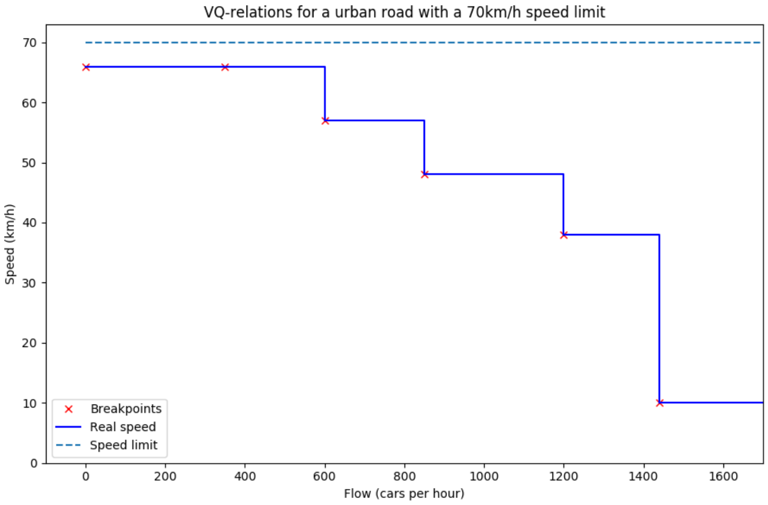

Figure 2.1: The VQ-relations show empirical data on how fast people actually drive depending on the flow of cars. As the flow passes certain breakpoints the speed goes down. Note that even when there is no traffic the speed is lower than the allowed speed limit.

2.5 VQ-relations

Another crucial part of the model is the so-called VQ-relations. Trafikverket has measured 'actual' speed on different kinds of roads [2][3]. The 'actual' speed of a road depends (obviously) on allowed speed but also on what kind of road it is and where the road is situated. People tend to drive faster than the speed limit on roads in the countryside, while they drive slower than the speed limit in urban areas. Furthermore, the actual speed of a road depends on the amount of traffic. Each road type has a certain capacity with certain breakpoints. If the flow is below the first breakpoint then every car can go with maximum speed. As mentioned before, the maximum speed is not necessarily the speed given by the road signs (it is often lower but in some cases also higher). After passing the first breakpoint, the traffic is so high it starts to affect the driving speed. At first, it is just a little bit slower but by each breakpoint passed the speed gets lower and lower until after the last breakpoint when the speed is set to 10km/h, see figure 2.1.

Trafikverket has VQ-relations for a number of road types depending on the number of lanes, sight, speed and if it is in a rural or urban area. The classification in OSM is, unfortunately, a bit cruder. The roads in OSM are classified only by name as, for example, 'motorway', 'residential' or 'primary'.

In the model, each piece of road from OSM is approximated to the one that seems to fit best with the VQ-relations. This approximation is done by hand and because of the discrepancy of how the road types are defined it is a bit rough. The approximation is based on the speed of the edge and position of the road. For example, a road classified as ’residential’ by OSM would get the VQ-relation of ’V-Q centrumområde 2-fält Citygata’ for the speeds 40km/h and 50km/h and ’V-Q mellanområde 2-fält Tangent’ for 60km/h and 70km/h. The other road types in OSM are approximated in the same way but with different VQ-relations.

2.6 Algorithm and mixing betweenness centrality

With these pieces we can form the algorithm for the model, it goes as follows:

1. Calculate the shortest paths between nodes using Dijkstra’s algorithm.
2. Calculate the activity in the different cells.
3. Calculate the betweenness centrality given by the cells and weight it by the activity.
4. Mix the new betweenness centrality with the previous one to create a mixed betweenness centrality (optional).
5. Calculate an estimated number of trips on each edge with the population and the augmented betweenness centrality.
6. Update the speeds on the edges according to the VQ-relations.
7. Start over from 1 with the new speeds.
8. Compare the travel times to Google Maps.

Step 4 is introduced to keep the traffic flows from changing too fast. With a constant $0 \leq \alpha < 1$ the newly calculated betweenness centrality B_{new} in each iteration is mixed with the previous betweenness centrality B_{prev} to form a mixed betweenness centrality B_{mixed}:

$$B_{mixed} = \alpha B_{prev} + (1 - \alpha) B_{new}, \tag{2.8}$$

where $\alpha = 0.24$ as suggested by [8]. This value is then used as the new betweenness centrality for the following calculations.

2.7 Checking the results

In the last part of the algorithm, when the new speeds of the edges have been calculated, it is time to check the results. This is done by first selecting a number of pairs of start and end points in the network and calculating both the travel times and length between these nodes. This is then compared to what Google Maps would say for the same pairs of start and end points. By comparing both the baseline model (with unaltered speeds) and each iteration with Google Maps, one can see how the model develops. With Google Maps Distance Matrix API [12] both the travel time and length of a given route are easily available. Google Maps was chosen for its availability and flexibility (you can compare any routes in the network).

There are some problems with comparing with Google Maps. We can't control how the data is gathered nor which parameters are taken into account. In the simple model of this report, no heed is taken to traffic lights, intersections nor acceleration, which *is* taken into account to some extent by Google Maps. The simulated data will probably not be exactly as Google Maps', even with a perfect model, but comparing with Google Maps should give an indication if it is correct or not. Keep in mind that Google Maps' data is not perfect either.

The comparison with Google Maps is done in two ways. The first one is straightforward, it is just the average ratio of the travel time of each route calculated as:

$$C = \frac{1}{N}\sum_{ij}\frac{Calculated(i,j)}{Google(i,j)}, \tag{2.9}$$

where N is the total number of pairs being compared. The average ratio shows how much, on average, the travel time of each route has improved. This has the drawback that we do not know in what way it is improving/or getting worse. By increasing the travel times of some routes a lot while decreasing for others could give the same results as increasing the travel times of all routes a little. To be sure this is not the case another correlation is introduced, the Pearson correlation. The Pearson correlation, r, measures the linearity between two sets of data and is defined as

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}, \tag{2.10}$$

where x, y are the two datasets, $\bar{x}, \bar{y}$ are the averages of each dataset and n is the number of data pairs. Furthermore, $r = 1$ corresponds to total positive linearity, $r = 0$ is no correlation at all and $r = -1$ is totally negative linearity. A perfect model compared with the true travel times of the routes would give $r = 1$.

Note that the Pearson correlation does not say anything about the actual difference between the two datasets, just that there is a linear relation. So $r = 1$

does not mean that there is a 1:1 relation between the datasets, this is why we need the first measurement.

Of the two measurements, the Pearson correlation is the most important since it says more about the overall behaviour of the network.

2.8 Cost

In the previous attempts at using BC to describe traffic flow, different suggested augmentations of the BC [7][8] have been made, or rather how you calculate it. When calculating the shortest paths between nodes, you can instead of using distance define a *cost* to go between nodes. There are different ways to define this, for example, the actual price in money you have to pay for fuel costs etc. or the number of intersections and stoplights.

There are also more abstract ways of defining the price, reflecting that it does not actually matter how short a road is, the only thing that matters is how it is *perceived*. One example is using an angular distance, where a high price would be to make a lot of sharp turns. Another one is just the number of turns themselves, people would rather continue straight when it is perhaps more efficient to make a couple of turns. It is simply easier to just continue straight.

In this thesis, the travel time is used as a cost because it is a logical way to measure distance but also because of the limited information from OSM, as there is little information about intersections and angles.

3
Method and data

The model is implemented in Python. For convenience and relevance, the simulation is, for the most part, performed on Gothenburg. The network is imported to Python with the package OSMnx [13] as a NetworkX graph [14]. The activity is then calculated, followed by the betweenness centrality. The betweenness centrality is then weighted by the activity and, if used, mixed with its predecessor. A traffic flow is then spread out according to the betweenness centrality and new speeds for the roads are obtained according to the VQ-relations. This is then repeated for a number of iterations or until convergence is met. Finally, the result is compared to Google Maps.

3.1 Settings

The model is tried out for five different settings:

1. Static, the activity is calculated once and is not updated in each iteration.
2. Mixed, the resulting BC in each iteration are mixed with the BC in the previous iteration.
3. Static and mixed.
4. Non-static and non-mixed.
5. No activity and no mixing (only betweenness centrality).

Each setting is executed for five iterations with a range of different numbers of total trips to find the best fit. Each setting was evaluated with the same paths given by Google Maps. Comparing how the relations with Google Maps develop in each iteration compared to the baseline model, one can see if the model improves or not.

The measurements are done three times, one for rush hour traffic in the morning, on for rush hour in the afternoon and another one in the middle of the night when there should be no congestion.

For reference, the model is also tried out for the cities of Lund, Sweden, and Verona, Italy.

3.2 Data

The traffic network is taken from Open Street Map (OSM) [15]. OSM is a crowdsourced map where users themselves can add data. OSM was chosen for its convenience, being both free and easy to access. To import the traffic network the Python module 'OMSnx' was used [13]. OMSnx imports a network from OSM to python so one can use the Python package NetworkX [14]. NetworkX is a package for graph theory with a lot of built-in functions.

While the VQ-relations themselves are taken from empirical data, how to classify each road to align with the classification of Trafikverket is not trivial. OSM is in some aspects very accurate and there is data on all kinds of roads from small walking paths to highways. While data such as distance and speed limits are accurate and easy to interpret, there are other aspects that are harder to use straight out of the box. For example, the lane counts can be off. At an intersection it is common that a road that previously had one or two lanes splits into three or even four lanes. As a result, the whole edge has three or four lanes. The number of lanes is approximated to fit the VQ-relations, which can cause some errors.

Also, the classification of different roads differs. Since it is editable it seems like what some people would classify as a certain road type, other people would call something else. This together with the fact that how OSM and Trafikverket classify roads differs also affects the chosen VQ-relations.

Another thing to think about is as the model is implemented now it is possible to drive in both directions on all roads. And while this is not a problem overall, some roads will be wrongly classified. Furthermore, this can cause problems on a one-directional road, as when going on or of a highway or when at a traffic circle.

4
Results

When comparing using a static activity and/or mixing the BC the best results were obtained when mixing but it mattered less if the activity was static or not. It was only for high amounts of traffic that it was possible to see a difference. In the following results, the model with static activity and mixed BC was used. The runtime for each iteration was of the order of calculating the activity.

4.1 Correlations

In Gothenburg, the model gave the best results when comparing with Google Maps on a Tuesday at 16:30. The resulting relations can be seen in tables 4.1a and 4.1b. Looking at the relations, increasing the traffic flow causes the average ratio to go up, giving a better relation with the travel times given by Google Maps. However, it is not the same for the Pearson correlation. For small amounts of traffic, the Pearson correlation is more or less unaffected while the average ratio improves. But as the traffic increases the Pearson correlation goes down. It seems like there is a trade-off in the model between a good average ratio and a good Pearson correlation. The best results were obtained for the third iteration with a trip factor of $\beta = 0.06$.

In figure 4.1, the calculated travel times are plotted against the travel times given by Google Maps. The orange line is the 1:1 relation while the green line is fitted to the data. For shorter routes, the relation is highly linear but as the length of the routes increase the travel times are more spread out. After a couple of iterations, most of the travel times have moved towards the orange line, giving a better average ratio. But, since some travel times have not changed, the Pearson correlation suffers.

4.2 Traffic flows

A graphical comparison of the congestion in the network on a Tuesday at 16:30 in Gothenburg can be seen in figures 4.2 and 4.3. In the figures, red corresponds to

Gothenburg, Tuesday, 16:30		Iteration					
		0	1	2	3	4	5
Trip factor	0.02	0.557	0.568	0.568	0.568	0.568	0.568
	0.04	0.557	0.576	0.572	0.574	0.573	0.575
	0.06	0.557	0.596	0.583	0.584	0.584	0.583
	0.08	0.557	0.63	0.591	0.598	0.597	0.599
	0.12	0.557	0.693	0.647	0.652	0.621	0.678
	0.14	0.557	0.724	0.679	0.665	0.665	0.692
	0.16	0.557	0.753	0.717	0.693	0.688	0.719
	0.18	0.557	0.774	0.753	0.701	0.752	0.755
	0.20	0.557	0.798	0.785	0.713	0.742	0.74

(a) Average ratio

Gothenburg, Tuesday, 16:30		Iteration					
		0	1	2	3	4	5
Trip factor	0.02	0.855	0.854	0.854	0.854	0.854	0.854
	0.04	0.855	0.854	0.853	0.853	0.854	0.854
	0.06	0.855	0.844	0.845	0.853	0.844	0.852
	0.08	0.855	0.802	0.842	0.834	0.829	0.833
	0.12	0.855	0.748	0.795	0.765	0.802	0.768
	0.14	0.855	0.733	0.777	0.767	0.79	0.769
	0.16	0.855	0.73	0.768	0.762	0.785	0.75
	0.18	0.855	0.728	0.737	0.756	0.778	0.692
	0.20	0.855	0.72	0.729	0.75	0.765	0.735

(b) Pearson correlation

Table 4.1: The tables show the different correlations obtained for the model on a tuesday at 16:30 in Gothenburg. The average ratio increases with the traffic but the Pearson correlation goes down after the flow factor reaches $\beta = 0.06$.

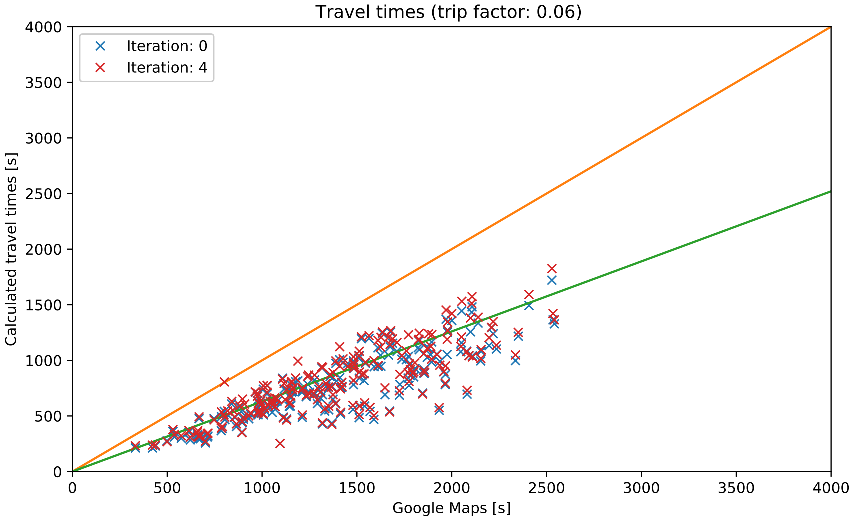

Figure 4.1: The figure shows the calculated travel times plotted against the ones given by Google Maps. The orange line is the 1:1 relation while the green line is fitted to the data. Most of the data points move towards the orange line after a couple of iterations.

that the flow has passed a high breakpoint in the VQ-relations. While the model seems to predict some congestion quite well, especially along some of the bigger roads, it seems to do a bad job at predicting the congestion in the inner city. For example, Korsvägen is marked red by Google Maps while in the model it is barely affected at all. One thing to note in figure 4.2 is the congestion along highway E6 and from Gullbergsvass to Majorna, both roads passing 'through' Gothenburg. Google predicts heavy traffic while the model predicts very little traffic.

Figure 4.4 shows the betweenness centrality calculated for Gothenburg. It is worth to note how different it looks compared to the predicted congestion in figures 4.2 and 4.3. A high flow of traffic does not necessarily lead to congestion.

4.3 No activity

To try the hypothesis that using the activity to augment the betweenness centrality would give better results, a run was made without using the activity. The result can be seen in table 4.2. As when increasing the amount of traffic, the average travel time improves when not using activity but the linearity goes down. The Pearson correlation is higher when using the activity.

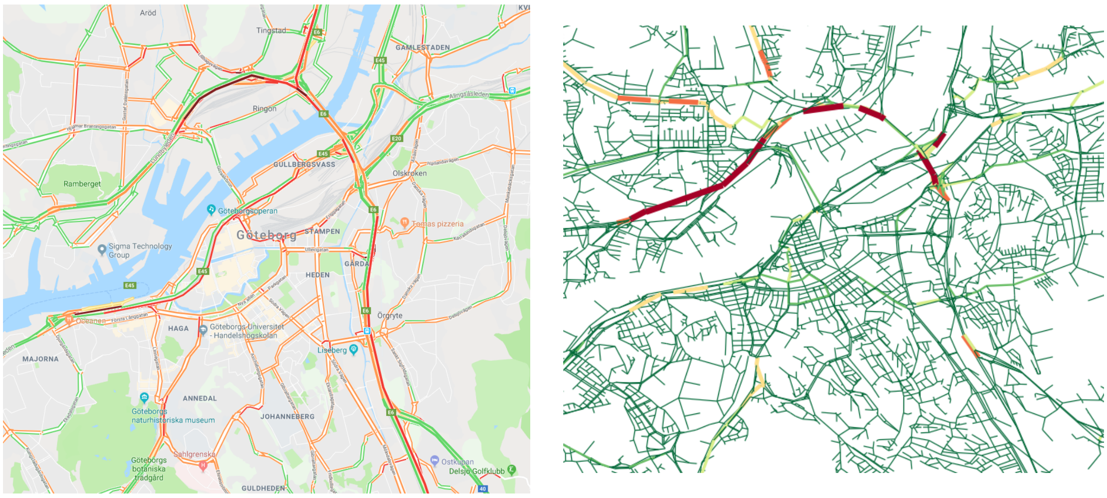

Figure 4.2: The figures show the congestion given by Google Maps (left) from central Gothenburg on a Tuesday at 16:30 and the best corresponding results given by the model (right). The colors in the figure to the right correspond to the breakpoints in the VQ-relations, red being the worst. The model captures some behaviours. Looking at the 'bend' in the north it seems to work quite well while in the more central parts in the south it does not work as well. Furthermore, the model fails to capture the behaviour along highway E6.

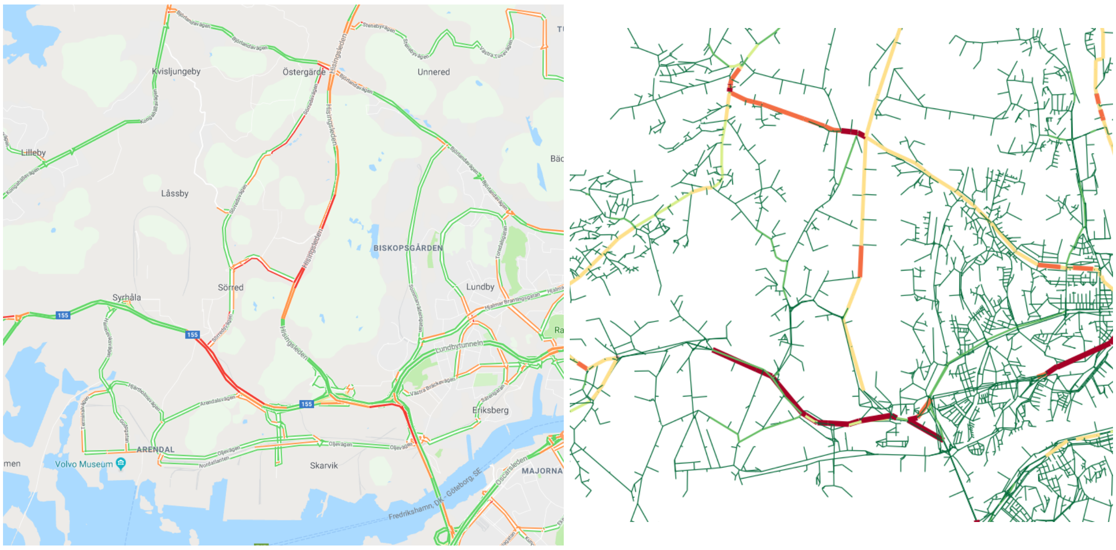

Figure 4.3: As in figure 4.2 the figures show the congestion given by Google Maps (left) from central Gothenburg on a Tuesday at 16:30 and the best corresponding result given from the model (right). The model catches some congestion well while it misses others.

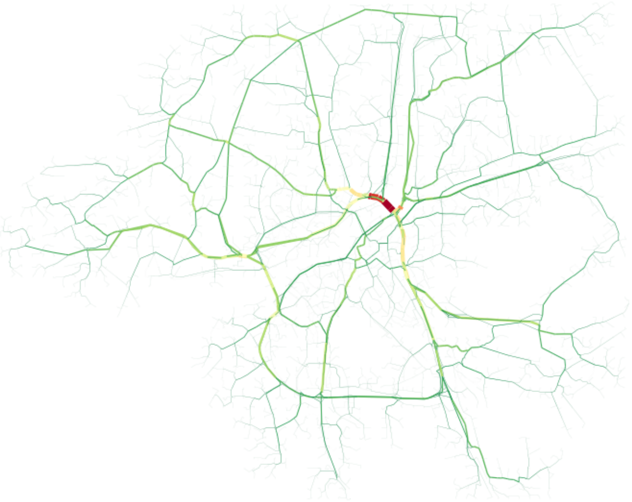

Figure 4.4: The figure shows the resulting betweenness centrality in Gothenburg, where red means a high value. The BC predicts that there will be heavy traffic in Tingstadstunneln. A high flow of traffic does not necessarily lead to congestion.

Gothenburg, Tuesday, 16:30		Iteration					
		0	1	2	3	4	5
Average ratio	Act.	0.557	0.596	0.583	0.584	0.584	0.584
	No act.	0.557	0.639	0.591	0.613	0.597	0.605
Pearson correlation	Act.	0.855	0.844	0.845	0.853	0.844	0.852
	No act.	0.856	0.802	0.835	0.831	0.822	0.821

Table 4.2: The table shows the different correlations obtained when and when not using the activity to augment the betweenness centrality. While the average ratio has improved when only using BC, the linearity is better when using the activity. A trip factor of $\beta = 0.06$ was used.

4.4 Convergence and mixing

Looking at tables 4.1a and 4.1b, the speeds of the roads do not seem to converge. Even for longer runs as in table 4.3, it does not reach convergence, instead, it oscillates between two levels. The mixing step was introduced to encourage convergence and clearly does not work as intended. It did, however, give better results in the model compared to when not mixing.

		Iteration							
		13	14	15	16	17	18	19	20
Trip factor	0.06	0.853	0.843	0.853	0.845	0.853	0.843	0.853	0.845

Table 4.3: To see if the model reaches convergence a longer run was made. Not even for longer runs it reaches congvergence, instead it seems to oscillate betwenen two levels. For this run the model was static and the BC was mixed.

4.5 VQ-relations

To see how well the VQ-relations work, runs were made when just updating the speed according to the VQ-relations with no traffic, see table 4.4.

Baseline	VQ
0.557	0.567

(a) Average ratio, afternoon.

Baseline	VQ
0.856	0.853

(b) Pearson, afternoon.

Baseline	VQ
0.743	0.756

(c) Average ratio, night.

Baseline	VQ
0.956	0.960

(d) Pearson, night.

Table 4.4: The tables show the measured relations for the baseline model and after the speeds have been updated with the VQ-relations. There is a slight improvement both in rush hour and in the middle of the night. Note that there is no implemented traffic in either of the cases.

Both in the afternoon and in the middle of the night the average travel time improves. For the Pearson correlation, we only see an improvement in the night, while it gets a bit worse in the afternoon.

Monday, 16:30		Iteration					
		0	1	2	3	4	5
Average ratio	Lund	0.557	0.606	0.583	0.602	0.589	0.601
	Verona	0.458	0.557	0.513	0.543	0.51	0.545
Pearson correlation	Lund	0.822	0.81	0.838	0.81	0.839	0.811
	Verona	0.892	0.874	0.891	0.874	0.89	0.873

Table 4.5: The model seems to work well for Lund, there is improvement in both the average ratio and in the Pearson correlation. For Verona the results are similiar to those of Gothenburg. The flow factor was $\beta = 0.18$ for Lund and $\beta = 0.06$ Verona.

4.6 Other cities

The model was also tried on the cities of Lund, Sweden, and Verona, Italy. The best runs can be seen in table 4.5. The model gives good results for Lund while it gives results similar to Gothenburg for Verona. Note that the average ratio before the first iteration is the same for Lund and Gothenburg while it is a bit lower for Verona.

5
Discussion

The model gave some mixed results. In most cases, the model did not show any improvement compared to the baseline model and when it did it was only a slight improvement. That being said, there are still things to point out and conclusions that can be drawn to improve future work.

5.1 Baseline model

One thing to note first of all is the good correlation given when just using the speeds given by the network. While the travel times themselves are a bit off the linearity is high. This suggests that perhaps an advanced traffic model is not needed to get an accurate activity model. One can instead just decrease all the speeds with a factor. In this way the linearity is conserved but the travel times better match those given by Google Maps. This is a lightweight solution and is easy to implement but has the drawback that the constant varies over the day and would be different from city to city (and even from area to area in the city). We saw in table 4.4 that Verona would have different constant than Gothenburg/Lund (which suggests that traffic congestion is a bigger problem in Verona than in Gothenburg/Lund).

If one continuously works with the same city this could be a good solution to make the traffic more realistic. And as long as you are working with urban areas, even a moderate decrease in speeds could still improve the activity model.

5.2 Time of day and asymmetrical flow

As mentioned before, there is no real time dependence in the model. The idea is that there should be a relation between the time of day and amount of traffic (trip factor). In the morning, when people go to work, there should be one number of total trips while during the day it should go down and then increase again when people want to go home. In the model, this is controlled by changing the amount of traffic but the model still gives better results in the afternoon than in the morning. The different results probably come from the fact that the flows during the day are asymmetrical.

As the model is built, the flow is not biased. In real life, one goes *to* work in the morning and *from* work in the late afternoon. This causes different traffic flows and subsequent congestions, on different sides of the bottlenecks. The model takes no heed to this in the same way. The flows are constant and the same in both directions. In combination with using an undirected network, this poses a problem.

Making the network directed could probably improve this, one would in many cases effectively get two separate networks. Also, the traffic junctions at motorways would be affected for the better as the flow would be more realistic. Traffic circles would also benefit from this. As mentioned before the model does not capture the congestion at Korsvägen, which is a real problem.

Another reason for the discrepancy between morning and afternoon can be attributed to that office buildings can have very high assessed value without any people living there. An office works a traffic consumer in the morning but as a generator in the afternoon. The model superimposes these behaviours and seems to fit better in the afternoon.

One solution to this is to in some way classify the activity into different categories. If one could distinguish between what is an office and an apartment house, it would be possible to implement a directed flow of traffic. In this way, one could also get a time dependence in the model.

5.3 VQ-relations

Updating the base speed on the edges after the VQ-relations does seem to improve the correlations, see tables 4.4. This is expected as it is empirical data from Trafikverket. The gain is not great but updating the speed on each edge after the VQ-relations is computationally light and could easily be implemented in Hellerviks model.

While it is hard to tell anything about how much the VQ-relations make a difference in the rest of the model these results suggest that the VQ-relations hold and that the fault is at the modelled traffic and not in the VQ-relations.

As stated above there is a certain discrepancy between how Trafikverket and OSM classify their edges. As a result, the edges in the model are approximated to some extent. Making a more detailed translation of the classification of the edges would probably improve the correlations.

It is good that the model improves a little bit when just using the VQ-relations but the main reason for using the relations is how the speed drops with an increased traffic flow. This behaviour is key to updated the speeds and getting a dynamic traffic flow.

5.4 Traffic flows

Looking at figures 4.2 and 4.3, the model captures some aspects of the traffic flow. The model predicts that there should be very heavy traffic in the 'bend' north of the river, as does Google maps. Furthermore, to the west just above where it says 'Oceanen' both Google Maps and the model says there should be traffic congestion.

Along highway E6 there is a lot of traffic according to Google Maps but this is not captured at all by the model. This can probably be attributed to the fact there is a lot of through traffic in real life. A lot of the traffic along all of the west coast passes through Gothenburg. Furthermore, all the traffic in the model is generated and 'consumed' inside of Gothenburg. Adding some kind of flow through the city would handle this problem and give a more realistic traffic flow. In the same way, increasing the computational area would probably be good to improve the flows in central Gothenburg. Unfortunately, the computations were done on a computer which can only handle networks roughly the size of Gothenburg.

While the model captures some aspects of the traffic flow, large areas of the inner city are unaffected in the model. The model does not add any penalty for stop lights or intersections, which does not pose a big problem when riding on a highway but is highly relevant when traversing the inner city. Implementing some kind of punishment in speed for traffic lights and intersections could take care of this. Changing the cost to take into account number of turns, as suggested by [7], would also change the flow in the central parts of the city where there are a lot of intersections.

5.5 Using Google Maps as the correct traffic

Google Maps was used for its convenience and availability and not because it represents the truth. While it is important to bear this in mind, it still gives a good indication of how the traffic flows compare to the real flows. It is possible that using a different source for the travel times would change the results in section 4 but one would have to give up the flexibility of the Google Maps data where you can compare virtually any points anywhere.

5.6 Activity

The motivation for using the activity to augment the betweenness centrality is twofold. The first reason is simply that the activity is the main thing being calculated in Hellervik's model and it would be neat to have something that depends on the activity and could be updated in each iteration. In this way, it is natural to try to use the activity.

Furthermore, the activity should correspond to the assessed value of a property. The assessed value depends, among other things, on the size of the property. The model is based on the assumption that this corresponds to population size and that traffic is generated accordingly. Using the activity you can get around the problem that each node does not generate or consume the same amount of traffic.

There are some problems with this. For example, houses close to the sea usually are quite expensive and have a high taxation value while not housing so many people. Another example is a central office building or school, where the assessed value can be high but having no one living there. As mentioned before, it is possible that this contributes to the discrepancy we see when comparing the traffic flows in the morning and the afternoon. In the model, offices would generate traffic all through the day, while it actually almost only generates traffic when people are going *home* from work.

The activity used in the model is a simplified model, Hellervik uses a more advanced model that gives a more accurate activity. The simplified model is used to avoid singularities and ensure convergence of the activity, which is not always the case in the more advanced model. Using the more advanced model could give a better traffic flow.

6
Conclusions and future work

There is very little in-data in the model, only the road networks and a populatin, and the model still manages to capture some real traffic behaviours. While there is still room for improvement, the results obtained in the report show that betweenness centrality can indeed be used to model traffic.

The model does a good job of capturing some aspects of the traffic flow while it has a hard time capturing others and there are certain areas that can be improved to get better results. These areas can be divided into two parts, namely how to model the traffic network and how to perform the calculations.

When it comes to the network one should take in a larger network than just the city itself, since the traffic in a city is not a closed system. In addition to this, some through traffic should be added. Especially a city like Gothenburg with its important geographical setting this is important as all traffic on the west coast passes through the city. Furthermore, the network should also be directed and having more exact road types could give better results. Introducing some kind of penalty for intersections and turns is also a future prospect.

For the calculations, trying out different ways to define the shortest path with topological or angular distance instead of metric distance could improve the model. Introducing one or more other centralities such as information centrality or straightness centrality as suggested by Porta et al. [10][11] is also something that could be explored.

An inherent time dependence is also something that is needed to get a robust model. One way of doing this is to have different activity types. If one could distinguish what is an office and what is a home, the flows could be directed and also time dependent.

Bibliography

[1] A. Hellervik. Statistics and dynamics in urban modelling. Unpublished manuscript, Chalmers, Gothenburg, Sweden, 2017.

[2] Trafikverket, Borlänge. *Bygg om bygg nytt: Bilaga 4.1 VQ-samband 2014 Landsbygd*, 2014.

[3] Trafikverket, Borlänge. *Bygg om bygg nytt: Bilaga 4.2 VQ-samband 2014 Tätort*, 2014.

[4] Trafikverket, Borlänge. *Sampers trafikprognoser - en kort introduktion*, first edition, 2015.

[5] I M L N Jayaweera, K.K.K.R. Perera, and Jayantha Munasinghe. Centrality measures to identify traffic congestion on road networks: A case study of sri lanka. 13:13–19, 04 2017.

[6] Jan Scheurer, Carey Curtis, and Sergio Porta. Spatial network analysis of public transport systems: Developing a strategic planning tool to assess the congruence of movement and urban structure in australian cities. 01 2007.

[7] A. Jayasinghe, K. Sano, and H. Nishiuchi. Explaining traffic flow patterns using centrality measures. *International Journal for Traffic and Transport Engineering*, 5(2):134–149, 2015.

[8] Y. Altshuler, R. Puzis, Y. Elovici, S. Bekhor, and A. Pentland. Augmented betweenness centrality for mobility prediction in transportation networks. *Finding Patterns of Human Behaviors in Network and Mobility Data*, 2011.

[9] Aisan Kazerani and Stephan Winter. Can betweenness centrality explain traffic flow. 01 2009.

[10] Sergio Porta, Paolo Crucitti, and Vito Latora. The network analysis of urban streets: A primal approach. 33:705–725, 02 2006.

[11] Sergio Porta, Paolo Crucitti, and Vito Latora. The network analysis of urban streets: A dual approach. 369:853–866, 09 2006.

[12] Google Inc. Google Maps Distance Matrix API. `https://developers.google.com/maps/documentation/distance-matrix/`, 2017.

[13] G. Boeing. Smnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65(126-139), 2017.

[14] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.

[15] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.