Proceedings of the SWI 2014 Held in Delft

Calculating Traffic based on Road Sensor Data

Rob Bisseling[1],
Fengnan Gao[2],
Patrick Hafkenscheid[3],
Reijer Idema[4], *
Tomasz Jetka[5],
Valia Guerra Ones[6],
Debanshu Ratha[6],
and Monika Sikora[7]

[1] Universiteit Utrecht
[2] Universiteit Leiden
[3] VU Amsterdam
[4] VORTECH
[5] IPPT, Polish Academy of Science
[6] TU Delft
[7] TU Wrocław, Poland

Abstract

Road sensors gather a lot of statistical data about traffic. In this paper, we discuss how a measure for the amount of traffic on the roads can be derived from this data, such that the measure is independent of the number and placement of sensors, and the calculations can be performed quickly for large amounts of data.

We discuss how a graph of the road sensors can be constructed, and how the number of cars and car-kilometers can be estimated on this graph. Further, methods for dealing with missing data are presented, and the benefits of principal component analysis are discussed.

KEYWORDS: traffic index, road sensor, graph construction, statistical imputation, principal component analysis, CUR decomposition.

*Corresponding author, email: reijer.idema@vortech.nl

1 Introduction

This paper reports on findings regarding the *traffic index* problem, as posed by the CBS (Centraal Bureau voor de Statistiek [NL], Statistics Netherlands [EN]) at the Study Group Mathematics with Industry 2014, held at the Delft University of Technology.

The problem posed to us was to determine a traffic index comparing the average traffic on the highway system in the Netherlands (or a region such as South Limburg) from a particular year to a previous year, based on traffic measurements by road sensors (such as inductive loops, traffic cameras, etc.) taken every minute of the day, every day of the year. We call this problem TI. We decided to tackle an even more ambitious problem, which we call problem C, namely estimating how many vehicles (cars) there are on a particular road, at a given moment in time, based on the measurements of the past minute. Solving this instantaneous estimation problem by computing the number of cars C on the road, will also give a solution to the TI problem, by averaging over all the minutes of the year, and adding the results for all the roads of the network.

An advantage of tackling the more general problem C is that a solution can give more detailed information, such as changes in traffic patterns during the day, or differences between different days of the week, and it also enables zooming in on certain regions, roads, or even road segments. Another advantage of trying to achieve a precise estimate of an actual physically meaningful number is that such an approach is fault tolerant and adaptable to changes, e.g. new road sensors appearing, old sensors being removed, and some sensors malfunctioning temporarily.

The increase in our ambitions from computing mere statistical indicators to achieving a very precise estimate of the actual situation on the road is possible because of the wealth of detailed data that is now available. So to speak, Big Data is driving analysis from statistics towards detailed answers to specific questions on the systems and subsystems studied. For this approach to work, it is essential that computational algorithms become available that are efficient and preferably scale linearly in the number of data entries.

To formulate our problem, we define a number of variables, where we first consider a road segment between sensors A and B. Define d_{AB} as the distance between sensor A and sensor B. Let t be the current time and T the time interval of measurement (1 minute in our data). The measurements at time t are the *intensity* $I_A(t)$ of the traffic, i.e., the number of cars measured at sensor A in time interval $[t, t+T)$, and the average *velocity* $v_A(t)$ of the traffic measured at sensor A (for our data, the arithmetic average during the time interval). Furthermore, a road segment concerns one direction, see Figure 1.

In Section 2, we discuss methods to construct a graph of the road network

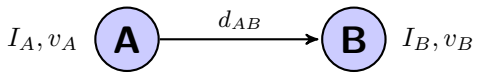


Figure 1: Road segment from sensor A to sensor B, with length d_{AB} and at each sensor the measured traffic intensity I and velocity v.

from the data, such that the edges are road segments as in Figure 1, and in Section 3 methods are presented to calculate traffic indices based on car count and on car-kilometer count, using the constructed graph of road sensors. In Section 4, ideas are presented that can help deal with missing data, and Section 5 discusses how Principal Component Analysis can be used to support traffic index calculations. Finally, in Section 6 conclusions are presented.

2 Graph construction

The idea to reconstruct the road network has been devised to utilize the maximum information provided by the data. Constructing a graph for the full road network would require more than just the road sensor data, e.g., OpenStreetMap data. Here, we only construct a separate directed graph for each road that has sensor data. This means that roads without sensors, including all connecting roads with exits from and entrances to roads with sensors, are ignored.

For the purpose of calculating a traffic index, we feel that this is the best way to deal with these 'dark roads'. First, there is no way to tell what exactly happened. How many cars exited and entered a measured road between sensors A and B? What dark road did they go on or come from, and did they drive that entire road or were they just visiting the closest house along that road? Second, assuming that the number of cars on dark roads is, in approximation, a constant fraction of the total number of cars, missing data of these roads should not significantly impact the relative traffic index.

We can roughly split the construction of the graph for a single road into two parts: a) determining most likely neighbours of sensors, and b) determining an order of the sensors that a car traveling on a road follows. We will need the following three ingredients:

1. coordinates of each sensor in latitude and longitude,
2. the name of the road each sensor is on,
3. the direction of the recorded traffic at each sensor.

The latitude and longitude coordinates of a sensor are used to locate the sensor on the map, and to compute the distance between two sensors. If the earth is assumed to be a perfect sphere, then the shortest distance between two points on the sphere is the smaller arc length of the great circle passing through these two points on the sphere. This is called the *geodesic* distance, and can be calculated from:

$$\text{haversin}\left(\frac{d_{AB}}{R}\right) = \text{haversin}(\phi_B - \phi_A) + \cos\phi_A \cos\phi_B \,\text{haversin}(\lambda_B - \lambda_A), \tag{1}$$

where d_{AB} is the geodesic distance between points $A(\phi_A, \lambda_A)$ and $B(\phi_B, \lambda_B)$ on a sphere of radius R, ϕ_A and ϕ_B are the latitudes, λ_A and λ_B the longitudes, for A and B, respectively, and

$$\text{haversin}(\theta) = \sin^2\frac{\theta}{2} = \frac{1-\cos\theta}{2}. \tag{2}$$

Note that MATLAB has a built-in geodesic distance function. The region of South Limburg has latitudes between 50.75° N and 51.05° N and longitudes between 5.7° E and 6.1° E.

Using the name of the road each sensor is on, the sensors can be assigned to their respective roads, and using the direction data the lanes in opposite directions can be separated. The main aim is to find the *successor* sensor for each sensor when travelling in a certain direction, as our car traffic calculation method requires that we know the order in which the sensors are traversed. For this we employ a modified nearest-neighbour algorithm, processing one direction of one road at a time.

When we have the order of the sensors for one direction of a road, we create a *pseudo-connection matrix* P of size $n \times n$, with a row and column for each of the n sensors (the name will be explained later on). For each sensor A in the data, we first find the sensor B that is closest to A. In the matrix P, there will then be a 1 in the (B, A)-entry.

Except for the first and last sensor on the road, each sensor has two neighbours. Rather than just looking for the sensor that is closest to A after B, we apply a second criterion to make the result more realistic, i.e., we only consider sensor C where the angle between AC and AB is at least 90 degrees. We illustrate this criterion with an example. In Figure 2 the closest neighbour to A is B, and the second closest neighbour is C. Still, because we assume that roads do not make turns sharper than 90 degrees, we consider D the more likely second neighbour of A, as in Figure 3.

We apply the rules for first and second neighbours to all of the sensors involved, and fill the matrix P accordingly. This matrix will only have 0s

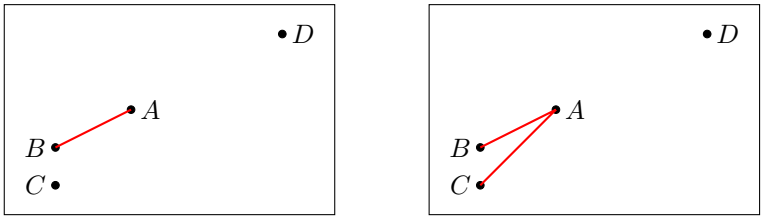


Figure 2: Neighbours of A when not using the 90 degrees criterion.

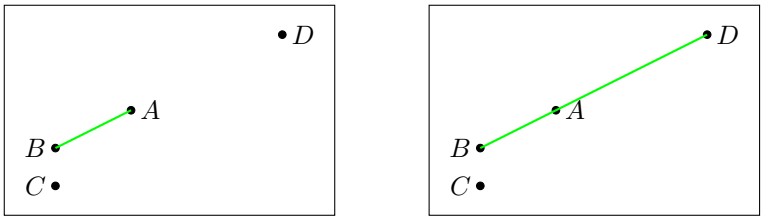


Figure 3: Neighbours of A when using the 90 degrees criterion.

and 1s, with at most two 1s per column. The reason we call this a pseudo-connection matrix is that it is not the standard connection (adjacency) matrix for a directed graph; it also need not be symmetric, as the connection matrix of an undirected graph would be.

The next step is to use the direction of the road to make sure we start at the beginning of the road and then find the successor sensors. To determine the starting point we simply look at the raw coordinates, in combination with the direction, e.g., if a road is eastbound we start at the westmost point. This first sensor, with index o_1, should only have a single connection to other nodes, i.e., the o_1-th column of P has a single entry at some row o_2. Thus, we have found the second sensor on the road. We then keep going by looking at the o_2-th column, which should have 2 nonzero entries, one at the o_1-th row and one at some row o_3. If the sensors are placed nicely on a straight road, we can continue this way until the end of the road (see Figure 4).

However, when the road is not straight this method may sometimes go awry, as illustrated in Figure 5. Starting at B we go to the successor A. There we have a problem, since A connects to both C and D. To fix problems as these we choose whichever point has the largest distance to A, skipping the other point. We may lose some sensors in the process but this will not invalidate the traffic estimation; it only reduces the accuracy of the approximation (see Section 3)

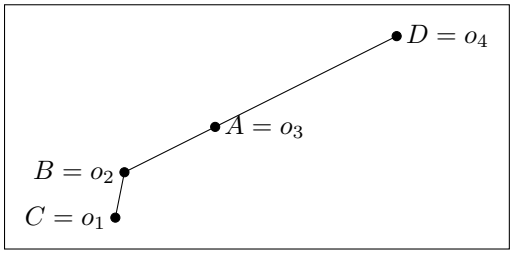


Figure 4: Eastbound road constructed from sensor coordinates.

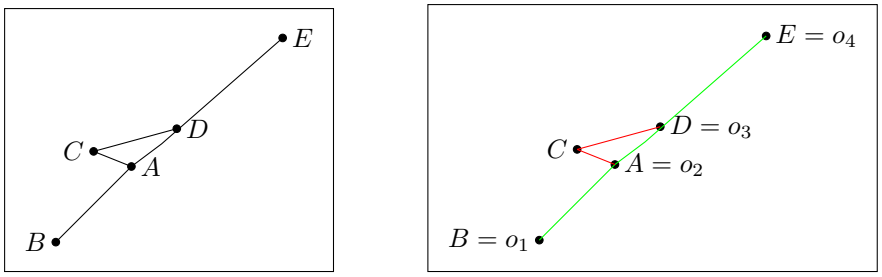


Figure 5: A problematic graph and the used solution.

Another method to construct the graph for a single road, is to assume that the shortest path that connects all sensors is the correct way to connect them. This graph can be calculated by adding a dummy node with distance 0 to all other nodes, and then solving the Traveling Salesman Problem (TSP).

Solving a TSP for each road is doable as long as the number of sensors on a single road is not too high, but using the knowledge that a road should be mostly straight can also inspire a number of heuristics that are quick to calculate. For instance, a linear regression line through the sensors can be calculated, and then sensors can be ordered by projecting them onto that line. If needed, simple local search heuristics can be used to improve on the initial solution.

3 Calculating traffic

Given a directed graph, where the nodes represent sensors and the edges represent the road segments between these sensors, the intensity and velocity measurements, in conjunction with the segment lengths, can be used to estimate traffic quantities. Here, we discuss estimating the number of cars on the road at a given time, and the number of kilometers driven by all cars on the road during a certain time period.

3.1 Car count

To estimate the number of cars on the road at a given time, we estimate the number of cars on each road segment. In the simplest case, we assume cars to be driving at constant speed for some time around the measuring sensor. The estimated number of cars on the road segment AB, based on the measurements at sensor A and sensor B, respectively, equals

$$C_A(t) = I_A(t)\frac{d_{AB}}{v_A(t)}, \tag{3}$$

$$C_B(t) = I_B(t)\frac{d_{AB}}{v_B(t)}. \tag{4}$$

Note that to be correct, the estimates $C_A(t)$ and $C_B(t)$ need the average velocities $v_A(t)$ and $v_B(t)$ to be the harmonic mean of the velocities of the passing cars. Unfortunately, only the arithmetic mean is available in the data. This issue is further discussed in Section 4.

The estimate $C_A(t)$ assumes all cars that enter the road segment at sensor A to drive along the entire segment. It does not take into account cars leaving the road somewhere along the segment AB. Similarly, $C_B(t)$ does not take into account that cars may have entered somewhere along the segment.

Assuming that leaving and entering occurs halfway along the road segment, these two effects can be incorporated in the car count by averaging $C_A(t)$ and $C_B(t)$:

$$C_{AB}(t) = \frac{C_A(t) + C_B(t)}{2} = \frac{d_{AB}}{2}\left(\frac{I_A(t)}{v_A(t)} + \frac{I_B(t)}{v_B(t)}\right). \tag{5}$$

Note that this formulation is independent of how many cars leave and enter the road on segment AB. All that matters is the difference between the number of cars that left and that entered.

For short road segments AB, specifically if $\frac{d_{AB}}{v_A(t)} < T$, equation (5) should give a good estimate for the number of cars on the segment. For longer road segments, the constant speed assumption may be too much of a simplication. This problem can be alleviated by using measurements of multiple time intervals. We still assume the cars to traverse the road segment with constant speed as measured at the sensor, but the different speed of cars that arrive within different time intervals will be taken into account.

In the time interval $[t + kT, t + (k + 1)T)$, cars are measured at a sensor with average speed $v(t + kT)$. Relative to the placement of that sensor, at time t these cars are expected to be at $(-(k + 1)Tv(t + kT), -kTv(t + kT)]$.

At sensor A only measurements before t are interesting, i.e., $k < 0$, because measurements after t correspond to cars that had not entered the segment AB yet at time t. For $k < 0$, if

$$-kTv_A(t + kT) \leq d_{AB} \Leftrightarrow \frac{d_{AB}}{Tv_A(t + kT)} + k + 1 \geq 1 \tag{6}$$

then all the measured cars are within the segment AB at time t, while if

$$-(k + 1)Tv_A(t + kT) \geq d_{AB} \Leftrightarrow \frac{d_{AB}}{Tv_A(t + kT)} + k + 1 \leq 0 \tag{7}$$

then all the cars already passed the segment at time t. In all other cases, assuming a uniform distribution of the cars within the measured time, the fraction of the measured cars that are in segment AB at time t is equal to

$$\frac{d_{AB} - [-(k + 1)Tv_A(t + kT)]}{Tv_A(t + kT)} = \frac{d_{AB}}{Tv_A(t + kT)} + k + 1. \tag{8}$$

The number of cars on the road segment AB can then be estimated from the measurements in sensor A by adding contributions for all $k < 0$,

$$C_A(t) = T \sum_{k=-N}^{-1} I_A(t + kT) \max\left\{0,\ \min\left\{\frac{d_{AB}}{Tv_A(t + kT)} + k + 1,\ 1\right\}\right\}. \tag{9}$$

Similarly, the number of cars on segment AB can be estimated from the measurements in sensor B by

$$C_B(t) = T \sum_{k=0}^{N-1} I_B(t+kT) \max\left\{0,\ \min\left\{\frac{d_{AB}}{T v_B(t+kT)} - k,\ 1\right\}\right\}. \quad (10)$$

Here, the truncation value N should be such that no contributing measurements are neglected. A safe value for N can quickly be calculated from

$$N = \frac{d_{AB}}{T \min_{t,\xi\in\{A,B\}} v_\xi(t)}. \quad (11)$$

Again averaging to account for cars leaving and entering the road along the segment AB, we get

$$C_{AB}(t) = \frac{T}{2}\left(\sum_{k=-N}^{-1} I_A(t+kT) \max\left\{0,\ \min\left\{\frac{d_{AB}}{T v_A(t+kT)} + k + 1,\ 1\right\}\right\}\right.$$
$$\left. + \sum_{k=0}^{N-1} I_B(t+kT)\ \max\left\{0,\ \min\left\{\frac{d_{AB}}{T v_B(t+kT)} - k,\ 1\right\}\right\}\right). \quad (12)$$

Note that the treated methods for counting traffic are essentially independent of the number and placement of the sensors. That is, if the simplifications assumed to model the traffic on a road segment would be exact, then using measurements from any set of sensors would lead to the same traffic count, provided that the same part of the roads is covered. Evidently, using more sensors that are closer together does lead to more reliable estimates.

Many more extensions are possible to better estimate the number of cars on a road segment at a given time. For instance, using an interpolated function for intensity and velocity, or incorporating a typical distribution of velocities for a certain road. However, the intended use of these statistics is a traffic index, i.e., an aggregation over a time period and a geographical area. In this case, the provided estimates are expected to be accurate enough, as the local approximation effects should not significantly impact the aggregate.

Figures 6–8 show the estimated car count on two major roads in South Limburg, based on equation (5), using the minute data of the road sensors. Figure 6 shows the estimated traffic on the A76 on a Friday. The morning and afternoon rush hours are clearly visible. There is slightly more westbound traffic in the morning, and more eastbound traffic in the afternoon. Figure 7 shows the traffic on the A2 on a Friday. Again, the morning and afternoon rush hours are clearly visible. Further, it is clear that there is a lot more southbound traffic, towards the city of Maastricht, the entire day. Figure 8 shows the traffic on the A2 on a Saturday. There is still a lot more southbound traffic, but there is no morning or afternoon rush hour.

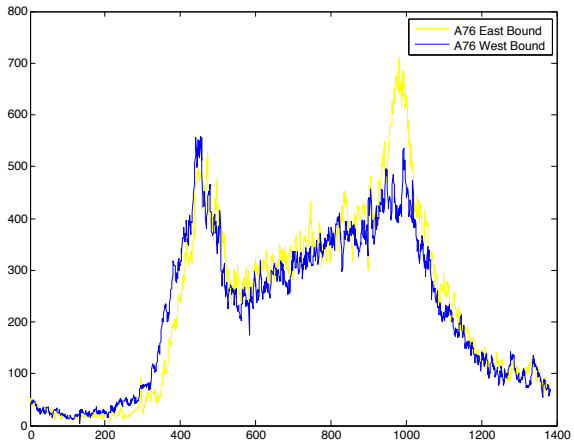


Figure 6: Car count on the A76 on a Friday (Feb. 1, 2013).
Horizontal: minutes past midnight. Vertical: number of cars on the road.

3.2 Car-kilometer count

We want to build a traffic index as an indicator of the traffic usage on the infrastructure. One way of measuring this is the average number of cars on the road, as described in the previous section. Another way would be the total usage of the roads by car-kilometers. In this case, one is not only interested in the current number of cars but also how far they travel. This is measured by the total number of car-kilometers, i.e., if there is a way of recording all cars and their whereabouts, the sum of kilometers of all car trips on the road network. The car-kilometer approach is studied in this section.

Imagine an abstract straight road $[0, A]$. The road sensors (observation points) are located at positions $x_1, x_2, \ldots, x_K$ (assume $0 = x_0 < x_1 < x_2 < \cdots < x_K < x_{K+1} = A$). Take $\rho(x, t)$ as the car density at location x at time t, such that the total number of car-kilometers on the road, in the time period $[0, \tau]$, is

$$K(\tau) = \int_0^\tau \int_0^A \rho(x, t) dx dt. \tag{13}$$

We do not know $\rho(x, t)$ at every x, but we have traffic records at the sensors x_i. Thus, we can approximate $\rho(x, t)$ with piecewise constant functions using our observations $\rho(x_i, t)$, and then come up with an approximation of K. From a numerical integral theory viewpoint, the best method is to divide the intervals to construct the piecewise function as follows. Given $\rho(x_i, t)$, we

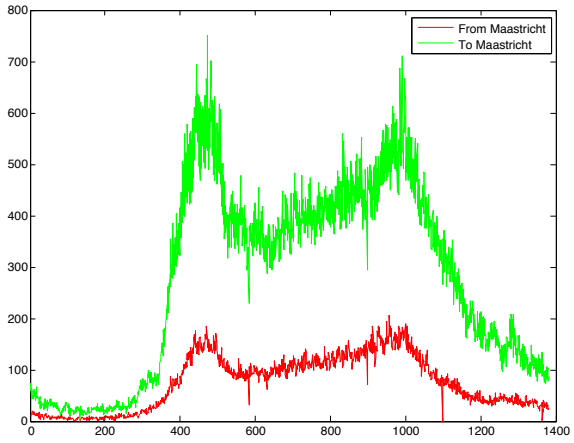


Figure 7: Car count on the A2 on a Friday (Feb. 1, 2013).
Horizontal: minutes past midnight. Vertical: number of cars on the road.

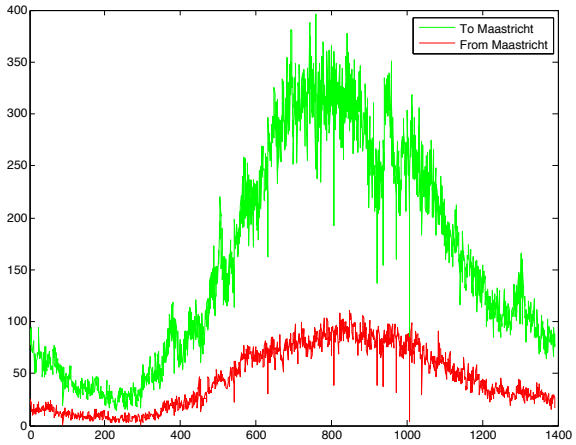


Figure 8: Car count on the A2 on a Saturday (Feb. 2, 2013).
Horizontal: minutes past midnight. Vertical: number of cars on the road.

approximate $\rho(x,t)$ by

$$\rho(x,t) \approx \sum_{i=1}^{K-1} \mathbb{1}_{\left((x_{i-1}+x_i)/2,(x_i+x_{i+1})/2\right]}(x)\rho(x_i,t). \tag{14}$$

The values $\rho(x_i,t)$ can be approximated by the observations at sensor x_i. We then arrive at the following approximation of K:

$$K(\tau) \approx \sum_i d_i N_i, \tag{15}$$

where N_i is the number of cars passing sensor x_i in the time interval $[0,\tau)$, and d_i is the length of the interval around x_i, i.e., for three consecutive sensors x_{i-1}, x_i, and x_{i+1} take $d_i = \frac{1}{2}(\|x_{i+1} - x_i\| + \|x_i - x_{i-1}\|)$. Thus, d_i is the estimated travel distance of a car captured by sensor i, covering half of the interval before sensor i and half the interval after.

In terms of the provided data, an approximation $\mathrm{CK}(t)$ of the car-kilometer count in the time interval $[t, t+T)$ can be obtained by

$$\mathrm{CK}(t) = \sum_i d_i I_i(t) T, \tag{16}$$

where $I_i(t)T$ is the number of cars passing sensor x_i in the time interval $[t, t+T)$, and d_i is a length that represents the part of the road around x_i, as in equation (15).

We developed the above method for an abstract straight road. However, the principle works for any topography. We just have to associate the correct length to sensors. Naturally these lengths are difficult to deal with, as they need detailed information on the geography of roads. However, this approach has some nice properties. First, it does not care whether a car is recorded by several sensors. If a car is recorded twice, it means the car travels more kilometers, which means more damage to the roads. (It could also mean more air pollution.) Second, this approach does not mind too many sensors, as it is cheap to compute, and the more sensors the more accurate the approximation will be.

The car-kilometer count is a fairer indicator of infrastructure usage than the car count. It represents how widely and extensively the infrastructure has been used for a certain time period. It is simple to calculate and easy to implement, and robust to using more or fewer sensors.

4 Data reconstruction

In the previous sections, a detailed method for the calculation of a traffic index has been proposed and formulated. Although theoretical considerations cover

siteID	date	location	roadNo	lane	direction	flow	speed
...				...			

Table 1: Excerpt from the sensor data.

some of the practical problems, there are still several issues with the input data that need detailed analysis. In this section, the structure and characteristics of actual data are discussed, the main problems are identified, and solutions and corrections are proposed.

4.1 Data format and data problems

The data provided by CBS covers both 1 minute and day measurements. The former are restricted to the region of South Limburg for two consecutive days: Friday, 1 Feb. 2013 and Saturday, 2 Feb. 2013, and include 2 million observations from 424 road sensors. The latter include observations from all of the Netherlands for a period of 3 months, and include 900 000 observations from 15144 sensors. The available variables describe a wide range of different features, including flow and velocity measurements, directional information, and location coordinates.

Table 1 presents an excerpt of the data that covers the most important variables from the point of view of the method proposed in previous sections.

As the traffic system is constantly changing, sensitive to unpredictable phenomena, and vulnerable to different factors, one must also expect possible errors in the traffic data. Fortunately, the system is designed to return "−1" if there was an error. This allows to differentiate between no traffic (zero measurements) and malfunctioning equipment.

A preliminary analysis of the data shows several obstacles for the implementation of the proposed solution. First, we found that there is a problem with the availability of directional data of sensors. Second, we identified also missing and incorrect intensity and velocity data. Finally, the velocity averaging method needs some investigation.

Brief comments regarding all the variables are summarized in Table 2.

4.2 Missing data

The first step of our solution is to create a graph using the data on the location and direction of the sensors. Therefore, it is important to be able to match every sensor to an exact geographical point. Figure 9 shows all sensors

Variable	Comment	Problems
siteID	Unique number for each sensor	No problems
date	Time of measurements	No problems
location	Geographical coordinates	No problems
roadNo	Exact road number	No problems
lane	Number of the lane	No problems
direction	Direction of the sensor	Many missing entries
flow	Measured intensity	Both errors (-1) and missing data (unexpected 0)
speed	Measured velocity	Averaging: arithmetic mean Both errors (-1) and missing data (unexpected 0)

Table 2: Summary of variables and problems in the data.

in the Netherlands and in South Limburg, drawn according to the provided longitude and latitude information.

As the presented figures illustrate, the sensors represent a network of the main roads in the Netherlands, which is compatible with our proposed approach. However, if we were to use only sensors for which directional data is available, we would have to omit a considerable part of the Netherlands. About 30% of the sensors lack information on the direction, see Figure 10. Hence, having a systematic procedure to reconstruct the missing data is vital, in order to provide a reliable calculation of the traffic index.

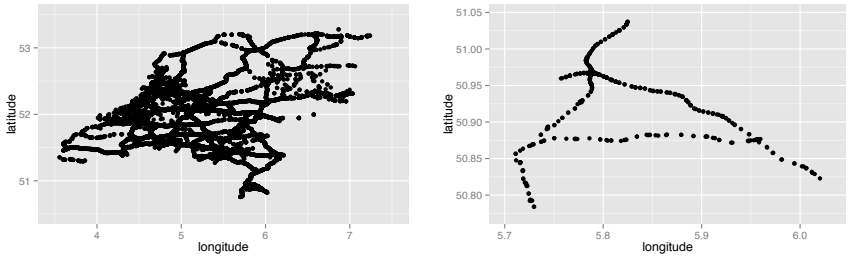


Figure 9: Sensors in the Netherlands (left) and South Limburg (right).

The second problem with the data concerns missing and incorrect measurements of intensity and velocity of traffic. We distinguish between two possible errors: 'blind sensors' and 'blind time points'. The first indicate the situation where a sensor does not give any data for any period of time, whereas the second represents holes in the time series data for a given sensor. Table 3 shows the percentages of these two types of errors in the provided data.

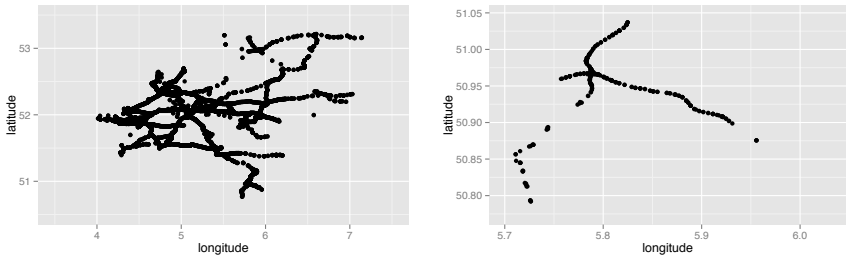


Figure 10: Sensor with available direction data in the Netherlands (left) and South Limburg (right).

	Velocity	Intensity
NL	6.5	4.5
SL	10.5	8.0

	Velocity	Intensity
NL	2.5	2.0
SL	15.5	0.0

Table 3: Percentage of blind sensors (left) and blind time points (right), for the Netherlands (NL) and South Limburg (SL)

The percentages of errors are tolerable, as our proposed solution does not need to be applied using all possible measurements. Problematic sensors can be excluded with a little loss of precision.

The third identified problem concerning the data lies in the method of averaging velocity measurements. For our purpose the correct way to compute an average of speed is to use the harmonic mean of the observations. Unfortunately, all the provided data give are the value of the arithmetic mean.

4.3 Direction reconstruction

The main problem with the data is the lack of directional information in a substantial number of sensors. We propose to retrieve this data on the basis of the behaviour of the traffic intensity. We hypothesized that, as far as different directions of one road are concerned, patterns of traffic should differ within a weekday. For example, in the morning rush hours the traffic should converge mostly to the biggest city in the neighbourhood, whereas during the afternoon the profile should be the opposite. The same is true in the case of a significant event in a specific place.

From the mathematical point of view, we would like to compare a set of time series data with each other and then cluster the sensors according to some distance function for a pair of sensors. We identified two approaches, which can be applied to this problem:

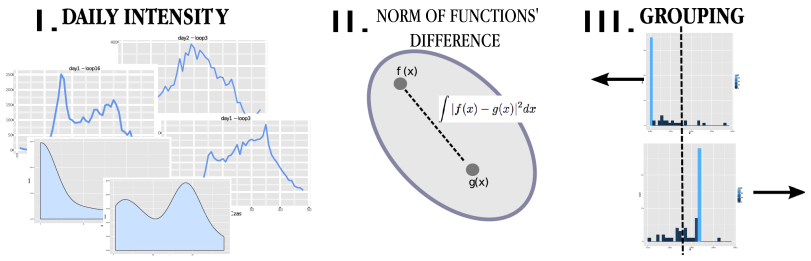
1. Compare a whole-day time series for each sensor. One simple method is just to use a mean square error as a distance function. This will not take into account the possible time dependence (time lag) between consecutive sensors. Hence, we propose to distinguish them using a cross-covariance (cross-correlation) function, which compares two time series and their lagged transforms[1].

2. Calculate the distribution function of the intensity for each sensor in one, chosen rush hour, i.e., either in the morning or in the evening. In order to estimate the distribution, one can use histograms or kernel density estimators. In the next step, the distance function would be the difference between two histograms/densities, which can be calculated using Kullback-Leibler divergence or just the mean square error.

The method which compares only distributions of intensities is much simpler to apply and cheaper to implement, but it aggregates and loses information included in the time series. Therefore, it has a potentially smaller range of applicability. Nonetheless, if the hypothesis stated above is true the distinction made in that way should be reliable.

[1] In the statistical software package R, this is implemented via the function *ccf*.

The presented idea can be summarized in the following picture:



The steps for method 1 are:

a. Identify continuous approximation of intensities.

b. Compute how the time series differ between sensors.

c. Divide sensors into two groups according to the distance.

For method 2, the steps are:

a. Identify distribution of intensities in the rush hour.

b. Compute distance between densities (Kullback-Leibler).

c. Divide sensors into two groups according to the distance.

We believe that such an approach should be useful, when there is strong evidence that the traffic is very different in two directions. Unfortunately, as the provided dataset for South Limburg shows, it is not a general rule that always holds. In particular, such an approach was not successful enough in differentiating the direction of sensors for roads A2 and A76, where we know most directions. We cannot assume then that the approach can be applied for the road A79 in South Limburg. In that case, we must in addition make use of information on the location of the sensors. Therefore we propose the following procedure, which is a modification of the initial proposition.

Let us assume we are given a set S of sensors within a single road without direction. Sets S_1 and S_2 will include sensors in different directions.

1. Choose an arbitrary sensor $s_0 \in S$, add it to S_1 and delete it from S.

2. Choose n (we propose $n = 4$) different sensors $s_1, \ldots, s_n \in S$ that are the nearest (in the sense of location) to s_0, but not farther than a fixed tolerance distance d (we propose $d = 3$km). If it is not possible, go to step 1 (with different s_0).

3. Calculate the cross covariance for the intensity time series data between s_0 and each sensor from the set $\{s_1, \ldots, s_n\}$. The number of calculated lags (mostly 1, 2 or 3) should be chosen according to the average velocities and distance between sensors; car-count formulas from section 3.1 can be used here.

4. Choose one sensor, s_h, with the highest covariance. Add s_h to S_1 and delete it from S.

5. Choose one sensor, s_l, with the lowest covariance. Add s_l to S_2 and delete it from S.

6. If S is empty, stop. Otherwise, go to step 2 and repeat on the current S with $s_0 := s_h$.

The method will be most efficient, if the first chosen sensor is approximately in the middle of the set S in the sense of location. Moreover, the algorithm will work better, if sensors are uniformly distributed in both directions, i.e., the number of sensors in each direction is similar. The more the number of sensors in each direction differs, the higher the choice of the variable n should be. The procedure needs to be carried out only once, but should be tested on a regular basis with newly available data.

Using this method, the identification of the directions for roads A2 and A76 in South Limburg was at a level of 80%. We apply the method to the data for A79, a reconstruction of which is presented in Figure 11.

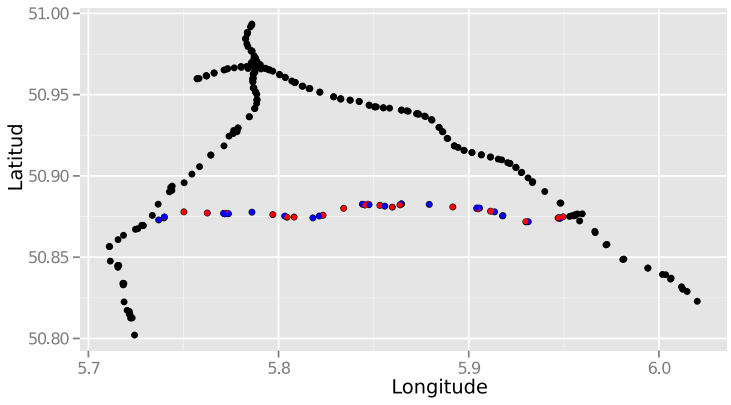


Figure 11: Reconstructed directions (red/blue) for the A79 in South Limburg.

4.4 Velocity averaging

Another issue that needs special attention, is that the velocity is given as an arithmetic mean instead of the harmonic mean. The best method to correct this problem is of course to get the harmonic mean from the source of the measurements. If this is impossible, however, the suggestions below can be used to improve the calculations.

An elementary inequality between means states that for positive $x_1, \ldots, x_n$

$$\frac{n}{\frac{1}{x_1} + \ldots + \frac{1}{x_n}} \leq \frac{x_1 + \ldots + x_n}{n}, \tag{17}$$

i.e., the harmonic mean is less than or equal to the arithmetic mean.

In the case of the calculation of a traffic index, this means that using an arithmetic mean of observations leads to an upper bound on the true value. Unfortunately, there is no method to quantify the error. The difference between the arithmetic and harmonic mean can attain any value depending on the dispersion of observations. The higher the difference between the observation values, the larger the error of using the arithmetic mean.

We see the following options to achieve more accurate calculations:

- aggregate different types of vehicles into one group using a harmonic mean for the velocity,
- aggregate minute measurements into longer periods using a harmonic mean for the velocity,
- aggregate observations with respect to location, using a harmonic mean for the velocity.

All these proposals give better approximations to the traffic index, but at the same time they sacrifice some details of the data, e.g., information on different types of vehicles. This approach to correcting the dataset will be effective especially in the case of low-traffic periods. Moreover, the proposed solution in previous sections implictly assumes harmonic averaging with respect to location, which already alleviates the problem somewhat.

A more involving method could be devised, if a typical distribution of vehicle speeds on the road is available. The arithmetic mean could then be used to select a likely set of velocities that realised that mean, and from this set the harmonic mean could be calculated. The accuracy of such a method would rely heavily on the predictability of the vehicle speeds.

5 Principal Component Analysis (PCA)

The procedure described in the previous sections gives an approximation of the number of cars on the roads in a specific time interval using the minute-by-minute information collected by the sensors. However, this information could not be available in certain scenarios where the collected dataset is reduced to the number of cars detected by the sensors in one hour or one day.

In this case, the data show a main feature: they contain much redundant information because one car can be detected by several sensors. In matrix terms, it means that if we assume that the information of each sensor is in the columns of a matrix A, then the dimension of the linear subspace spanned by the columns of A is much smaller than the number of sensors.

In this case, describing the data using an orthonormal basis of this 'smaller' subspace that contains a compressed representation of A, is a good strategy. The Singular Value Decomposition (SVD, Golub and Van Loan (2012)) of A permits to calculate a possible set of basis vectors formed by the singular vectors corresponding to the largest singular values of the matrix A.

This fundamental data analysis tool, known as Principal Components Analysis (PCA), is one of the methods investigated by the CBS specialists to know the variability of the data and calculate a traffic index.

Some recent approaches can improve the application of PCA on the traffic data:

- Robust PCA, Ke and Kanade (2005): Techniques to recover the low-rank matrix approximations from highly corrupted and/or missing measurements.
- CUR matrix decomposition, Mahoney and Drineas (2009): Here, the basis vectors are explicitly expressed in terms of a small number of actual columns and/or actual rows of the data matrix. In the traffic index context, the application of the CUR matrix decomposition could detect sets of sensors where the redundance in the information is minimized. In contrasts with the known limitation of the classic PCA, with CUR one can interpret the produced basis in terms of the original data. An additional advantage of PCA based on CUR decomposition is avoiding the calculation of the Singular Value Decomposition, giving the possibility of working with a much larger dataset.

A final observation: the calculation of a traffic index using a PCA version on the hourly or daily data does not use the information of the velocity of the cars on the road. This can be an advantage considering the high proportion of missing data for this parameter.

6 Conclusions

In this paper, the problem of calculating a measure of the amount of traffic on the roads, based on data from road sensors, has been treated.

A method was presented for constructing a graph, based on the sensor data, with the sensors as nodes and the road segments between sensors as edges. Using such a graph, we presented methods for estimating the number of cars, and the number of car-kilometers. Both estimates are more accurate when more sensors are used, but are otherwise independent of the number and placement of sensors used, provided that the same set of roads is covered by the sensors.

Further, procedures were proposed that deal with missing data, focussing on reconstructing the traffic direction for sensors that missed the direction data field, and on dealing with the arithmetic velocity mean being given, when our methods rely on the harmonic velocity mean.

Finally, the uses of principal component analysis for this particular problem were discussed.

References

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, MD, fourth edition, 2012.

Q. Ke and T. Kanade. Robust l^1-norm factorization in the presence of outliers and missing data. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *PNAS*, 106(3):697–702, 2009.