

Real-time schattingen van voertuigaccumulatie met FCD

In ruwe vorm zijn *floating car data* rijk en uiterst gedetailleerd. Het punt is alleen dat we om privacyredenen niet met die ruwe data mogen werken – we moeten het doen met *geaggregeerde* data die door een *privacyfilter* zijn gehaald. Is daarmee nog enige verkeersmanagement-er te behalen? Jawel, mits je de FCD slim combineert met bijvoorbeeld *lusdata*. Zelfs het real-time schatten van wachtrijen behoort dan tot de mogelijkheden, aldus de auteurs.

Elk datatype heeft z'n sterke punten, maar ook z'n zwakke punten. Die laatste beperken de gebruiksmogelijkheden van de betreffende data – en vandaar ook dat elk datatype specifieke, vastomlijnde toepassingen heeft. Dankzij datafusie is het echter mogelijk om twee 'beperkte' datatypes zo te combineren dat zich nieuwe toepassingen aandienen.

In deze bijdrage zoomen we in op één interessant voorbeeld van datafusie: het combineren van *floating car data* (FCD) en *lusdata* om het aantal voertuigen op een wegvak in real-time te schatten. Die gebruikstoepassing speelt een belangrijke rol in het bepalen van de overgebleven ruimte op buffers, bijvoorbeeld van een toerit met een toeritdoseerinstallatie. Hoe nauwkeuriger die ruimte wordt geschat, hoe gericht en veiliger we de toeritdosering kunnen inzetten.

Wachtrijwachter op basis van instroom en uitstroom

Er zijn verschillende manieren om een wachtrij op een wegvak (buffer) te beschrijven. Wij hebben er in onze studie voor gekozen om het aantal voertuigen op het wegvak, de voertuigaccumulatie, als uitgangspunt te nemen. Die accumulatie schatten we aan de van de in- en uitstroom op het betreffende wegvak. Die meten we met lussen aan het begin en einde van het wegvak. Het verschil tussen de instroom A en uitstroom D in periode p , oftewel $A(p) - D(p)$, geeft de verandering in voertuigaccumulatie.

Als we weten hoeveel voertuigen zich op een zeker moment in het wegvak bevinden (een mooi vertrekpunt is als het wegvak leeg is) en we de instroom en uitstroom observeren met lussen, dan kunnen we op elk tijdstip het aantal voertuigen op het wegvak bepalen.

Tenminste, in theorie. In praktijk werkt deze methode namelijk niet goed: er kunnen zich altijd meetfouten voordoen, bijvoorbeeld als een voertuig net van strook verandert als het een van de lussen passeert. Zulke meetfouten stappelen zich over de tijd op, waardoor de schattingsfout erg groot kan worden: het *cumulative error problem*, het cumulatieve-foutenprobleem.

Beperking van het cumulatieve-foutenprobleem

Het cumulatieve-foutenprobleem is onderwerp van meerdere academische studies. Door gebruik te maken van andere data, zoals *floating car data*, kan de fout op gezette tijden worden hersteld, zo toonden Ashish Bhaskar e.a. (2011) en Hans van Lint e.a. (2015) aan.^{*} Daarnaast kunnen we aan de hand van geobserveerde fouten de structurele meetfouten leren doorgronden, aldus Zahra Amini e.a. (2016).^{**}

Het probleem van de meeste bestaande studies is echter, dat er gewerkt wordt met *ruwe* FCD, trajectoriedata. Met dit type data kun je afzonderlijke voertuigen volgen over de tijd – en uit privacyoverwegingen zal een wegbeheerder nooit met deze data mogen werken. Waar wegbeheerders en dienstverleners hooguit over beschikken zijn *floating car data* die zijn *geaggregeerd* over tijd en ruimte. Vaak is er ook nog een *privacyfilter* overheen gegaan. Denk dan aan data uit het Google Better Cities-programma en aan FCD zoals NDW die levert. Is er ook een (datafusie)oplossing denkbaar die geschikt is voor dit type 'bewerkte' FCD?

De methode

De methode voorgesteld in ons onderzoek aan de TU Delft combineert *lusdata* en snelheidsdata op basis van *geaggregeerde* FCD. Door de relatief lange aggregatieperiode die deze FCD hebben – voor de Google Better Cities-data bedraagt die periode maar liefst 5 minuten – zijn de data op zichzelf niet geschikt voor real-time schattingen van de voertuigaccumulatie. We kunnen deze data echter wel gebruiken om het cumulatieve-foutenprobleem te beperken.

Dat doen we op twee manieren. Zodra de *geaggregeerde* FCD beschikbaar zijn, gebruiken we die om (1) de schattingsfout te herstellen en (2) onze schatting van de structurele meetfout van de lussen te updaten. De methode is volledig beschreven in een wetenschappelijk artikel dat op korte termijn gepubliceerd wordt.^{***} In het kort: De notatie $M(p)$ wordt gebruikt om de voertuigaccumulatie aan het einde van periode

* Het gaat om de studies 'Fusing Loop Detector and Probe Vehicle Data to Estimate Travel Time Statistics on Signalized Urban Networks' (Bhaskar, 2011) en 'A generic methodology to estimate vehicle accumulation on urban arterials by fusing vehicle counts and travel times' (Van Lint, 2015).

** Zie 'Queue-Length Estimation Using Real-Time Traffic Data' (Amini, 2016), 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC).



p te beschrijven. Deze schatten we in real-time met behulp van de volgende recursieve vergelijking:

$$M(p) = \max[M(p-1) + A(p) - D(p) + bD(p), 0]$$

De wegvak in- en uitstroom ($A(p)$ en $D(p)$) observeren we met behulp van lussen en $M(p-1)$ hebben we in de voorgaande periode geschat. Deze vergelijking houdt er rekening mee dat het fysiek onmogelijk is dat er minder dan nul voertuigen op het wegvak staan. Ook hebben we een term toegevoegd om de structurele meetfout te beschrijven ($bD(p)$). Hierbij gaan we ervan uit dat de structurele meetfout proportioneel is aan het aantal voertuigen dat het wegvak verlaat.

Zodra we de beschikking hebben over de geaggregeerde FCD, schatten we de gemiddelde voertuigaccumulatie voor de FCD-periode op twee manieren. Door de geaggregeerde FCD te fuseren met lusdata kunnen we een tijdgemiddelde voertuigaccumulatie gedurende de FCD-periode (bijvoorbeeld: 5 minuten) schatten. Vervolgens bepalen we de tijdgemiddelde voertuigaccumulatie aan de hand van de schattingen op basis van alléén de in- en uitstroom-lusdata, over exact dezelfde periode. Uitgaande van het verschil tussen de twee schattingen zetten we de twee bovengenoemde processen in – (1) en (2) – om zo de cumulatieve fout te beperken.

Het eerste proces dient om de fout opgebouwd in voorgaande perioden te verkleinen. Dit beïnvloedt de initiële voertuigaccumulatie ($M(0)$) die gebruikt wordt om de voertuigaccumulatie voor de volgende periodes te schatten. In het tweede proces leren we automatisch de structurele fout (biasterm b) tussen de twee lussen, die we gebruiken om de lusmetingen daarna *direct* te gebruiken voor verschillen tussen in- en uitstroom. Hierdoor verkleinen we op voorhand de fout die in een lusperiode wordt opgebouwd.

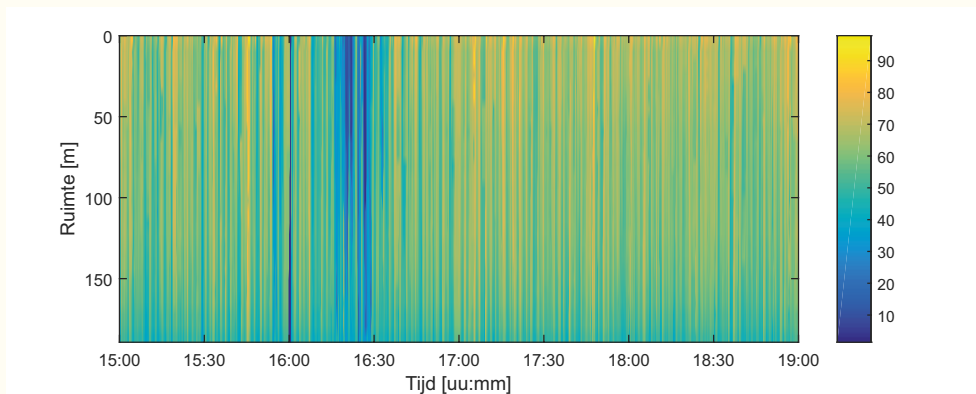
Casestudie met Google Better Cities-data

De voorgestelde methode is geëvalueerd in een casestudie die gebruikmaakt van echte data. Aangezien de methode ontwikkeld is om het cumulatieve-foutenprobleem te beperken, is het erg waardevol dat we met echte data (lees: echte fouten) kunnen werken.

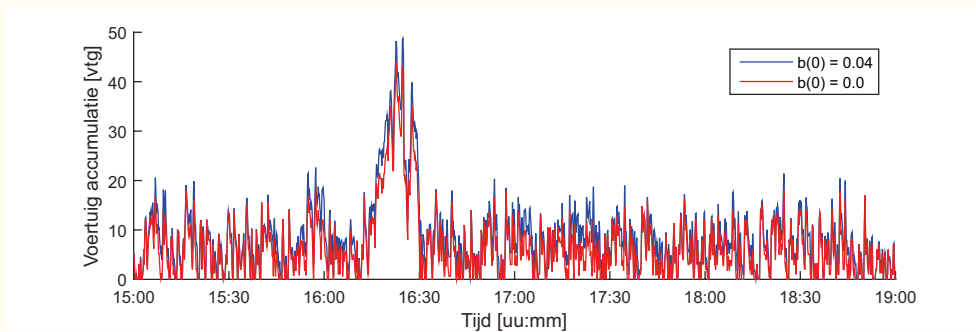
De drie datatypen die we in de case tot onze beschikking hebben zijn: (a) lusdata, (b) snelheden op basis van geaggregeerde FCD en (c) snelheden op basis van radarmetingen. Het gaat dan om data voor de toerit naar de A10-Zuid vanaf de s106 gedurende de avondspits (15:00-19:00 uur) op 6 juli 2016. Deze data zijn beschikbaar gesteld door Rijkswaterstaat en Google (als onderdeel van het Google Better Cities-programma).

De FCD worden verzameld met mobiele apparaten die gebruikmaken van Google-producten als Android of Google Maps. Er worden alleen locatiegegevens gebruikt van gebruikers die hiervoor expliciet toestemming hebben gegeven. Om de privacy te waarborgen aggregeert Google deze data (voor perioden van 5 minuten) en haalt het bedrijf ze door een privacyfilter. Bij zo'n filterbewerking wordt een random fout toegevoegd. Hoe groot die fout is hangt af van de verkeerscondities. In druk verkeer, wat voor de toepassing van onze methode het geval is, is de toegevoegde fout klein.

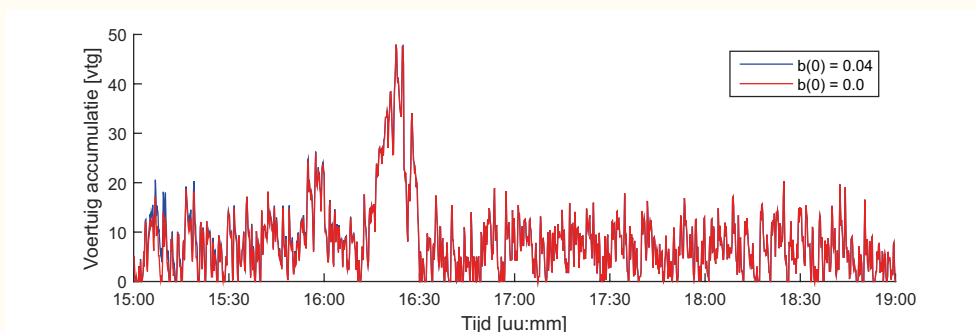
In de casestudie reconstrueren we in- en uitstroomtellingen voor perioden van 10 seconden, gebruikmakend van de lusdata (VLOG-data). Een praktisch puntje was dat de lussen aan de instroomkant van de toerit niet werkten tijdens onze avondspits van 6 juli 2016. Deze lusmetingen hebben we daarom gereconstrueerd op basis van de lusmetingen van de kruispunten stroomopwaarts van de toerit. Aan de uitstroomkant van de toerit vervalt de linker van de twee rijstroken kort na de lussen. Hierdoor zijn er een groot aantal rijstrookwisselingen op dit punt, wat resulteert in dubbele meting (meting door beide lussen) voor een enkel voertuig. Om de fout te reduceren gebruiken we alleen de metingen van de lus op de rechterstrook.

**Figuur 1:**

De snelheid op de toerit naar de A10-Zuid vanaf de s106 gedurende de avondspits op 6 juli 2016, bepaald aan de hand van radardata. De stopstreep voor de TDI is geplaatst op 0 m.

**Figuur 2:**

De schattingen van de voertuigaccumulatie op de toerit. Er is hier gebruikgemaakt van een combinatie van lusdata en Google-FCD. De FCD is gebruikt om schattingsfouten te herstellen (zonder leerproces).

**Figuur 3:**

De schattingen van de voertuigaccumulatie op de toerit. De Google-FCD is nu gebruikt om ook de structurele fouten aan te pakken (met leerproces).

Figuur 1 toont de snelheid voor de toerit gedurende de avondspits, bepaald aan de hand van radardata. Hierin is de stopstreep voor de TDI geplaatst op 0 m. In het figuur vallen twee perioden op waarin de snelheid op de toerit lager ligt, namelijk 15:55-16:00 uur en 16:15-16:30 uur, waarbij de vertragingen vooral hoog zijn in de tweede periode. Op basis van deze radardata verwachten we dat de voertuigaccumulaties in deze twee periodes het hoogst zijn.

Het verschil tussen de cumulatieve in- en uitstroom gedurende de (gehele) avondspits is -213 voertuigen ten opzichte van een totale uitstroom van 5.331 voertuigen, dus ongeveer 4%. Dit betekent dat we een hogere uitstroom hebben gezien dan instroom. Als we ervanuit gaan dat het aantal voertuigen aan het begin en einde van de avondspits gelijk is, moeten we vaststellen dat voor elk voertuig dat het wegvak verlaat er een meetfout van '0.04 vtg' is. Figuren 2 en 3 laten de schattingen van de voertuigaccumulatie zien voor de situatie waarin we de Google-FCD met de voorgestelde methode toepassen. Figuur 2 geeft de situatie weer waarin we alleen de schattingsfout herstellen en figuur 3 de situatie waarin we ook de structurele fout leren. In beide gevallen kijken we naar een (initiële) biasterm $b(0)$ van 0.0 en 0.04 – wat betekent dat we de (verwachte) structurele fout van -0.04 vtg per uitstromend voertuig compenseren.

In alle gevallen waarin de Google-data is gebruikt om de opgebouwde fouten te herstellen, zien we een duidelijke piek in de periode 16:15-16:30 uur. Dit komt overheen met de lagere snelheden geobserveerd

door de radar. Figuren 2 en 3 zeggen daarnaast iets over de waarde om de FCD te gebruiken om de structurele fout te leren en herstellen. Dit maakt de initiële waarde die we kiezen voor de bias, $b(0)$, minder cruciaal. De blauwe en rode lijn (welke een verschillende $b(0)$ hebben) convergeren snel wanneer b wordt geüpdatet met behulp van de FCD (figuur 3) terwijl er anders een verschil in de schatting blijft bestaan (figuur 2). Daarnaast is te zien dat de schattingen in figuur 3 beter de eerste periode met lagere radarsnelheden, namelijk 15:55-16:00 uur, aangeven dan de schattingen in figuur 2.

Conclusies

Dit onderzoek laat zien dat het mogelijk is om ook geaggregeerde en privacy-gefilterde floating car data te gebruiken voor real-time verkeersmanagementtoepassingen – als 'tool' om wachtrijschattingen op basis-lusdata te helpen corrigeren. De ontwikkelde methode kan worden toegepast in praktijk. Wel is het verstandig om de methode eerst verder te testen op andere locaties en andere FCD. Op basis van de bevindingen kan de methode mogelijk nog worden verbeterd en vervolgens in praktijk worden toegepast ●

De auteurs

Paul van Erp is promovendus aan de TU Delft op het gebied van het schatten van de verkeerscondities.

Dr. Victor L. Knoop is universitair hoofddocent Verkeersdynamica op de TU Delft.

Prof. dr. ir. Serge Hoogendoorn is hoogleraar Smart Urban Mobility op de TU Delft.