PEMS stations fundamental diagrams

Clark Fitzgerald   Professor Michael Zhang

January 2018

Abstract

This paper analyzes several hundred GB of highway loop detector data from the California Department of Transportation Performance Management System (PEMS). We use the data to estimate the fundamental diagram of traffic flow as a function of density and then perform clustering on the sensor locations based on a distance metric between functions. The clusters reveal a dominant group of fundamental diagrams with normal flow and a smaller group with lower flow. We demonstrate efficient computational techniques to process data both in and out of main memory.

Introduction

Traffic engineers model the flow of traffic (vehicles per hour) as a function of traffic density (vehicles per mile). This model dictates how traffic will flow in a given stretch of road, so it is known as the fundamental diagram Daganzo (1997). The motivating question for this paper is: what types of fundamental diagrams appear in empirical data sets? How can we characterize them?

At most highway locations we seldom observe high density traffic, which makes this question inherently challenging. This paper addresses the challenge directly by examining a massive amount of data and including all of it in the estimation and fitting of several different types of fundamental diagrams. The section on *Computation* explains our efficient computational approach.

Estimating the fundamental diagram in one location can be thought of as a massive data reduction, because it summarizes millions of observations into a concise functional parameterization. The details for these parameterizations can be found in the section on *Data Analysis*. Then we use the estimated fundamental diagrams as an input to a distance based clustering in the section *Clustering*.

Literature Review

The CalTrans PEMS database contains terabytes of historical traffic sensor data. Most academic analyses of the PEMS data focus on small areas for small time periods.

Li and Zhang (2011) fit a piecewise linear fundamental diagram to 30 second PEMS data by minimizing the absolute deviation from the observed data points to the fundamental diagram. This inspired the robust regression presented here.

Qu, Wang, and Zhang (2015) fit models of traffic speed as a function of density. They handle the rare high density observations with weighted least squares. Areas of density with few observations get high weights, reducing bias for various models.

Kianfar and Edara (2013) clusters individual observations of (density, flow) into congested and free flow regimes. In this paper we apply clustering techniques to the stations themselves.

Data

The size and structure of the data presented a challenge; this is why we wanted to work with it. The analysis examined the relationship between flow and occupancy for the second lane. **Flow** is the number of vehicles that pass over the detector in a 30 second period, and **occupancy** is the fraction of time that a vehicle is over the detector.

We downloaded 10 months of 30 second loop detector data in 2016 from the CalTrans Performance Measurement System (PEMS) http://pems.dot.ca.gov/ website. We chose Caltrans district 3, the San Francisco Bay Area, because this area contains many observations of high traffic activity and it's large enough to motivate the computational techniques.

Each downloaded file represents one day of observations. There are around 10 million rows and 26 columns per file that take up about 100 MB each when compressed on disk. Represented in memory as double precision floating point numbers each file will occupy about 2 GB of memory. This size becomes unwieldy with most programming languages. In total we considered 284 files with 2.6 billion rows and 26 columns for a total of 68 billion data points. This will take up 500+ GB if completely loaded into memory. This size motivated some new computational techniques.

Computation

This project required the execution of a handful of specific analytic computations on a large amount of tabular data that doesn't change. To process it efficiently we need an approach that is scalable and expressive.

The Hive database scales well as it is built on the distributed Hadoop map reduce framework Thusoo et al. (2009). However, it is difficult to express analytic computations beyond the basic ones offered through SQL (Structured Query Language). The R programming language offers the necessary statistical software and also the ability to easily express custom data analysis computations, but it doesn't scale well to this amount of data.

We combined Hive with R to make a solution that is scalable and expressive. Hive does the column selection and the group by; R performs the analytic calculation. Each group fits easily in worker memory, so the Hive workers can apply a vectorized R script to the data through POSIX `stdin`, one group at a time. This technique plays off the strengths of each system. Hive handles storage, column selection, basic filtering, sorting, fault tolerance, and parallelism. R lets us express arbitrary analytic operations through R functions.

A Hive cluster with only 4 machines was able to completely process the raw data in about 12 minutes. In contrast, a naive approach may require more complex code while also taking days to run. This speed comes from several sources. Hive can load the tables directly from Hadoop File System (HDFS) with no overhead because it uses *schema on read*. This means that it doesn't validate or transform the data when it's loaded into the database; instead it effectively references the raw text files. Hadoop's parallelism lets us fully utilize the physical hardware. R's single threaded vectorized model is reasonably efficient when combined with this parallelism.

After analyzing the data we became aware of the RHive package. The computational approach described here has less sophisticated interactive features, but is much more efficient for batch processing based on large groups, because groups are loaded in and operated on at a million elements at a time rather than line by line. An experiment showed that line by line processing would slow the program down by a factor of several hundred. Then we would be measuring run times in days rather than in minutes.

The only fundamental limit to this computational approach is that each group of data to process must fit in worker memory. This wasn't an issue here because each group only consists of around a million observations. One way to get around this is by using map reduce within R.

Data Analysis

We fit the fundamental diagram modeling vehicle flow per 30 seconds as a function of sensor occupancy. We used three different increasingly complex piecewise linear functions as shown in figure 1. This particular station is located on Interstate 80 on the Bay Bridge connecting Oakland and San Francisco, between the toll booth and Yerba Buena Island.

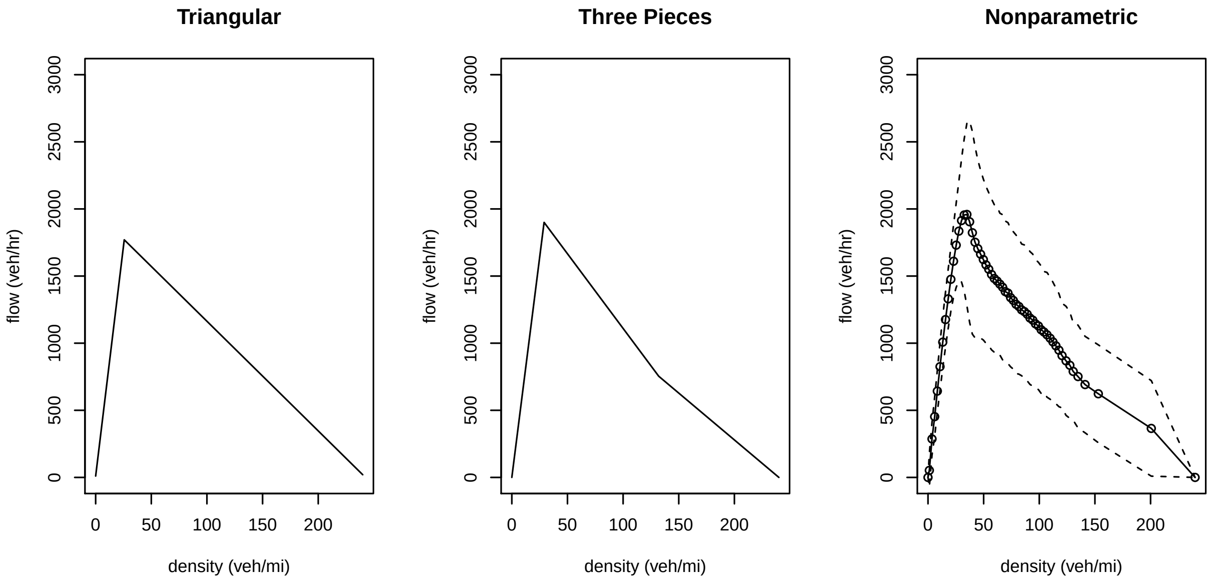

Figure 1: The models used to fit the fundamental diagrams for one typical station. The dashed lines represent the means ±2σ, where σ is the standard deviation in each bin. About 95% of the observations lie within the dashed lines.

The first method used robust regression to fit curves on the left and right hand sides of a cutoff where occupancy = 0.2. We initially chose robust regression because of its resistance to outliers. These models included an intercept, so each station is represented by two linear models, which becomes 4 floating point numbers. Including the intercept means that the fundamental diagram doesn't necessarily pass through the points (0, 0) and (1, 0). In the areas of high density many didn't pass close to (1, 0).

The second method fit three separate lines from points in different regions of occupancy:

- Left line comes from fitting in (0, 0.1)
- Center line comes from fitting in (0.2, 0.5)
- Right line comes from fitting in (0.5, 1)

We fit the lines using least squares subject to the constraints that the fundamental diagram must pass through (0, 0) and (1, 0). Enforcing this constraints makes for a more reasonable model, since we know that the fundamental diagram must satisfy this. We ignored the points in the region (0.1, 0.2) because points vary widely in this region as the traffic transitions to a congested state.

The last method used a nonparametric method based on dynamically binning the data using the values of the occupancy and then computing the mean flow in each bin. We started out with a fixed minimum bin width of $w = 0.01$, which means that there will be no more than $1/w = 100$ bins in total. We chose 0.01 because it provides sufficient resolution for the fundamental diagram in areas of low density. Furthermore, we required that each bin has at least k observations in each bin. Some experimentation for a few different stations showed that choosing $k = 200$ provided a visually smooth fundamental diagram.

We excluded from cluster analysis stations that satisfied any of the following conditions:

- **low variability** It's unlikely that 200 or more observations in one bin are the same.
- **low flow** This means that flow at every density level was less than 1 vehicle per 30 seconds.
- **few high density observations** We experimented a bit and found a reasonable filter to be fewer than 10 bins in an area of occupancy greater than 0.2. In contrast with the first two this may be real phenomena in the data rather than sensor errors; it simply means that very little congestion (high occupancy) events happened at that station in this data.

All this filtering brought the number of stations down from 3722 to 1379, so about 37 percent of the data was preserved. This is not a huge problem, because only about 50 percent of the stations even generate data in the first place.

Because there are more observations in areas of low occupancy we have more bins here. To construct the piecewise linear fundamental diagram we then simply define lines connecting the mean in each bin. This minimizes the assumptions we need to make about the fundamental diagram. This derived data could be used for further analysis of empirical traffic flow. For example, one can examine the maximum mean flow for the stations as in figure 2.

Clustering

For each of the fundamental diagrams we experimented with clustering based on the distance metric induced by the inner product on functions. Since the fundamental diagram representing flow as a function of occupancy is a function on [0, 1], the distance between two different fundamental diagrams f and g is defined as

$$d(f, g) \equiv ||f - g|| = \sqrt{\langle f - g, f - g \rangle}$$

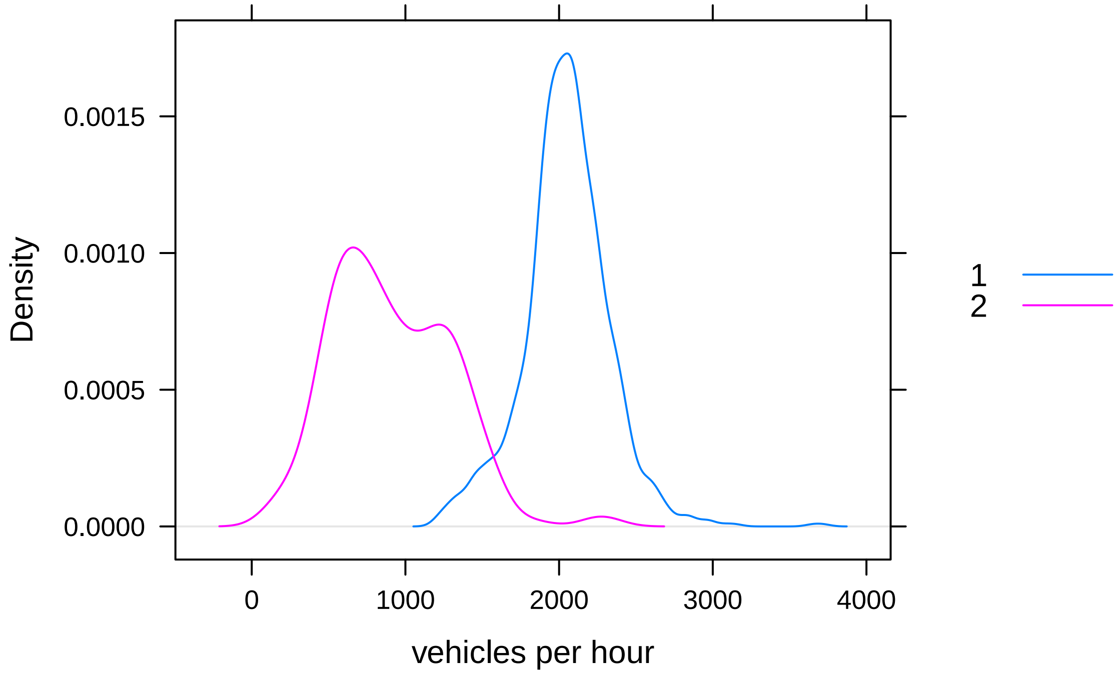

Figure 2: This density plot shows the maximum mean flow at each station, grouped by cluster.

where

$$\langle f, g \rangle = \int_0^1 f(x) \cdot g(x) dx.$$

We only considered piecewise linear functions, so all of these expressions have closed analytic forms that can be quickly computed.

The distance matrix provided the input into the Partitioning Around Medoids (PAM) algorithm. It is important here to use a partitioning method that accepts a distance matrix, because this allows us to cluster the functions themselves, which are not points in euclidean space. Inspection of the silhouette plots provided some evidence for clustering the fundamental diagrams into $k = 2$ groups. Silhouette plots for larger values of k provided no evidence that there should be more groups.

Figure 3 shows a sample of the fundamental diagrams corresponding to the PAM algorithm with two clusters. Cluster 1 contains 88 percent of the data, making it the dominant cluster. These fundamental diagrams look roughly like the triangular diagram we expect, and they have a max mean flow of around 2000 vehicles per hour. Cluster 2 contains the remaining 12 percent of the data, and these stations have much lower flow.

We can view the clusters on an interactive map here: http://anson.ucdavis.edu/~clarkf/fd/. We see points from cluster 2 on I 80 in the toll area for the Bay Bridge from Oakland to San Francisco.

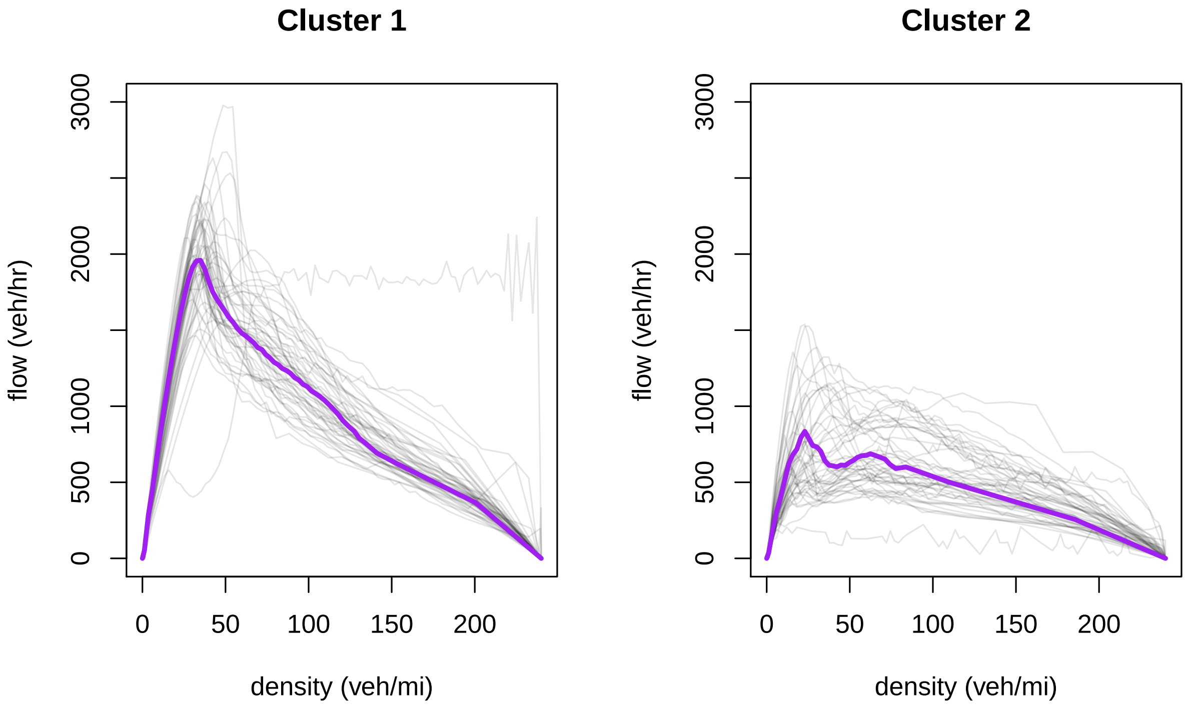

Figure 3: The bold lines come from the fundamental diagrams that have the lowest median distances to other stations in their clusters. In this sense they are the "median" stations.

We performed a similar cluster analysis on just the shape of the fundamental diagrams. We did this by normalizing each fundamental diagram f, ie. repeating the analysis on $f'(x) = f(x)/||f||$. This failed to reveal any interesting patterns in the shapes of the fundamental diagrams. Most shapes roughly follow the triangular fundamental diagram, with erratic deviations as seen in figure 4. Faulty sensors likely caused many of these anomalies.

Discussion

Figure 2 shows that the two clusters have different rates of maximal flow. The maximum flow in the dominant cluster 1 is around 2000 vehicles per hour, while the max flow in cluster 2 is much lower. Then we can answer the motivating question and conclude that the fundamental diagrams found in data naturally fall into groups with high and low flow. Furthermore, nonparametric models look quite similar to the triangular fundamental diagram.

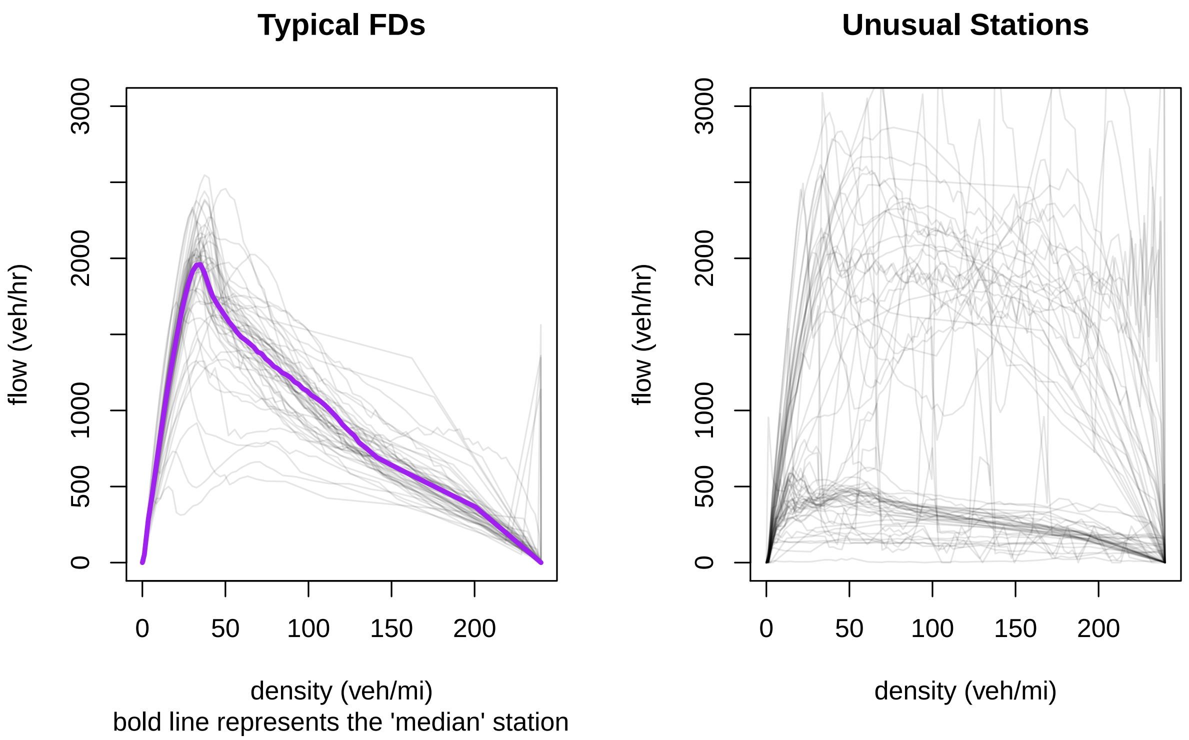

Figure 4: Typical versus unusual clusters

This paper demonstrated a technique for efficiently combining existing data analysis technologies to analyze data that will not fit in memory. Processing the entire data set allows us to go beyond a simple parametric fundamental diagram to a nonparametric model based on dynamically binned means. This allows us to more accurately estimate the function and then compare thousands of sensor locations across the Bay Area.

To extend this analysis, we could consider all lanes on the freeway, rather than just the second lane. This would produce a more complete picture of the traffic patterns. We could also join the PEMS data with data on historical weather and sunrise / sunset times to understand how changes in weather and light conditions influence the fundamental diagram.

References

Daganzo, Carlos F. 1997. *Fundamentals of Transportation and Traffic Operations.* Emerald Group Publishing Limited.

Kianfar, Jalil, and Praveen Edara. 2013. “A Data Mining Approach to Creating Fundamental Traffic Flow Diagram.” *Procedia-Social and Behavioral Sciences* 104. Elsevier:430–39.

Li, Jia, and H Zhang. 2011. “Fundamental Diagram of Traffic Flow: New Identification Scheme and Further Evidence from Empirical Data.” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2260. Transportation Research Board of the National Academies:50–59.

Qu, Xiaobo, Shuaian Wang, and Jin Zhang. 2015. “On the Fundamental Diagram for Freeway Traffic: A Novel Calibration Approach for Single-Regime Models.” *Transportation Research Part B: Methodological* 73. Elsevier:91–102.

Thusoo, Ashish, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. 2009. “Hive: A Warehousing Solution over a Map-Reduce Framework.” *Proc. VLDB Endow.* 2 (2). VLDB Endowment:1626–9. https://doi.org/10.14778/1687553.1687609.