Poisson Processes and Linear Programs for Allocating Shared Vehicles

Evan Fields

In 2019, micromobility—short urban trips taken on shared, lightweight vehicles such as bikes and electric scooters—became mainstream. For example, in Austin, the "scooter capital" of the United States, there were more than 525,000 micromobility trips during the two weeks surrounding South by Southwest, a festival held in Austin each spring. My day job is math modeling at Zoba, a small Boston-based startup providing data science tools for the growing micromobility industry. Much of our work involves using mathematical models and detailed historical data—when and where every trip starts and ends—to uncover patterns in customer-desired trips so that mobility operators may serve these trips.

The need to infer unobserved customer desires from observed historical data is best illustrated by the difference between utilization and demand. Utilization is simply the set of rides customers take. Demand, however, is the set of rides that customers would have taken in a counterfactual world where there were infinite vehicles everywhere all the time. Mobility providers should try to serve the actual rides users demand, not their past utilization—and these two sets of rides are almost never the same! For example, I might leave a coffee shop and look for a scooter to ride to work. Ideally there's a scooter waiting just outside the coffee shop, but perhaps I check and see that the closest scooter is a block away. So I walk down the block and begin a ride where the scooter is, not where I actually wished I could have started the ride. Because there are only finitely many vehicles and these vehicles are never perfectly distributed, many desired rides are substituted with an available ride or never occur—and are thus never observed—at all.

At Zoba, we help micromobility operators infer the spatio-temporal patterns of demand from their historical rides data; understanding demand allows vehicles to be placed to serve that demand. We model demand as a nonhomogenous Poisson point process. That is, the number of customers who would like to start rides in a given area over a given time interval has a Poisson distribution; the expectation of that distribution depends on both the area and the time. To infer patterns of demand for a given city, we partition the city spatially into small geographic cells and temporally into blocks of related times like "weekday mornings" or "weekend afternoons." Within each spatial cell and time block, the Poisson arrival rate is modeled as constant. Our task is then to estimate these cell-and-time-wise arrival rates.

To do so, we make heavy use of two great properties of Poisson processes: Poisson processes can be split into subprocesses, and Poisson processes have exponential

Evan Fields is the head of data science at Zoba. His email address is `evan@zoba.com`.

Communicated by Notices *Associate Editor Emilie Purvine.*

DOI: https://dx.doi.org/10.1090/noti2175

interarrival times. Informally, a Poisson process with rate λ can be split in the sense that if each arrival to the process independently goes to locations A or B with respective probabilities p and $(1-p)$, then arrivals to A and B are both a Poisson process with respective rates λp and $\lambda(1-p)$. Splitting processes allow us to convert between arrival rates for a spatial area and an individual vehicle. Under the moderate assumption that vehicles of a given type (such as scooters, bikes, etc.) are exchangeable, we can split the arrival of customers to a given location with k vehicles into k identical vehicle-specific processes, each with $1/k$ the rate of the overall process. For example, if five customers per hour arrive at a park looking for a scooter and three identical scooters are present in the park, we assume that each scooter individually has a Poisson arrival process with rate 5/3 customers per hour.

The second property we rely on is that Poisson arrival processes have exponentially distributed interarrival times. For an individual-vehicle Poisson process, these interarrival times have a natural interpretation: they are the durations of the periods when the vehicle is idle and awaiting the next customer. From historical rides data we can calculate the duration of each idle period for each vehicle. Thus we estimate the arrival rate of the vehicle-specific Poisson processes at a given cell and time, and by joining these vehicle-specific processes into a cell-wide process, we can estimate the overall arrival rate—the demand—for the cell.

Our demand estimation procedure explicitly uses only these basic properties of Poisson arrival processes. Nonetheless, nonhomogenous Poisson processes are such a natural model for micromobility that these simple properties carry us far; the simple demand estimation strategy outlined above (plus some corrections for weather and seasonality) is used in cities around the world to infer latent customer desires.

Inferred latent demand is most useful as a tool for operational decision making. A scooter company, for example, might be interested in distributing vehicles so as to maximize the number of rides customers take. At Zoba, we use large mixed-integer linear programming (MILP) models to help mobility companies discover ride-maximizing vehicle placements. These MILP models finely discretize time and space, and the inferred latent demands are input data to the model: they specify the average-case number of customers who want a ride at each time and location. Our MILP models thus attempt to maximize the number of rides that occur under average-case conditions.

These models can be fairly large. Roughly speaking, if we optimize vehicle placements for a city over the course of a day, there may be $\sim 10^3$ spatial cells and $\sim 10^2$ time periods for a total of $\sim 10^5$ variables representing the (expected) number of vehicles at each location at each time. The model's constraints describe the movement of vehicles throughout the city over the day: vehicles are conserved from one time period to the next and customers on rides move according to historically observed patterns.

Furthermore, because a mobility operator cannot deploy fractional vehicles, the thousands of variables representing per-location vehicle counts at the beginning of the day should take integer values. So a typical model may have hundreds of thousands of constraints and variables, including thousands of variables with (non-binary!) integrality constraints. Integer programming is NP-hard, and even with a state-of-the-art integer programming solver, solving such a large discrete model to certified optimality can take a prohibitively long time. In many mobility use cases, solution time matters more than certified optimality. Therefore, we instead solve the continuous relaxation of the problem—a problem instance without the integrality constraints—and apply specialized rounding heuristics to the resulting fractional solutions. In short, the geometric structure of mixed integer linear programs allows us to trade strict optimality in order to discover near-optimal (small duality gap) solutions quickly enough to inform dynamic shared mobility operations.

Evan Fields

Credits

Author photo is courtesy of Joseph Brennan.