ROI-based Task Optimization for High-Frequency E-Scooter Fleet Operations

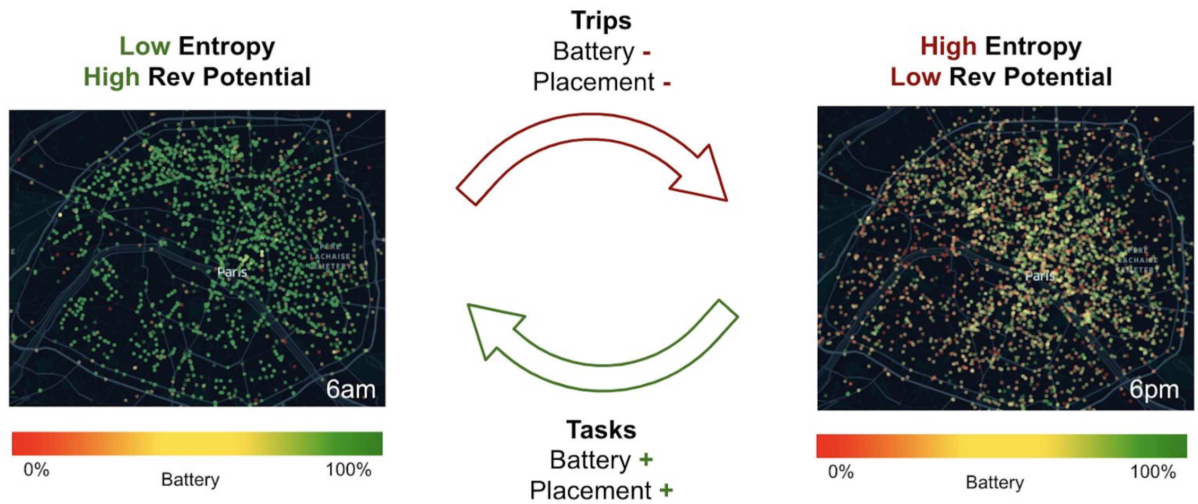
Benjamin Laufer, Dounan Tang
November 2019

Executive Summary

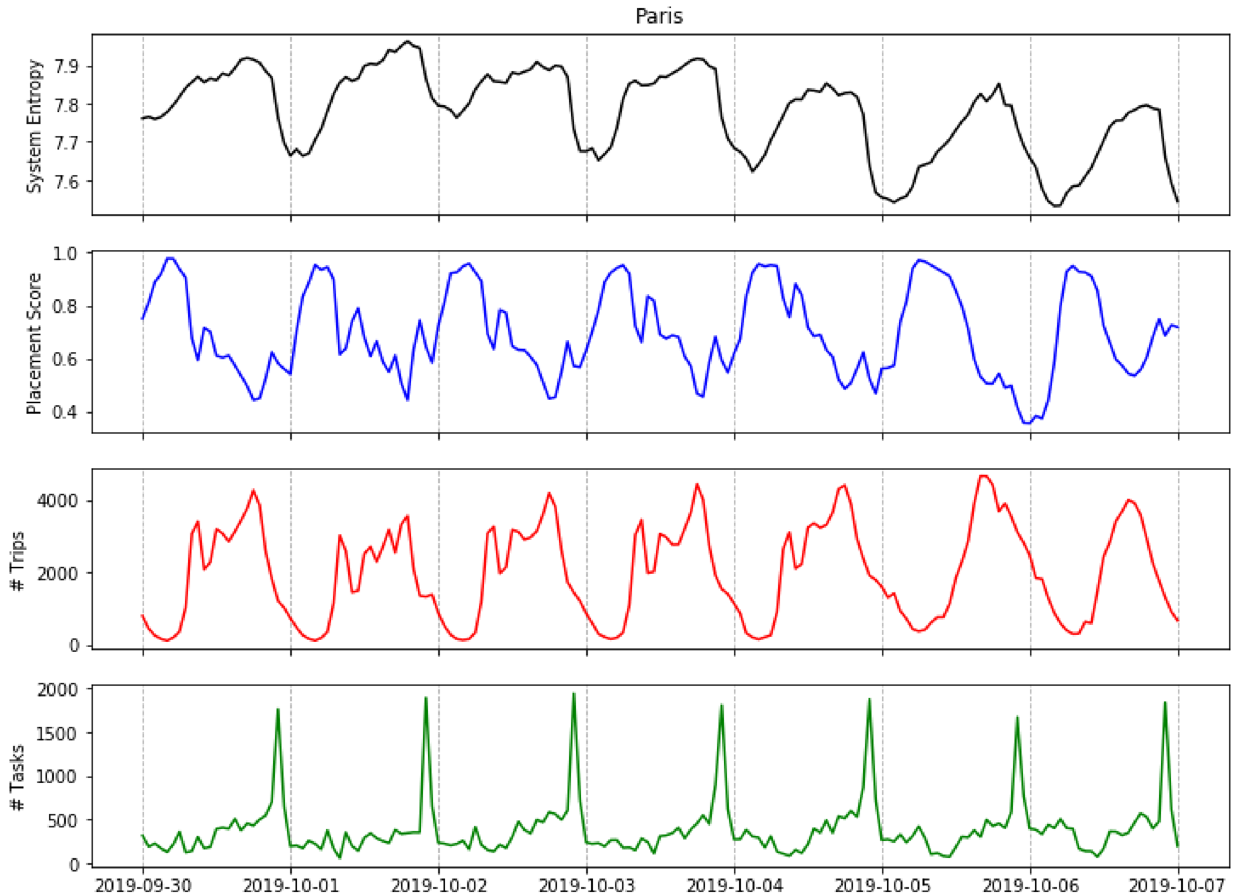
- A task changes the state of a scooter: its battery, location, operational hours, etc. As our operation requires more precision of execution, we want to evaluate the task performance, in other words, return on investment (ROI) for every single task. We compute a per-task ROI estimation on a scooter-basis:
$$ROI = Revenue - OpportunityCost - OperationCost$$
- We introduce the concept of operating cycle and use it to track the “lifespan” of a task. We utilize the technique of survival analysis and estimate the downstream revenue that a scooter could generate until the next non-operational event.
- We conducted a case study on juicer charging tasks. Historical data analysis allows us to identify regions where many negative-ROI tasks are being created, and refine our task-creation logic to reduce costs.

Background: Scooter Fleet System Dynamics

The status of a scooter can be described by a vector of location, battery level, and operational status, $S = \{L, B, O\}$. A rider takes a trip on a scooter, consumes its battery, and move it to another location (maybe better or worse). Consider the scooter fleet as a whole system, riders increase its entropy. An operator(juicer or OS) conducts a task on a scooter, charges its battery, and moves it to a better location. Operators bring the system entropy down to a healthy level.



Using Paris as an example, the figure above shows the spatial and battery distribution of the scooter fleet at 6 am (right) and 6 pm (left). In the morning, the battery level is high for most of the scooters and they are located in an orderly way at hotspots. In the afternoon, scooters run out of batteries and are distributed more randomly over the space. The figure below shows, in more detail, how the system evolves dynamically. From day to day, the entropy of the system (black curve) starts to increase in the morning, due to rider trips, and falls quickly after 10 pm, mainly due to charging tasks. Meanwhile, we see the placement quality score (blue curve) of the system reaches its highest point in the morning, directly after the deploy of fully charged scooters, and falls gradually throughout the day [for more details about placement score, see here].



System entropy and placement score describe the system at the overall fleet level. However, as our operation requires more precision of execution, we want to evaluate the task performance, in other words, return on investment (ROI) for every single task. Applications of this ROI metric includes task auto-creation, data-driven task prioritization, and task recommendation system. In addition, we see huge potential in bringing the current day-to-day operation to the next level: controlling the system entropy at an hourly level, increasing the placement score in the afternoon and providing a more reliable service 24/7. Some examples of ongoing projects include juicer rebalancing, charging hub, and daytime deployment. This brings difficulties to the task ROI calculation as the status of the scooter changes more often and downstream metrics are censored more often by tasks.

Task ROI Computation

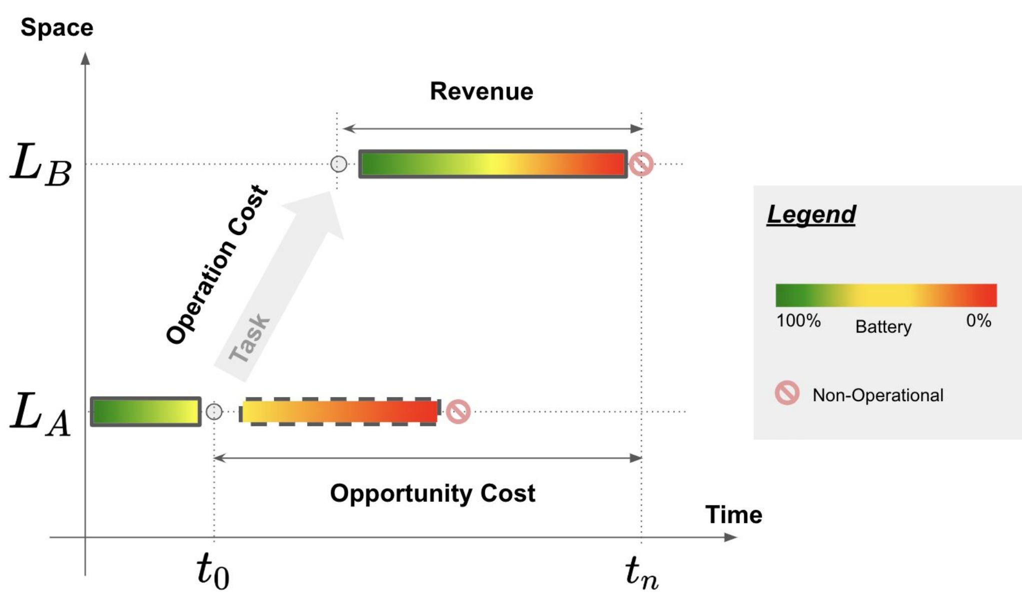
For an individual task, we measure the ROI of the task as:

$$ROI = Revenue - OpportunityCost - OperationCost$$

- Revenue

- After being moved to the new location (maybe be with battery charged), the revenue generated up to a certain time.
    - This time can be a fixed time period, e.g. 24 hours, or up to a certain event, e.g. collected by a juicer or a combination of both.
- To better align the operation cost to the cost margin, we propose to use the idea of the ‘deployment cycle’. The timestamp will be the time when the scooter becomes non-operational.
- In the future, we could further consider other indirect revenues, e.g. removing a low battery scooter from the street will reduce the error scan rate (bad advertisement).

- Opportunity Cost
    - If we leave the scooter at the original status, the revenue generated up to the same timestamp as we used to compute revenue.
- Operation Cost
    - The cost of conducting the task.



To put it down mathematically, a task $\mathbf{T}$ is an action that changes the status of the scooter from $S_A = \{L_A, B_A, O_A\}$ to $S_B = \{L_B, B_B, O_B\}$, consuming the resource of time, Δt, and money, C, where Δt is the service time of the task and C is the operation cost of the task.

$$\mathbf{T} = \{S_A \to S_B, \Delta t, C\}$$

Conducting the same task at a different time will result in different ROI, therefore the starting time of the task, t_0, will be an input to the ROI function. To make the formula general, the end time for revenue counting, t_n will be another input. We want to estimate the ROI of a certain task $\mathbf{T}$ starting at t_0 and forward-looking into the time t_n:

$$ROI(\mathbf{T}; t_0, t_n)$$

where t_n can be some predefined timestamp, e.g. end of the day, or a function of the scooter's status after the task, e.g. time to next non-operational event after deployment. We can then write down the task ROI as:

$$ROI(\mathbf{T}; t_0, t_n) = DSR(L_B, B_B, t_0 + \Delta t, t_n) - DSR(L_A, B_A, t_0, t_n) - C$$

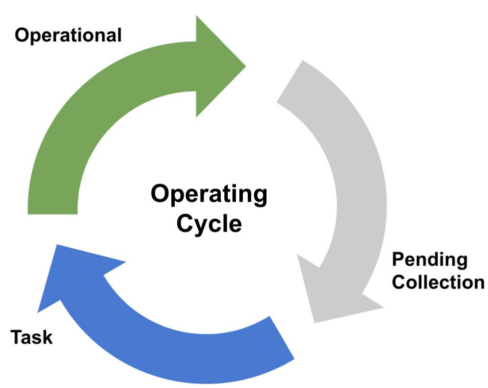
where $DSR(L, B, t_s, t_e)$ is a function computing the downstream revenue for scooter starting at a location, a battery level, from a start time t_s to an end time t_e. We then reduce the problem of computing task ROI to the problem of building up the downstream revenue function.

Challenges

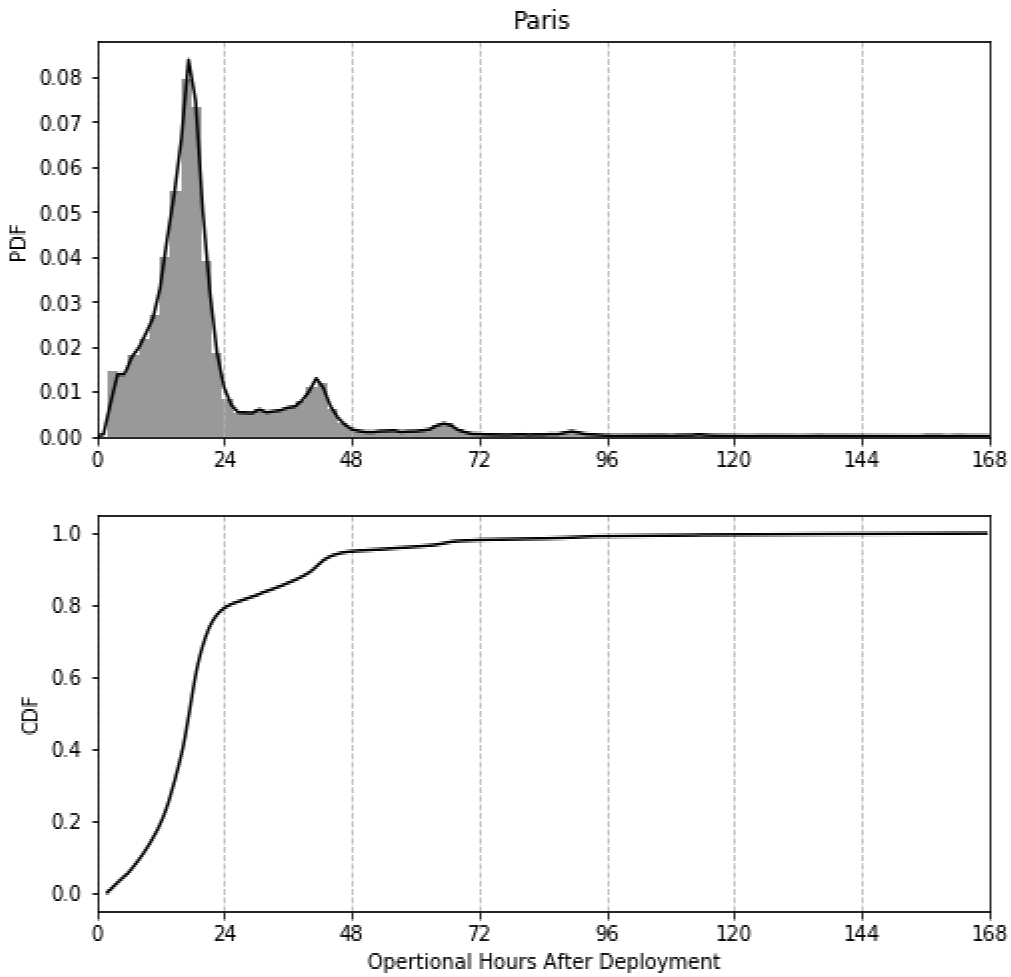
Temporal Censoring: Operating Cycle

A challenge for computing per-task ROI is the downstream revenue curve, which must forecast the arrival of revenue as a function of time of day, battery level, and location. To approximate this function, we have to understand how our historical data is censored. We can observe the arrival of trips for a bike starting a certain location and battery level at a given hour of the day. However, once the scooter is picked up, we don't know how many more trips it would have generated.

As mentioned above, the accounting time for downstream revenue is usually fixed in existing applications, e.g. to the next 6hr, 12hr or 24 hr. However, to compute the ROI for individual tasks, we suggest segmenting the time by the lifetime of a task. We defined the term operating cycle. We further breakdown the operating cycle into pending for collection time, task time and operational time. For the purpose of downstream revenue estimation, we focus on the operational time part, from task completion to the event that the scooter becomes non-operational.



Scooters become non-operational for a variety of reasons, including collected by juicer or task, becoming low battery or entering maintenance mode. For all scooters deployed through charging task in Paris, from 2019-09-02 to 2019-09-15, the operational time distribution is shown below.

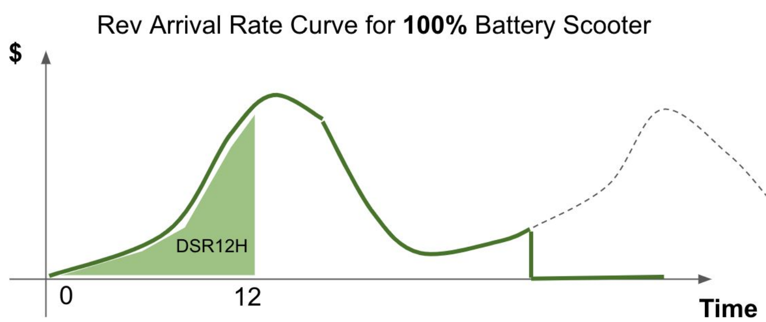


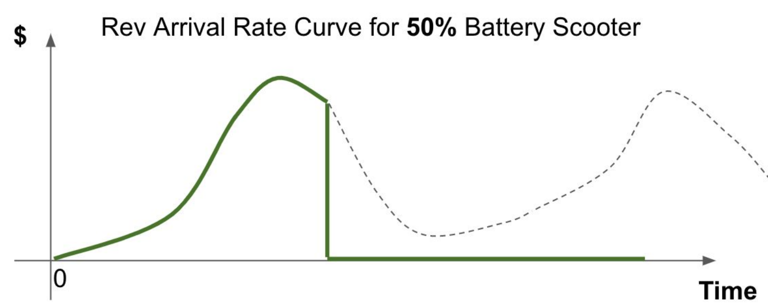
We see a concentration within the first 24 hr. However, the overall distribution is widespread. 19% of scooters have less than 12hr operational time after deployment, while 7% can operate for more than 48 hr. If we use 24 hr as a strict cutoff, we will overestimate the return of certain tasks as there is another task conducted on the same scooter within 24 hr. We will also underestimate the return of certain tasks as the scooter can last for more than 24 hours.

The nature of the problem is that the operational period is censored. In the future of the rebalancing task, or juicer bar, we expect to see the distribution shifts to the left and the operational time is censored more often before 24 hr. This is similar to the survival analysis where the event of interest is censored. The statistical solution for survival analysis is to compute a hazard rate for each time unite and compute the survival probability by integrating the hazard rate function. We take a similar approach here and we assume that, while operational, a scooter will accrue revenue at some (constant or variable) rate, $\lambda(t)$, in the unit of money per hour. This rate is a function of starting location, battery level, and time of the day. The downstream revenue is then the area under the curve up to a certain timestamp. The problem of estimating the downstream revenue function is equal to estimating the downstream revenue arrival rate function:

$$DSR(L, B, t_s, t_e) = \int_0^{t_e - t_s} \lambda(t; L, B, t_s) dt$$

The figure below shows examples of two downstream revenue arrival rate function.





In this example, we want to address the impact of the battery level. Assuming that two scooters are deployed to the same location, at the same time, with different battery levels, 100%, and 50%. The first figure shows the revenue arrival rate curve for the 100% battery scooter and second for the 50% one. For the 100% battery scooter, at a certain point, the revenue arrival rate will become zero as the scooter runs out of battery. For the 50% one, that point will come early. However, we are expecting that the underlying demand curve (dashed line) should be similar to both scooters.

Method

To fully predict revenue from a scooter's current state, we would need to model the arrival of revenue, hour-by-hour, after deployment time t_s. There are a few ways to approximate revenue generation:

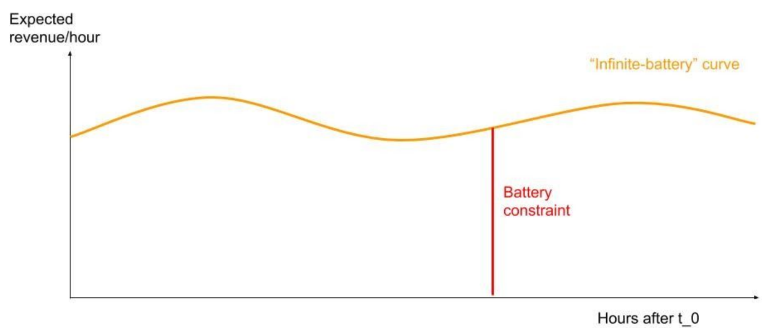
- [Method 1] Model expected revenue generation each hour after the start time, based on data available. This will require fitting $t_e - t_s$ variables for each geohash-hour (sparsity issues)
    - [Method 1.1] Assume revenue arrives at a constant hourly rate. This "Poisson" approximation allows us to measure revenue generated until next non-operational hour, and simply divide by the number of hours observed (accounting for any censoring).

- [Method 2] Use some kind of Origin-Destination transformation process to forecast scooter location over time, and then forecast expected revenue generation following scooters from t_s as they move through the city.

In v0, we use option 1.1 (with Poisson assumption).

Unconditional Downstream Revenue Curve

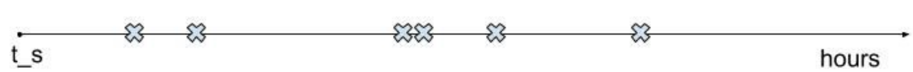
A bike can become non-operational because of a collection or a low battery level. While a scooter is operational and has a significant battery, however, there is always some probability that the scooter gets ridden. Thus, we consider some "true" or "infinite-battery" demand curve and separately model the cutoff caused by a low battery level.



The area under the curve is expected downstream revenue.

To estimate the infinite-battery demand, we can take scooters that are operational at some $\{S, t\}$ and sum their revenue generated, hour-by-hour, after time t. We want to know the revenue/hour (y-axis) for every hour (x-axis).

We assume initially that the expected arrival of revenue at a certain time t_s is constant over time (the infinite-battery curve is flat). We, therefore, model revenue generation as a Poisson process, with an arrival rate λ.



To estimate the arrival rate, we simply look at historical scooter hourly snapshot data. For every scooter described by $\{S, t\}$ in the past month, we measure the number of trips until the next non-operational status and divide by the number of hours elapsed. Our average revenue/hour is the lambda rate describing an infinite-battery scooter at $\{S, t\}$.

Lambda-approximation, DC, 5pm

Lambda-approximation, LA

yellow=higher λ, red = lower λ, size = # observations

We then consider battery level as a constraint, and there is a revenue per battery level constant can be estimated for each region. For a scooter at a certain time pair, $\{S, t_s, t_e\}$, we compute two numbers:

- Downstream revenue generated up to time t_e if the battery is unlimited.
- Maximum downstream revenue potential, given the current battery level.

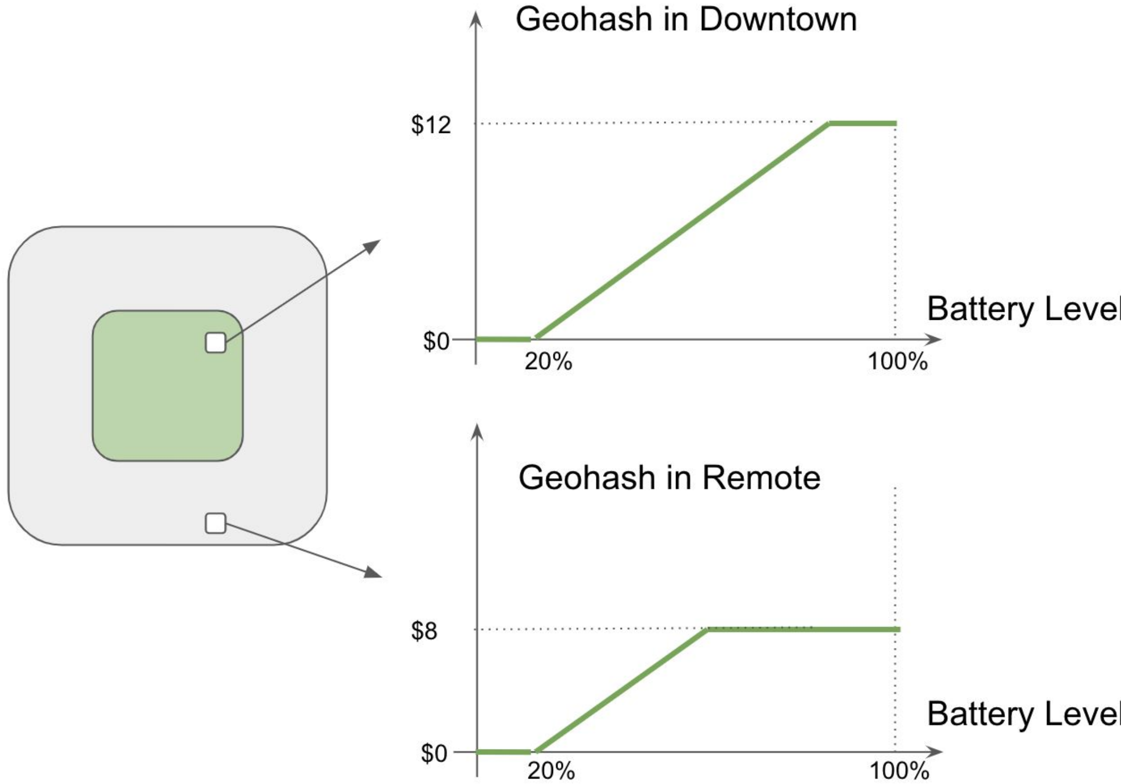
The final downstream revenue is the minimum of the two:

$$DSR(L, B, t_s, t_e) = \min\left(\int_{t_s}^{t_e} DSR_{\infty}(L, t_s), \left(\frac{revenue}{delta\ battery}\right) * B\right)$$

To sum up, to estimate the downstream revenue function we assume the following conditions:

- There is an underlying revenue arrival curve. It is different for different geohash and day of the week.
    - In the first version, we assume that the revenue arrival rate per hour is different for different geohash and day of the week. For example, a geohash in downtown could have a \$5 per hour revenue arrival rate while a geohash in the remote area could be \$1 per hour.
- There is a battery to revenue conversion rate. It is different for different markets.
    - Since the RPT is mainly based on the trip duration, it is correlated with battery consumption as well. This defines the maximal revenue a scooter can generate with a certain battery level. (no matter how popular the location is).

The figure below shows the downstream 24 hr revenue as a function of the location and battery level of a scooter.



Historical Charge Task Analysis & Insights

Implementation of the ROI estimation technique is significantly easier for historical tasks since two of the terms (RVD_B and C_j) are already defined for us. Instead of having to predict where a scooter will be deployed in the future, and how much revenue that scooter will accumulate thereafter, we only have to estimate the S_A downstream revenue.

This historical analysis can tell us two things: First, we can analyze the ROI-negative tasks that we're completing currently, and create rules to reduce the number of ROI-negative tasks. Second, we can visualize where high-ROI tasks are and understand what drives ROI for tasks.

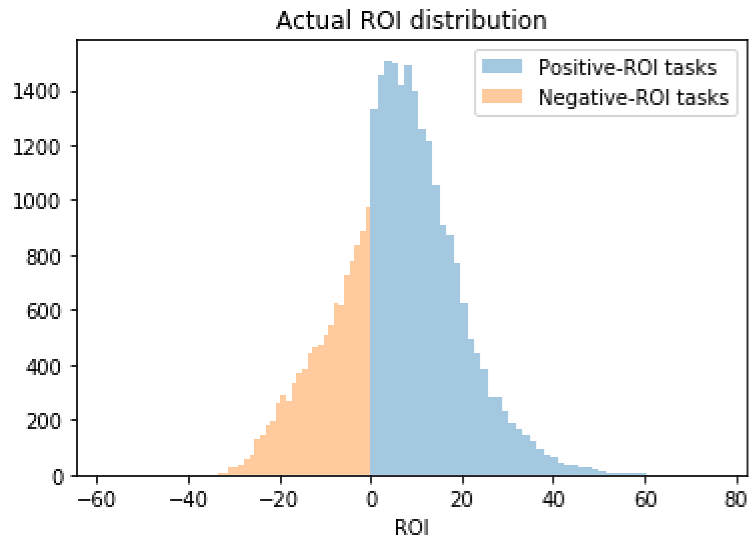
From these historical insights, we can also create ROI-maximizing recommendations for current levers. For example, we can use constrained optimization on historical tasks to find the battery threshold that maximizes cumulative return:

$$\max_X : \sum_i \delta_i ROI_i$$
$$st : \delta_i = 1\{B_i < X\}$$

The optimal battery threshold X is an estimate for ROI-maximizing battery threshold, assuming the region-wide battery threshold model as a constraint.

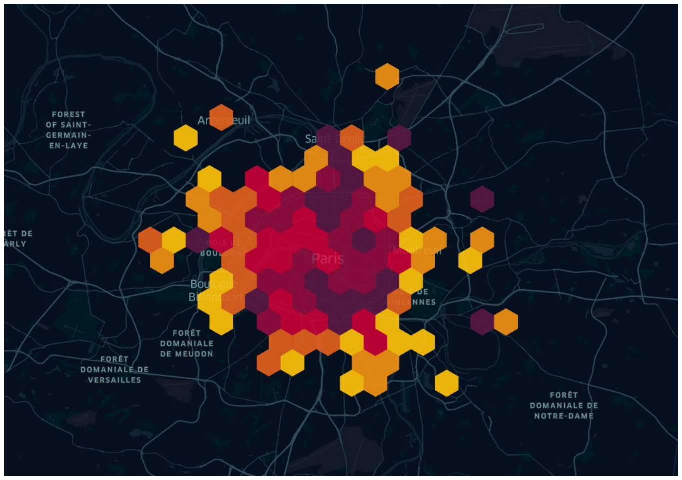
Case Study: Paris

Using two weeks of task data in Paris, we find a significant number of negative-ROI tasks that are completed on a given night.



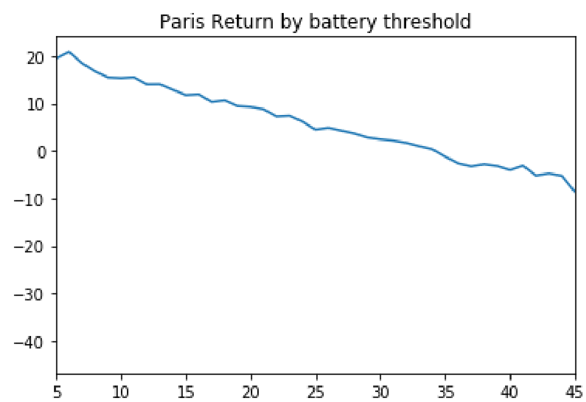
When we map out these tasks geographically, we see that downtown locations tend to have lower-ROI tasks. This validates our intuition that high-TVD subregions will have more opportunity cost associated with task completion, and should, therefore, have lower battery thresholds.

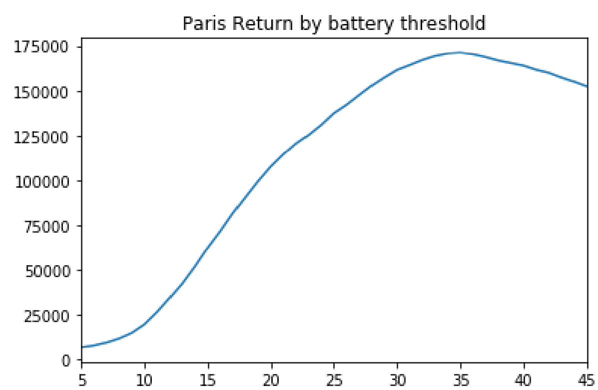
Paris geographical distribution of charge task ROI



The results suggest that segmenting battery thresholds by geohash and hour of day can significantly improve the efficiency of tasks and the ROI we observe from completing tasks.

To start very simply, we assume that battery threshold remains stagnant and region-wide. We simply wish to answer the question: what is the ROI-optimal battery threshold? When we look at the charge task ROI for scooters at different battery levels, we notice a (usually linearly) decreasing relationship between battery level and ROI. This is to be expected - high-battery scooters have higher opportunity cost associated with their task. We then plot cumulative return from all charge tasks below battery threshold b, for b in {5,...,current_battery_threshold}. The result is a concave function where the peak should be the maximum-ROI-generating battery threshold:





The peak return for Paris occurs at a battery level <35%, so we recommend 35% as the threshold. Next, we try to contextualize the projected ROI improvement by looking at average task ROI at baseline and the true optimal, “perfect prediction” ROI (which we’d achieve by only completing positive-ROI tasks and no negative-ROI tasks). Our results indicate that simply getting the battery threshold right may markedly improve task efficiency and per-task returns:

Current ROI: $1,707.9
Optimal BT ROI: **$12,223.0**
Perfect Prediction ROI: $17,873.2

Current cost/revenue: 22.12%
Optimal BT cost/revenue: **18.16%**
Perfect Prediction cost/revenue: 13.61%

More analysis for other regions can be found here

Next Steps

We are going to apply the existing model to the following application and experiments

- [P0] GBO Experiments
  - Test battery threshold recommendations (regional level)
  - Sub-regional variation. Having different battery threshold for different subregions
- [P0] Juicer Charging Task Auto Creation
  - Create charing task based on ROI
- [P1] Juicer Charging Dynamic Pricing
  - Price tasks based on task ROI (demand) and juicer willingness to accept (supply)

Model improvements in the future:

- [P0] Move from gross revenue to net revenue in ROI calculations
- [P1] Move from geohash6 (1km) to geohash7 (100m) accuracy. Utilize spatial interpolation and extrapolation methods to overcome sparsity.
- [P1] ML to predict RVD better for better ROI forecasting