

Inference Remapping for Vehicular Analytics

Paramvir Bahl Srikanth Kandula Ashish Patro Mohammed Shoib
Microsoft Research, Redmond WA 98052

Abstract— A phone+car+cloud system can improve many vehicular scenarios significantly due to improved telemetry and the resulting optimizations. The core problem however is the inability to cope when inputs are missing or impossible to obtain apriori. We develop the concept of inference remapping which learns using correlations how to best use available substitutes for the missing inputs. We also describe an end-to-end system *Sparc* that combines an OBD device, a phone app and a cloud backend to drive a variety of applications. In particular, for the case of fuel usage prediction, we obtain a mechanical engineering theory based model that is accurate to within 2% when given ideal inputs (OBD data). We show how to remap the inference to only use phone data (7% error) or data available from a map (within 20% error for half the rides, which is 4× more accurate than state-of-art). A side-effect of our model is that we can offer detailed comparative feedback to drivers on their driving behavior.

1. INTRODUCTION

We have to transform how we use automobiles to reduce urban gridlock, pollution, and our impact on the climate. Connecting vehicles to the cloud offers improved telemetry and can lead to optimizations that lower cost, both individual and societal, and improve user experience. A recent study [22] finds that 11-13% of commute time is due to traffic congestion, 10-17% of urban fuel is wasted at stoplights where there is no cross traffic, and 80% of accidents are due to driver distraction. A cloud-connected car can be routed along the least congested path, directed to an available parking space and can offer rich feedback on driving behavior and vehicle’s condition.

We would like to offer value in four different contexts: *Before driving*: E.g., Does my car have enough gas after my daughter/son drove it last night? What route should I take to save fuel/ save time? Is my car in good running condition? *While driving*: E.g., Where will I quickly find parking? Impending congestion or accidents along the forward route? I am running late, is my contact aware of my estimated time of arrival? *After driving*: E.g., How much did that trip cost? How could I have driven better? How do I compare to other drivers? *When not driving*: E.g., Is my child/mother driving safely? Where is my spouse? Is it safe to text him/her now?

Smartphones are a key enabler here. In part, because they have sensors and a backhaul to the cloud that cars may not otherwise have. More fundamentally however, phone software and hardware can update much more quickly relative

to the average lifetime of a car (averages 17 years [22]) allowing for rapid innovation. Further, it has become relatively easy to interface with the car. Bluetooth devices that plug into the on-board diagnostics (OBD) port or speak the car-area-network bus (CANBUS) protocol are available at low cost. These devices can monitor detailed vehicle state such as the instantaneous rate of fuel being injected into the engine. As a result, we note a recent substantial increase in startup activity in vehicular phone apps— some improve driver risk profiling for insurance [10, 23] and others offer trip analytics and assist with parking [24, 6].

We argue that a commonly recurring problem here is the inability to cope with missing inputs. For e.g., Automatic [6] infers how much fuel was spent on each segment *after* a ride. But, it cannot do so if the OBD device is not installed or if the phone app loses connectivity with the OBD device. Anecdotal evidence from our deployment reveals that this is by far the common case. Relatively few drivers install additional devices and even for those that do, the OBD feed is not always available for mundane reasons such as the driver bumping device with her leg or due to hardware and software faults on the OBD device.

Worse, in some cases, the inferences are needed *before* the inputs are available. For e.g., picking the most fuel-efficient route requires *predicting* how much fuel will be spent by a car along each potential route. But, how can we do this without the sensor feed from OBD for each route?

We call this the *inference remapping problem*. That is, suppose an inference algorithm requires some inputs and offers an output. Can we adapt the algorithm to work when some of these inputs are missing or are impossible to obtain (as they are for the case of predictions)? If solvable, this substantially increases the *coverage* of any inference technique— inferences may be available for {cars, drivers, routes, times} for which no sensor feed exists. Continuing our fuel usage example, with remapping, one could infer the fuel use for a car, driver, route and (future) time without the driver installing an OBD device or driving that car along that route.

Our basic intuition is simple: substitute the missing inputs with similar data that is available. For e.g., the speed limits, slopes or road segments and location of stop lights are available from a map. Could we estimate the absent OBD feed from such roadstate information? In some cases, other drivers or other cars may have driven along a route. Can we extrapolate from one driver or car to another? Also, the phone sensor information can partially substitute for the

OBD feed. Can we estimate what the OBD feed would have been from a phone sensor feed?

The key challenge however is that to get a high quality inference with these substitute inputs, one has to carefully capture the aspects of the ideal inputs that matter for the inference and identify how to estimate them from the substitute inputs. For e.g., we will show later that estimating fuel use on a road depends significantly on how the speed changes when driving on that road. The energy efficiency of an engine varies with the speed. Sharp speed increases require more energy than gradual increases. Decreases in speed due to braking mostly dissipate as heat so while the average speed of two rides may be the same, the ride with more changes in speed uses more fuel. There is also a complex dependence with the road slope and vehicular characteristics such as mass and frontal area. When going down a hill how much of the lost potential energy translates to speed increase depends on the aerodynamics of the vehicle. Another challenge is that the substitute inputs fundamentally lack some aspects of the ideal input. For e.g., OBD feeds reveal the torque generated by the engine. None of the phone sensors match directly to engine torque. More details follow later; but in short, transforming substitute inputs to mimic the ideal inputs is non-trivial.

In this paper, we present the Sparc system ¹ that takes a first step towards solving the inference remapping problem. For the example fuel usage inference, Sparc uses mechanical engineering theory to build a model of vehicular energy usage. For the case when ideal inputs are available from the OBD device, Sparc learns the parameter inputs for this model using regression from just a small amount of OBD data. Sparc also shows how to remap the inference to use inputs from just the phone sensors or just the information available from a map. The latter allows Sparc to predict fuel usage on a route even when Sparc has no drive-feed be it from a phone or OBD device for that route.

Further, we built Sparc end-to-end and offer results from a deployment study on twenty cars in two cities over six months. Sparc has an extensible data-collection platform, comprising a mobile app and a cloud-server backend. The app collects data from phone sensors and an OBD device in the car over Bluetooth. It has an asynchronous data-management framework leading to power-efficient data uploads without user intervention. Sparc map-matches GPS readings to acquire road grade and traffic information [18, 26]. We use data from OpenStreetMaps as well as a proprietary vendor. Finally, besides collecting telemetry information for data insights, Sparc also communicates to the cloud in real-time allowing it to enable the use-cases described in the second paragraph.

A side-effect of the remapping model used for fuel usage estimation is the ability to provide a precise breakdown of where fuel is spent. For e.g., Sparc breaks down the energy spent into the parts required to combat aerodynamic drag,

increase kinetic energy, increase potential energy, to combat rolling resistance etc. This breakdown helps answer questions such as: how does cruise control impact fuel use? Further, by comparing drivers traversing similar roads, we offer comparative feedback on driving styles such as acceleration and braking patterns and the impact of driving styles and car choice on fuel usage.

Our key contributions are:

- An extensible end-to-end smartphone+car+cloud system that collects telemetry data and offers various applications to drivers (§3).
- A first solution to the inference remapping problem for the case of fuel usage prediction (§2). The ideal inputs from OBD device yield a 2% error indicating that the mechanical-engineering based model is reasonably complete. We show that using just the phone sensors, the fuel estimation error is below 7% (§5.2). Further, when predicting fuel use on roads for which it has no driving feed, Sparc is roughly 4× more accurate relative to the state-of-art: 48% of trips are predicted to within 20% error (§5.3).
- Results for driver feedback and comparative analysis (§5.4) on a sizable longitudinal study identifies drivers with under-inflated tires (more rolling resistance than expected), a sedan driver with such gradual braking pattern that he reclaims more of the kinetic energy that would otherwise be lost than a hybrid SUV and drivers whose dominant energy loss is aerodynamic drag (high frontal area predominantly driven on highways at very large speeds).

We note however that the ability to remap inferences depends both on the property being inferred and the quality of the substitute input. For e.g., anomalies such as hard braking instances can be estimated using OBD feed; accelerometer readings from the phone can substitute but roadstate information would not suffice. Further, inference remapping is similar to techniques like PCA that leverage correlation between disjoint datastreams. However, whereas such work [25] saves energy by monitoring and communicating only the subset of inputs that is most useful for inference, here, we focus on obtaining the best possible inference given whichever subset of inputs and substitutes happen to be available. More specifically, the problem at hand – fuel usage prediction – is complex function rooted in the physical world. This leads to positives (can leverage mechanical engineering theory) and challenges (many factors affect fuel in intricate ways). Finally, GreenGPS [16] was the first to tackle the problem of fuel usage estimation. Sparc newly offers driving feedback and develops inference remapping. The major change is a much more detailed modeling of energy use and remapping. Without either the estimation errors were substantial on our dataset perhaps because it contains rides from urban settings with very different congestion profiles and a variety of road slopes. Our dataset lacks data

¹Sparc = Smartphone + Car + Cloud

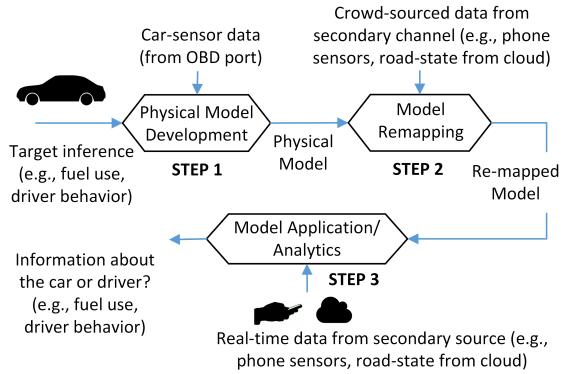


Figure 1: Inference remapping: We build a physical model using data from the car sensors and approximate it using training data from a secondary channel. This approximation helps us infer the car/driver state even in the absence of car-sensor data from the OBD port.

from very cold and very hot places, so there may still be some aspects missing in the model. We also take care to point out that not all of the use-cases have been implemented (details follow). However, we believe that the Sparc system and inference remapping are a good step towards enabling a phone+car+cloud architecture.

2. INFERENCE REMAPPING

In this section, we present details of our inference-remapping approach. The key idea is to carefully learn how to use substitutes when desired inputs are missing or impossible to obtain (Fig. 1). First, we develop a physical model that uses the ideal inputs to make some inference (such as how much fuel is used in a trip? time to next fillup? *etc.*). The features of the model depend on the trip under consideration and are computed from the ideal inputs, this could be just OBD data or some fusion across many sources. The parameters of the model are learnt from regression over ground truth. Second, we remap the features in the physical model so that they can be derived using only the available data, which is some subset of ideal along with substitutes (*e.g.*, just phone sensors, just road-state information from offline maps in the cloud, historical information from other drives *etc.*). To achieve the re-mapping, we train a new set of parameters that relate the available data to each of the features in the physical model. For this, we use training data that contains the ground truth, the ideal data and the data from the relevant secondary channels (*e.g.*, phone sensors, dash-mounted GPS, *etc.*). Finally, during inference we apply the most appropriate model given the data available.

To make the discussion more concrete, we focus on a specific case study: we show how to build a physical model that infers the amount of fuel-use in cars using data from sensors in the car. We then train two remapped models— one that uses only data from the phone sensors and the other using only data from a map— to approximate the physical model. We evaluate each of these models on scenarios where they apply. For e.g., at the end of a drive where the ideal OBD

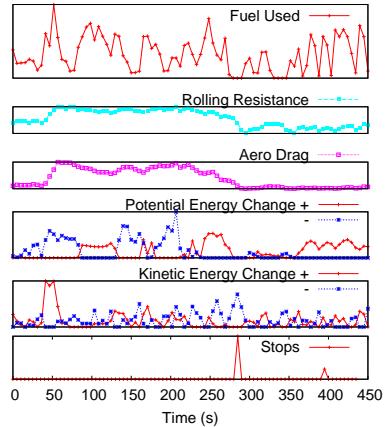


Figure 2: Anatomy of an actual ride. Energy to counteract rolling resistance is proportional to distance traversed; for aerodynamic drag it is proportional to $v^2 * \text{distance}$. The user starts on surface roads, enters a highway on a downhill ramp and picks up speed (after 50 s). The highway goes up and down a sequence of hills ending up at a lower elevation than where the user joins the highway (100-220 s). At around 250 s, the user exits the highway and stops at a traffic light. The remaining trip involves surface roads that gain elevation and finally a stop.

data is available, we use the physical model to compute trip analytics. For predictions, we use the remapped model that only uses maps and historical congestion data.

2.1 Step 1: Model Development

Burning fuel produces energy. Hence fuel usage can be estimated from the total energy expended during a trip. Energy is expended for various reasons. Figure 2 depicts an example trip in our dataset. The caption details what happened during the trip. Note how the instantaneous fuel used (at top) varies during the course of the drive. It starts at a low value when the user is on surface roads and hits the first peak when the user increases her speed to join the highway (at 50 s, correlated bump on increase in kinetic energy). The peak does not last for very long since the ramp that leads to the highway goes downhill (note the decrease in potential energy at that time). Note that the energy to combat aerodynamic drag ($\propto v^3 t$) and rolling resistance ($\propto vt$) are larger when the driver is on the highway (from 50 to 250 s). Once the user reaches highway speed, note that the fuel used goes up and down in sync with the highway (compare change in fuel to changes in potential energy). At stop (250 s), fuel usage returns to a small value. Finally, on surface streets note that the bumps in fuel use track bumps in both potential and kinetic energy since the streets changes in elevation and require frequent changes in speed. To sum up, total energy spent is an intricate function of several factors; each of which can be dominant depending on the conditions.

Model Summary: In summary, we informally note that the grade of road impacts change in potential energy as well as the rolling resistance. The speed of vehicle impacts the change in kinetic energy and the aerodynamic drag that is to be countered to sustain the speed. Braking dissipates extra

kinetic energy into heat. Vehicle-specific parameters such as the mass impact both potential and kinetic-energy terms. The vehicles' aerodynamicity affects drag. The engine and drivetrain efficiency impact how much useful energy is generated from burning fuel. Driver-specific behavior such as hard acceleration, braking, and transmission shifts also impact energy use: when accelerating hard, the vehicle's fuel injection unit has less time to modulate fuel injected leading to less useful energy for the same fuel burnt. Finally, miscellaneous aspects include windows being open, the temperature, use of A/C and other electrical equipment. The detailed model follows.

Model Details: To develop an *instantaneous* model of energy consumption from first principles, consider a small period of time Δ_t . Suppose that the vehicle moves with velocity v on a road of grade θ , changes velocity by Δ_v , and burns fuel at a rate f .

$$(\text{instantaneous}) \text{ Energy generated by the engine} = \eta f \Delta_t$$

where η , usually termed engine specific fuel consumption, indicates the engine's efficiency in burning fuel; this is a function of torque and engine revolutions-per-minute (RPM) [8]. Slower speeds (low RPM) and/or very high torque values lead to lower η values. However, most engines have a large operating region where the engine's efficiency is roughly the same. Typical combustion engines have an efficiency around 30%; *i.e.*, roughly 30% of the heat produced by burning fuel is converted into mechanical energy.

$$(\text{instantaneous}) \text{ Mechanical energy at engine} = \tau \omega \Delta_t$$

where τ is the torque and ω is the RPM at the engine. This energy is used in a few ways. The (Instantaneous) energy spent can be attributed to the various reasons shown in Table 1. Note that some of the energy *losses* can be used to offset other needs. For instance, one can slow down without braking, by letting loss in kinetic energy compensate for rolling resistance and aero drag. While every increase in potential or kinetic energy requires burning fuel, their loss is not fully recovered. For example, most braking dissipates kinetic energy into heat. Thus, we assume some recovery factors (R_1, R_2) and treat the changes differently based on whether they are increases or decreases.

By the law of conservation of energy, we have:

$$\text{Energy generated} + \text{Recovered} = \text{Energy used.}$$

Using all of the energy components show in Table 1 and re-

arranging terms, we have:

$$\begin{aligned} \eta f \Delta_t &= \tau \omega \Delta_t \\ &= P_e \Delta_t + P_s I_s \Delta_t \\ &\quad + \frac{mg[\sin \theta]_{0+} v \Delta_t + mv[\Delta_v]_{0+}}{\eta_t} \\ &\quad + \frac{c_{rr} mg \cos \theta v \Delta_t + \frac{1}{2} c_d A \rho v^3 \Delta_t}{\eta_t} \\ &\quad + R_1 mg[\sin \theta]_{0-} v \Delta_t + R_2 mv [\Delta_v]_{0-}. \end{aligned} \quad (1)$$

2.1.1 Features of a Trip

Observe that the physical energy-consumption model in Equation 1 relates fuel use (on the left) to measurable aspects of the trip (on the right). We will call the measurable aspects of a trip to be the *features* of that trip. Table 2 lists our current set of features. How to estimate these features in the ideal case? A device plugged into the OBD port of the car can report these sensor readings: f , v , ω , and τ , which represent the *mass air flow* sensor output (*i.e.*, the fuel injection rate), vehicular speed, RPM, and torque as measured by the vehicle's engine control unit. Phone's GPS provides location, which when map-matched can reveal the slope of each road segment θ . The data is usually sampled once every few seconds at each of these sources but needs to be fused properly (more details in §2.2.1).

Note that the values in the right column of Table 2 are a multiplicative parameter away from the corresponding energy term on the left in the vehicular energy model (Equation 1). These parameters m , A , and η , stand for the mass of the car, its effective area and the engine's efficiency, which is itself a function of torque and RPM. These coefficients are specific to a car and can vary from one trip to the other.

Given trip features and the fuel usage ground-truth information from OBD, we pursued a few approaches to learn car-specific multiplicative parameters: (a) linear regression with the above features, (b) non-linear regression with the underlying raw variables such as velocity, time and slope (v, t, θ), (c) a classifier that uses discretized "fuelUsage" as the label (*e.g.*, [0-1] gallons, [.1-.2], ...) and (d) decision trees. Linear regression worked the best when used in the following way: (1) 10-fold cross-validation to avoid overfitting, (2) to keep the parameters robust to trip duration, combine features from contiguous epochs so as to effectively train on epochs of many different sizes.

Why does linear regression do well? Because, at first blush the features are not even independent (see Table 2), so shouldn't linear regression be a bad choice? We found that using the underlying variables (*e.g.*, v, θ) as features requires learning a non-linear model for which known algorithms are less effective. Decision trees or a classifier with fuelUsage would have been better if the epochs were divisible into regimes where the relationship between fuel usage and the trip features differs substantially. Decision trees did obtain slightly lower error, however the trees were much bigger, hinting at potential overfitting.

Source	Energy model	Contributing factors
Losses at the wheel	$mg[\sin\theta]_{0+} v\Delta_t$	To oppose gravity. Here, m is mass of the vehicle.
	$c_{rr}mg \cos\theta v\Delta_t$	To oppose rolling resistance due to visco-elasticity (of the part touching the road bends) and the pressure differential in the tire due to movement; the coefficient of rolling resistance, c_{rr} is about .01 for radial tires on concrete; c_{rr} depends weakly on v^2 and strongly on road conditions (e.g., concrete vs. sand differs by 3×); friction is a much smaller component.
	$\frac{1}{2}c_d A \rho v^3 \Delta_t$	To oppose aerodynamic drag.
	$mv [\Delta_v]_{0+}$	To increase kinetic energy.
Electrical losses	$P_e \Delta_t$	P_e is the electrical load induced by A/C and other car accessories. Alternators are 40-60 % efficient; load due to headlights is 110 W; A/C can be up to 720 W; not a large factor; about 2-5 mpg.
Standby losses	$I_s P_s \Delta_t$	P_s is the power drawn to keep the engine in standby when the car is stationary. I_s is an indicator variable denoting car is in standby; we set $I_s = 1$ if $v < 5$ mph.
Drivetrain losses	η_t	This term denotes the fraction of engine's energy that is delivered to the wheel by the transmission system η_t ranges from 94 % efficient with manual to 70 % for some old automatic transmissions; the effective number, over a ride, depends on how many gear shifts were needed; e.g., maintaining speed on a flat road vs. frequent starts-and-stops are very different.
Other losses	$mg[\sin\theta]_{0-} v\Delta_t$	Loss in potential energy e.g., when going down a slope.
	$mv [\Delta_v]_{0-}$	Loss in kinetic energy e.g., when slowing down.

Table 1: Components that contribute to the instantaneous energy spent in a car.

feature represents	how computed
Energy from burning fuel	$\sum f \Delta_t$
Energy generated by engine	$\sum \tau \omega \Delta_t$
Change in kinetic energy, +ve (and -ve)	$\sum v [\Delta v]_{0+}$
Change in potential energy, +ve (and -ve)	$\sum [\sin\theta]_{0+} v \Delta_t$
Aerodynamic Drag	$\sum v^3 \Delta_t$
Rolling resistance	$\sum \cos\theta v \Delta_t$
Standby	$\sum I_s \Delta_t$
Miscellaneous	$\sum \Delta_t$
Supporting	$\sum v^2 \Delta_t$ $\sum v \Delta_t$

Table 2: Trip features that help with estimating energy use.

2.2 Step 2: Model Remapping

In this section, we will show how to remap the physical model of Equation 1 to data from two different substitute sources. First, we show how to re-map the physical model to use phone-sensor readings and a map (§2.2.1) and then to the road-state information obtained from just the map (§2.2.2).

The trip features shown in Table 2 are the anchor points for the remapping. We observe that the substitutes are lacking in a few ways: neither can account for engine torque τ ; map only has static information such as speed limit, grade and stop lights but lacks dynamic changes in speed. Hence, both these remappings lower inference accuracy (due to lack of requisite data) but increase coverage (since they require more easily available data).

2.2.1 Remapping to Phone sensors + Map

Use cases: We will argue below that not having the OBD sensor only lowers inference accuracy slightly! Though this remapping cannot help with predictions since phone sensor readings are only available after the drive, it is helpful in a few cases: (1) after a driver has collected some OBD data, she no longer needs the OBD device to obtain trip analysis and driving feedback; the car parameters inferred from ground truth are reusable and trip features can be obtained from phone app + map and this remapping and (2) additionally, even a driver who never installed OBD device can obtain trip analysis and driving feedback by using either the best-guess car parameters based on manufacturer specs or by matching onto the car parameters derived for similar cars (similarity based on model/ year/ engine size/ etc.) that have reported OBD data to Sparc.

Remapping: The goal of remapping is to compute features in Table 2 from just phone sensors + map. We have no equivalent information for the $\tau \omega$ term. From Eqn 1, this means that any changes in η , the efficiency of burning fuel, are not captured. To compensate, we do the following: (a) derive velocity v from phone sensor instead of OBD, (b) derive θ from GPS location + map matching as before and (c) divide data into regimes that are likely to have the same η value and train different parameters by regression per regime. This works because most engines have one large operating regime where η is roughly the same and other regimes (slow or very high speed) where η is lower. Dividing the training data into regimes helps learn the different η s.

Data fusion and cleaning: When fusing readings from different sources, a key insight is to compute a sum (or integral) over the instantaneous observations. Given data that is ob-

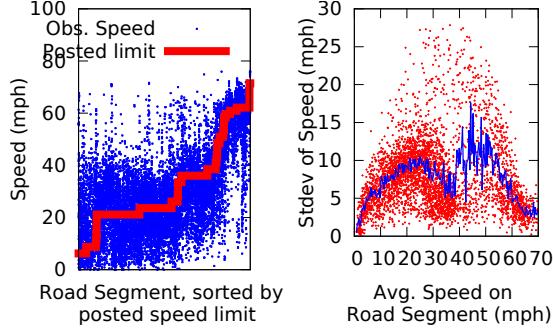


Figure 3: Speed on segments: the left graph shows the (average) speed achieved by all drivers *vs.* the posted speed limit per road segment. We see that the average speed can be quite different from posted speed limits. The right graph compares the variation in speed *vs.* the average again per road segment. Solid blue plots the trend; we see that the standard deviation is between 10 and 15 mph for most road segments.

served at different times (and frequencies), we want to compute feature values that are joint functions of the raw values. To see why this is hard, observe that the grade of the road can change many times between successive speed readings from GPS; the fuel sensor ground truth reading may have been sampled three times per speed sample and none of the sampled times may match.

Sparc extracts feature values per *epochs* which are non-overlapping intervals of time sized such that each data source is observed a few times per epoch. We use epoch size of 5 s for most cars but some older cars need larger epochs of 60 s. Computing feature value mimics a piece-wise integral over the raw readings. First, the readings are transformed into piece-wise linear functions of time. For *e.g.*, if the i 'th velocity reading at time t_i is v_i , then

$$v(t) = v_i + \frac{t - t_i}{t_{i+1} - t_i}(v_{i+1} - v_i) \quad \text{if } t \in [t_i, t_{i+1}].$$

Second, the feature value is computed by integrating over the piece-wise linear functions of the corresponding readings. For *e.g.*, for epoch $[t_b, t_e]$, the rolling resistance feature is $\int_{t_b}^{t_e} \cos\theta(t) v(t) dt$.² Finally, wherever possible we avoid integrating over a *rate* if the underlying *value* is available. For *e.g.*, when using GPS data, we replace $\int_{t_b}^{t_e} v(t) dt$ with a piece-wise sum of the distances between the locations observed during $[t_b, t_e]$. Otherwise, even small errors in estimating the rate accumulate over time.

We handle missing readings carefully. Since many of the readings are *rates*, *e.g.*, f is fuel injection rate, interpolating a large hole with the value at the ends can lead to large error. Hence, we only use epochs that have no holes larger than a threshold and have data for at least a threshold fraction.

2.2.2 Remapping to Map (Road-state Data)

Use-cases: This remapping is likely to result in much more error as we discuss below. However, it is very important since the map information is static data that is always avail-

²We use closed form expressions for each of the integrals.

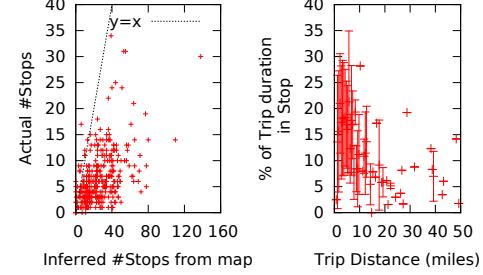


Figure 4: Comparing actual stops (speed < 5 mph) with those inferred from the map.

able, the remapping can predict fuel usage apriori and on roads that were not observed before by Sparc.

Inadequacies: Maps only have static information such as speed limit, slope and stop light information. But, fuel use depends on the changes in speed: kinetic energy depends directly on the change in speed and terms like aerodynamic drag $\propto v^3$ vary substantially with small changes in velocity.³ Hence, using the speed limit as GreenGPS [16] does leads to sizable error.

Measurements from our deployment help quantify this problem. Figure 3 (left) compares average speed on a road segment with the posted speed limit on that segment. The plot on the right depicts the standard deviation in speeds observed on that road segment *vs.* the average observed speed. Note that the actual speed on a segment is correlated to the posted limit but there is substantial noise.

Stop lights are another complication. Figure 4 compares the actual stops observed in a trip with the inference from the map. We say that a vehicle is stopped if the speed from GPS is below 5 mph. The actual stops can be less than that estimated from the map (*e.g.*, traffic signal is green) or many more (*e.g.*, due to congestion). In our dataset, we see both types. Worse, not every stop costs the same amount of energy. Stops on higher speed roads have a larger effect since more kinetic energy is lost per stop.

Remapping: We realize from above that there is at best a tenuous connection between map info per road segment and the desired features (Table 2 that correlate with fuel usage). However, our key insight is that similar road segments have similar desired features. That is the desired features of a segment with posted speed limit of 40mph, 2% grade, with a highway on one end and a stop light at the other are likely to be similar to another segment with the same characteristics! With just a few drivers, Sparc already has ground-truth features for tens of thousands of road segments. This set is large enough that any unseen road segment has a large number of similar segments for which features are available.

Hence, Sparc's strategy to remap is (a) given a route on the map, obtain from the map the information shown in Table 3 for each road segment along the route, (b) add time-of-day/ day-of-week values to capture congestion, and (c)

³say vehicle has speed 1 and 2 for unit time each; aerodrag $\propto \sum v^3 t = 9$; but, using average velocity leads to $2 * (1.5)^3 = 6.75$

Road-state parameters	how computed
road length	x from map
road speed limit	v, v^2 , from map
road grade	θ , from map
road rolling resistance	$x \cos\theta$
road change in potential energy, +’ve and -’ve	$x[\sin\theta]_{0+}$
road aero drag	xv^2
road rush hour velocity multipliers	r_v, r_{v^2}
road num stops	s

Table 3: Road state parameters

use training data to learn a regression model from the values in Table 3 to each feature in Table 2. An example such model would compute how much energy will be spent in kinetic energy change on a road segment based on the kinetic energy change observed on similar road segments and at similar times-of-day/ days-of-week.

2.3 Step 3: Model Applications

In this section, we describe how the developed models are used in different scenarios with different available inputs. In addition, we describe how to offer feedback to drivers.

[A] Estimate fuel use after a trip. As discussed already, a driver need only use an OBD device for a short while to provide training data. Subsequently, Sparc can work roughly equivalently with and without data from the OBD device by using just the phone sensors and the map via the remap model in §2.2.1. Drivers who never install OBD can also receive estimates if Sparc happens to have training data from similar cars (make, frontal area, engine size etc.). Sparc’s remapping simplifies the burden on the user substantially.

[B] Predict fuel use before a trip. As discussed already, the remap model in §2.2.2 that uses just the map information can help here. The error is likely to be smaller as more and more data reaches Sparc because, then it would be able to train the remap model at finer granularity; more support lets it divide the observed road segments into finer bins and train different remap models per bin.

[C] Provide feedback on driver behavior. Observe that a side-effect of the model construction is that we can breakdown, at the end of each trip, how much of the energy went into each of the components in Table 1. That is, for example, we could tell for each trip how much energy went into combating aerodynamic drag vs. rolling resistance vs. changing kinetic energy. Further, all our remapping models preserve this property. With the substitute input (just map or just phone+map), we can still breakdown per component.

We leverage this property to offer feedback to drivers. Such feedback takes two forms. First, we can compare per component values to what they should be leading to corrective guidance to drivers. An example would be rolling resistance using up a larger fraction of energy than it should for a ride on surface streets due to under-inflation of tires. Second, on segments that are driven by multiple drivers, we could

compare their relative driving behavior. An example here would be identify driver or car peculiarities such as graceful braking or aerodynamic drag that are atypical relative to other drivers/cars. More specific examples from our deployment are in §5.4.

In conclusion, we note that so far we have discussed how remapping can estimate fuel usage for a {car, driver, route, time} tuple even when many of the ideal inputs are missing or impossible to obtain. The approach can easily extend to similar cases such as estimating the time-till-fuel-dipping-below-10%. Rare occurrences such as the likelihood of accidents are amenable to similar remapping because the substitute inputs (roadstate) has aspects correlated to the ideal features; however, doing so would require more data due to the rarity of occurrence. Anomalous driver behavior, as discussed before, such as likelihood of braking is less amenable: roadstate information is unlikely to suffice, but phone+map may be useful. While more work is needed, we believe that remapping is a crucial part of any phone+car+cloud system.

3. SPARC DESIGN

Here, we describe aspects of the Sparc system. There are three main components: (a) an OBD device to connect vehicle with the smartphone, (b) a smartphone app that serves as a driving assistant, collects sensor data and acts as a conduit between the vehicles and cloud, and (c) a cloud based backend that supports inferences (map matching, building models, remapping) and real-time interactions.

3.1 Communicating with the car

To communicate with the vehicle, Sparc uses off-the-shelf bluetooth devices that plug into the On-Board Diagnostics (OBD) port in our volunteers’ vehicles. Vehicles sold in the US after 1996 are required to have an OBD port; the port is often found underneath the steering. The device can poll data (e.g., speed, RPM, temperature etc.) from sensors on the vehicle. Figure 5 shows two of the devices that we used. They retail for under 30\$ and can fit non-obtrusively. Devices from two off the three manufacturers that we tried were reliable; the third had > 50% failure rate. The OBD device interfaces with the vehicle using the Controller Area Network (CAN) bus. Readings from the OBD device are used as features for post-drive analysis of trip activity. They also help train remapping models such that subsequent inferences only need phone sensor data or map based data. As future vehicles become more capable, it might be possible to poll even more data directly from the vehicle using standardized APIs (e.g., over DSRC).

3.2 Sparc smartphone application

We built a smartphone application for the Windows Phone platform. The app connects to the OBD device via Bluetooth and queries the engine control unit of the vehicle. The Parameter Identification Number (PID) in the query identifies



Figure 5: OBD-II devices used

the information requested (e.g., 010D for speed). Vehicle manufacturers implement several proprietary PIDs but to be broadly applicable we only rely on the PIDs in the OBD-II standard. Some vehicles do not have the sensors needed for some PIDs; for e.g., some Audi models do not have the fuel injection sensor. Further, sensor values update at different timescales and queries on some (older) cars take over 10x longer than normal for the same PID. Hence, our app first sweeps the PID space to identify PIDs that offer non-trivial responses and the frequency at which they update. Then, it generates a polling schedule such that the more relevant PIDs (e.g., speed, torque, fuel) are polled at least once every few seconds. To compare, polling all the standard PIDs in a round-robin manner retrieves much less information⁴.

To appeal to driving aficionados, a live dashboard in the app displays some of the information from OBD-II (see app screen-shots in Fig. 6).

Bluetooth usage is not a problem. Surprisingly, we found that even when our app is connected to the OBD device, the phone can establish other Bluetooth connections to say headsets or the car speakers. Thus, the app does not disrupt these activities. This is because Bluetooth allows one connection per *profile* at a time; and OBD devices have a profile different from these other devices.

To be widely useful, the app has to satisfy a few constraints. First, it should be able to run continuously in the background and collect data whenever in a moving vehicle. Otherwise, the app will either miss rides or drain the battery. In an earlier version we found that users remembered to turn the app on (or off) less than 20% of the time. Second, the app’s power drain should be an insignificant fraction of the total power draw. Typical users drive a car for less than two hours a day; so two hours of data collection (and 22 inactive hours) should cost say less than 10% of the day’s power draw. Third, to preserve volunteer’s privacy, the app should allow scrub personally identifiable information such as location of homes and destinations.

Energy management. Our app, as implemented, achieves most of the aforementioned goals. The application stays inactive in the background consuming almost zero energy while the phone is stationary. A minimal background task periodically searches for a Bluetooth paired OBD device and wakes up the app when it succeeds. This lets us catch every trip; it is implementable on different platforms unlike the significant location change trigger (only iOS) and is less power hungry than continuously processing the accelerom-

⁴in the information theoretic sense

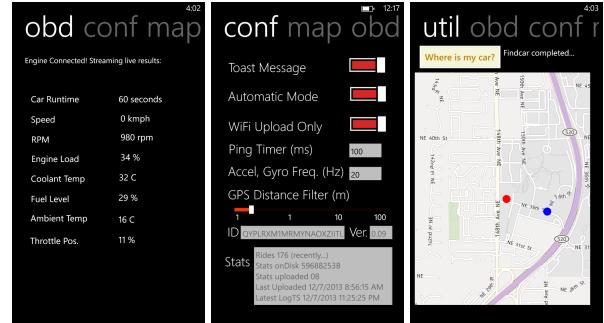


Figure 6: Screen-shots of our app.

eter to detect movement. On windows phone 8.0, the app is ejected after four hours of inactivity; hence, we *toast* the user to restart the app. The app goes back to sleep when it senses that the vehicle is no longer moving (based on a distance traveled and speed check). Across all of the deployed users, the average power draw is 0.4%/ minute when app is awake and 0.01%/ hour otherwise. The app is awake on average for 35 minutes each day. Logs are uploaded to Azure lazily when the device is connected over WiFi and sufficient battery charge remains.

3.3 Cloud services

The Azure based services manage the data collection activities from the users’ smartphones. It is also responsible for map-matching and providing the features for the map-only remapping model (§2.2.2). We use OpenStreetMaps and maps from a proprietary vendor to build these models. We map-match, i.e., from GPS location readings, we identify the most likely path along road segments through an inference algorithm that is similar to Viterbi decoding. Other cloud services include a public-portal for our users where they can view and analyze their driving history.

Building models. The server also coalesces data from multiple users and vehicles to train and build models for Sparc based applications. These models (e.g., for inference remapping) become more accurate as the server gathers higher volume of data from the drivers’ phones and vehicles. In section §5.4.2, we discuss how much training data is necessary. Remapping also enables Sparc to offer value to drivers and vehicles that use the system with just the smartphone application. Till date, we have collected more than 4,400 miles of data from 20 vehicles. The deployment and user study were conducted under purview of Microsoft Research’s privacy policy. We plan to release the Sparc smartphone application in the app store before publication date.

3.4 Beyond fuel prediction

Our phone+car+cloud architecture lets us build a set of diverse applications. We have already implemented a handful in addition to trip analytics, fuel usage prediction and driving feedback. They include a location based real-time traffic alerter (using the Bing Maps API). Also, a "FindMy-Car" button that shows a user where her car is parked. We

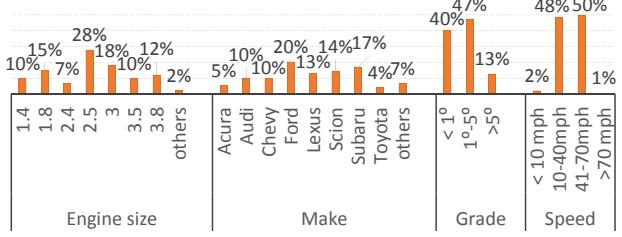


Figure 7: Breakdown of miles traveled in dataset along various aspects.

say that a car is present near the last (first) GPS reading obtained by our smartphone app before (after) it lost access to the OBD device, which happens when the engine turns on->off. We note some limitations for this heuristic; in particular it cannot disambiguate between floors in multi-tiered or underground parking lots. Finally, a website that (1) offers FindMyCar from a browser, (2) visualizes the user’s trips on a map and (3) offers some longitudinal analysis about driving patterns and energy use.

4. DEPLOYMENT & MEASUREMENTS

In this section, we present details about our system deployment. We point out the diversity across cars (makes, years) and roads (re: speed, grade and congestion).

4.1 System Deployment

We deployed our app and OBD devices to twenty volunteers and collected data from Aug. 2013 to Dec. 2014. The volunteers drove a total of 151 hours covering 4423 miles and 15846 unique road segments. The system deployment is under the approval of an internal review board supervised by our organization’s privacy team.

Figure 7 shows a breakdown of the miles traveled by the volunteers in our dataset based on various aspects. We see a wide range of manufacturers and engine sizes varying from small sedans (≤ 2 liter engines) to large SUVs (> 3 liter engines). Roughly half of the miles were from roads with non-trivial banking grade ($\theta > 1^\circ$). Also, roughly half are at speeds below and above 40 miles-per-hour (mph), indicating a mix of highway and surface-road miles. The top fifteen drivers contributed most of the data. Figure 8 visually depicts all the road segments that we have collected data over; the color of a road segment indicates the number of distinct drives through that segment.

4.2 Challenges and opportunities in estimating fuel usage

Predicting fuel usage would be simplest if a vehicle always operated at the same fuel efficiency. Figure 9 shows the per-vehicle distribution of fuel efficiency (MPG). We see that for a majority of cars, their inter-quartile difference is larger than 30% of their average. In fact, for over 70% of the cars, the observed median fuel efficiency was outside the range indicated by EPA’s MPG estimates for city and highway use. The median was away by up to 10 MPG. Note that the range

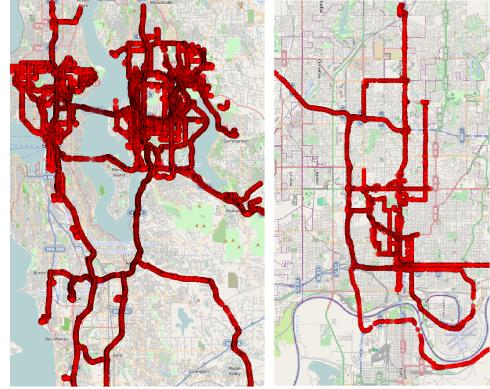


Figure 8: Routes traversed by our drivers in two cities; darkness of the color indicates the number of distinct trips per segment.

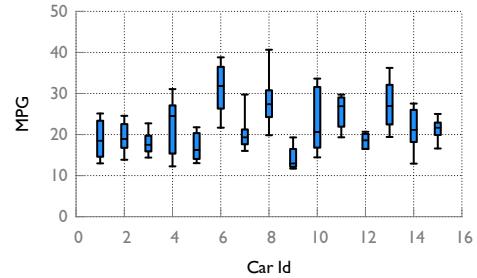


Figure 9: Distribution of fuel efficiency (Miles per Gallon) across vehicles in dataset. The 10th, 25th, 50th, 75th and 90th percentile values are shown.

is quite large to begin with since the MPG estimate for city driving is often 6 – 10 MPG less than the highway estimate. To understand further, Figure 10 plots the fuel used vs. distance traveled in contiguous two minute periods. The points appear to cluster into two groups; those on the right are from faster roads. However, even within each group, there is substantial variation. This means that predictions based on expected MPG are unlikely to be useful.

We next consider trips between the same begin and end locations. When driving within a metropolitan area, there are often a handful of usable routes to take between a given pair of locations. The route that a driver picks impacts fuel use. For the same route, varying congestion levels would impact fuel use. Finally, the variation is likely to be larger when multiple cars are considered, owing to different car types and driving styles. Figure 11 plots the ratio of the fuel used in a trip by the average fuel used by all trips between the same locations. Figure 12 computes the number of rides that are off by more than a given error threshold from the average. We see that even when limited to trips

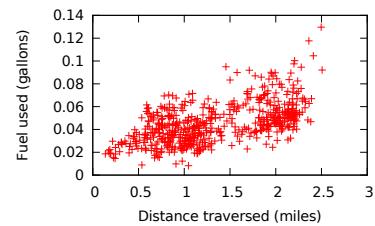


Figure 10: Fuel use vs. distance travelled in a 2 min. period across all cars

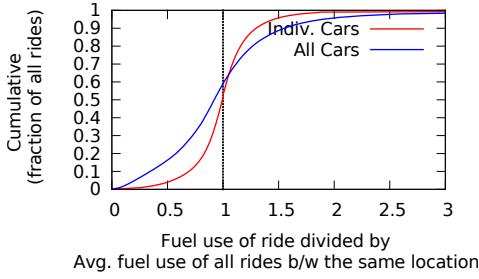


Figure 11: Even trips between the same begin and end points have substantial differences in fuel used

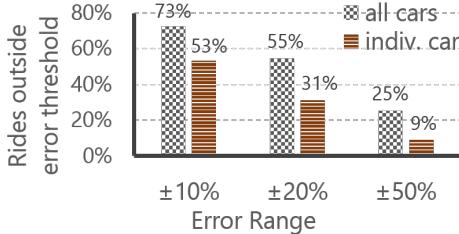


Figure 12: For trips between the same begin and end points, the fraction that are outside a given error threshold from the average

for the same car (and driver), 31% (9%) of trips are off by more than 20% (50%) of the average. The average error only becomes larger when trips from all cars are included. This illustrates both the challenge and the promise: even for the same car and route, fuel use varies substantially; however, if only accurate predictions were available, choosing the best from among the different routes between a pair of locations can reduce fuel use substantially.

5. EVALUATION

In this section, we evaluate the accuracy of our physical model for fuel usage. We show that our physical model that uses data from the OBD port of cars helps us estimate fuel usage with less than 3% average error. Through inference remapping we show that we can use data from phone sensors to estimate fuel usage with less than 6% average error (compared to the ground truth). By remapping to the road-state data, we also show that we can predict fuel use before a trip with $\leq 20\%$ error on the 85th percentile trip. We also describe some interesting feedback on driver behavior.

Qualitative comparison. Before delving into the numbers, we first qualitatively compare Sparc with related research and commercial systems. Table 5 presents this comparison. GreenGPS [16] estimates fuel usage given OBD training data. However, it does not provide feedback re: driving behavior and since it does not remap the roadstate inputs, its error is very high when OBD data is not available. CMT [10] focuses on risk profiling drivers based on smartphone sensors only; it examines aspects such as hard acceleration/ driving above speed limit and offers insights to both drivers and insurance companies. However, CMT does not consider fuel consumption and does not connect with the car. Automatic [6] and Mojio [24] use OBD, an app and cloud

connectivity to provide trip analytics, vehicle diagnostics information and location-related applications. While Sparc offers similar services, our inference remapping deals better with missing data sources (e.g., no OBD device) and hence is likely to be more widely usable.

5.1 Methodology

Dataset: We use the traces collected from unconstrained drivers that we described in §4.1 for this analysis.

Metrics: Per trip, we measure the error in estimating fuel used:

$$\text{relativeerror} = 100 * \frac{\text{estimate} - \text{actual}}{\text{actual}}$$

The estimate can either be generated before the trip was taken, *i.e.*, a prediction, or could be computed post-facto from the sensor readings obtained during the trip. When the sign of the error is not relevant, we show the absolute value of the relative error. Per driver and given a collection of trips, we also compute the contribution to fuel usage due to each of the major *components*: rolling resistance, increasing potential energy, increasing kinetic energy, aerodynamic loss, idling and the others. We also estimate the net positive contribution from the decreases in potential and kinetic energy (e.g., rolling downhill requires less fuel to maintain speed).

Compared alternatives: We have a choice in how the trip features are obtained, how the car-specific parameters are obtained, and how both are combined. In our experimental results, OBD features and Phone features refer to trip features computed based on sensor readings from the corresponding device (see Table 2). Road features refers to trip features computed based on data from the map (see Table 3). Further, OBD parameters and Phone parameters refer to the car parameters that are used in the physical model derived from OBD features (§2.1) and the remapped model based on the Phone features (§2.2.1), respectively. By Stock parameters, we refer to car parameters that are obtained from automobile specifications. Finally, Road→Phone features refer to parameters of the remap model that relies on just map information (§2.2.2).

Not all of the parameter+feature combinations are interesting. We use the combinations that are most relevant per §2.3 for our evaluation.

- SR refers to using the stock car parameter along with the road features in the physical model. Without remapping, this is the best one could do for prediction and post-facto estimates.
- OO refers to using the OBD parameters along with the OBD features in the physical model. We expect this combination to have the smallest error. Fuel usage is directly estimatable given OBD data. So, the value of this datapoint is primarily to check the correctness and completeness of the physical model.
- PP refers to applying the phone parameters along with the phone features in the physical model. Both OO and PP are only usable post-facto, *i.e.*, after the drive, since the features are not available apriori.

Section	Experiment	Summary of Results
Application 1: Post-drive fuel usage prediction §5.2 (Fig 13)	Estimating fuel consumption <i>after</i> a drive using smartphone + OBD model vs. smart-phone only model.	For estimation intervals over 100 sec., (a) average error < 7% using smart-phone only. Using both OBD + smartphone, average estimation error < 2% for 100 sec. intervals and < 6% for 10 sec. intervals.
Application 2: Pre-drive fuel usage prediction §5.3 (Fig 14)	Estimating fuel consumption <i>before</i> a drive.	Using road state model to covert road features into phone features (PhRPh) results in < 20% errors for 48% trips. Better than the baseline (PIR) and other schemes (PhR, OR) which have < 20% errors for 13%, 11% and 12% trips respectively.
Application 3: Analyzing driving behavior §5.4 (Fig 16)	Analyzing driving behavior impact on fuel consumption for the same trip.	Our model identifies driver's impact on different fuel consumption factors (e.g., resuable kinetic loss, using older tires). We identified drivers whose gentle braking behavior enabled reuse of vehicle's kinetic energy.
§5.4.1 (Fig 17)	Per-user long term driving impact on fuel consumption.	Analyzing kinetic energy re-use resulted in identifying drivers with gentle braking and its impact on fuel consumption. Insights about the impact of drivers' most frequent commutes on overall fuel consumption.

Table 4: Summary of evaluation results.

Features	Us	CMT [10]	GreenGPS [16]	Mojio [24]
Analyze driving behavior	✓	✓		
Can drive without OBD	✓	✓	✓	
Analyze fuel consumption	✓		✓	✓

Table 5: Comparing features of Sparc vs. other applications.

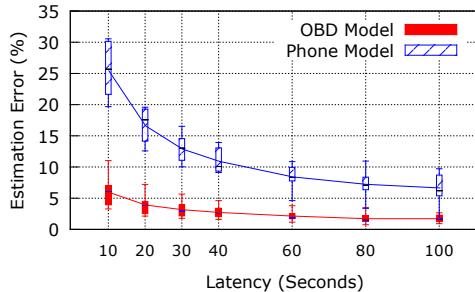


Figure 13: Estimating fuel consumed during a trip using data from (a) the OBD device and phone (2) the phone alone. We can estimate the fuel consumed during a trip with good accuracy using only the phone. The variation shown is across different drivers.

- PRP refers to using the phone parameters and the road→Phone features. This is usable for prediction since it only uses road features at runtime.⁵

Table 4 summarizes our evaluation results.

5.2 Estimating Fuel Use After a Trip

After completing a trip, can we measure the amount of fuel consumed during the trip? This is of some value to a driver; she can estimate how much emissions her driving caused. It may be especially useful when the tank is nearly empty. Even this simple use-case is not possible today without installing an OBD device. Using inference remapping, we can estimate fuel use using data from phone sensors.

Figure 13 plots the absolute value of estimation error *vs.* the duration that the estimate was computed over for both the OBD model (OO) and the Phone model (PP). Per car, we compute the average error over all non-overlapping contiguous traces of a given duration. The error bars show the

⁵ We skip Road→OBD features and OR which applies the OBD parameters on the road features since the results are similar to PRP.

quartiles and min, max across drivers. Most trips last well over 100s; here, PP yields an average error of 7%. All the cars have less than 10% error. Comparatively OO has a much smaller error, 2% at 100s and just 6% for 10s intervals. Recall that if an OBD device is installed, the fuel usage is directly available. Rather, we compute the error of OO to sanity check our models. We conclude that for most trips just the data available from the smartphone suffices to obtain highly accurate estimates.

5.3 Predicting Fuel Use Before a Trip

In this section, we evaluate our ability to predict fuel usage. In our dataset, we observe that a large fraction of the trips that drivers take involve road segments that were not traversed before. Hence, we are more interested in predicting fuel for such cases. Figure 14 depicts the error for a few schemes. Recall that PRP first uses the road-state model to remap road features (from a map) to phone features; to which it then applies the car parameters learnt using training data from the phone. We see that for 49% of the trips, their predictions of fuel usage are within 20% error (gray region in figure). This is significantly better than the results for the baseline (SR) and the other predictive schemes (PR, OR). The fraction of trips that can be predicted to within 20% error are 13%, 11% and 12% respectively for these schemes. We see that about 4× more trips can be predicted to within 20% error by PRP.

It is interesting to note that related pieces of work for fuel-use estimation like GreenGPS [16] learn the vehicular model from OBD data but cannot extract features for unseen roads. Hence, its performance is slightly worse than that shown for OR. Worse because their energy model relies on average velocity rather than piece-wise integrals (§2.2.1). Further they do not separate out the positive and negative parts of changes to potential and kinetic energy which is important as we see next when consider per-component contributions.

Why does using the road features, readily available from the map, lead to such poor predictions? This is because, as we saw in §2, the actual speed for a driver on a road segment can be very different from the posted speed limit; and variations in speed due to acceleration and braking are

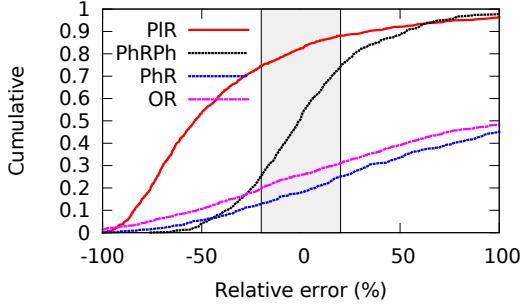


Figure 14: Comparing prediction error across a few schemes. PhRPh significantly out-performs the rest.

not directly estimable from map information. SR, which infers car parameters from automobile specifications has potentially another source of error: the specs. report mass and frontal area but several other parameters are not publicly available per car. For instance, we do not have the aerodrag coefficient per car (c_d), which depends on the specific shape of the vehicle. Similarly unavailable are the efficiency of its transmission shaft in transferring energy from the engine to the wheel (η_t) and its efficiency of burning fuel (η). For these coefficients, we use best guess estimates from mechanical engineering literature. We use $c_d = 0.29$, $\eta_t = .7$, $\eta = .3$, $c_{rr} = .01$. We evaluated an alternative scheme that learns these parameters, *i.e.*, given training data consisting of road features and ground truth fuel usage information, it learns car-specific parameters. Call that the road parameters. Applying these parameters to the road features does slightly better: 21% of trips are within 20% error. We do not show this on the plot for clarity. Finally, the key reason why PRP does better is its ability to learn from crowd-sourced data; a method that estimates aspects relevant to fuel consumption based on easily available features of roads.

Is there room to improve the predictions? Figure 15 compares our best predictive scheme PRP with two alternatives that are not well suited for predictions. $P_{Avg. \ over \ common}$ applies the phone-learnt parameters to the average phone features from other cars that traversed common road segments. In our dataset, while most trips have some common segments, the trips are dominated by road segments not traversed by any other car. So, this scheme has a small coverage (the fraction of road segments that we could predict fuel for). Looking at just the common segments, we see that $P_{Avg. \ over \ common}$ is only marginally better than PRP. We attempted to improve this by only using phone features from similar cars but that further reduced coverage. If much more data from many more cars was available, this scheme may perform better; however, PRP appears to be more suitable when only a little bit of crowdsourced data is available. Finally, we see that the PP is significantly better than PRP: 86% of the trips are within 20% error, indicating that there is room to improve our predictions further. Recall however that PP requires phone features and cannot be used before the drive.

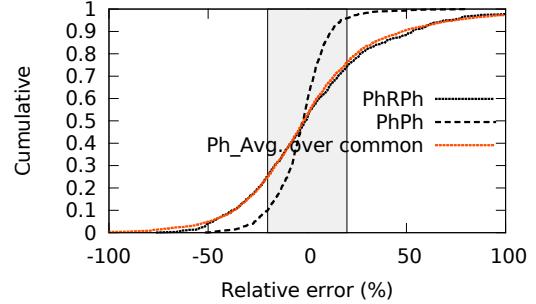


Figure 15: Comparing the prediction error of PhRPh with variants that are not usable for prediction.

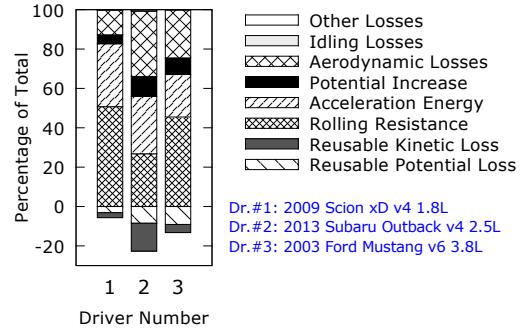


Figure 16: For the same trip, different factors that contribute to the fuel consumption impact drivers differently. Our application can provide such feedback to drivers enabling them to improve their driving habits.

5.4 Providing Feedback on Driver Behavior

Crowd-sourcing data collected from our app allows us to observe how different drivers traverse the same road segments. This lets us compare driving behaviors across drivers. The feedback can help highlight potential poor driving practices or opportunities to reduce emissions as we show below.

To illustrate the value of such analysis, Figure 16 plots the per-component contributions of three drivers. We only use the portion of their trips that traversed road segments that the other drivers also traversed. The results here and in subsequent subsections are from the PP combination unless otherwise noted. We see that driver #2 has high aerodynamic losses. This is expected because his car (listed in the legend) has larger frontal area than the others. Further, we see that driver #2 appears to get more usefulness out of his kinetic energy loss; that is, rather than braking hard, he may be letting his car slow down by counteracting aerodrag and rolling resistance. The other drivers can benefit by less aggressive braking or braking less often. Rolling resistance is large for #1 even though this car weighs the least and should correspondingly have the smallest rolling resistance. Recall that rolling resistance is $mg \int \cos \theta v dt$ and the integral value is essentially the same for a given sequence of road segments since $\int v dt$ is the sum of segment lengths. This hints that driver #1 may be using older tires or may need to optimally inflate her tires.

5.4.1 Long-term Driving Feedback

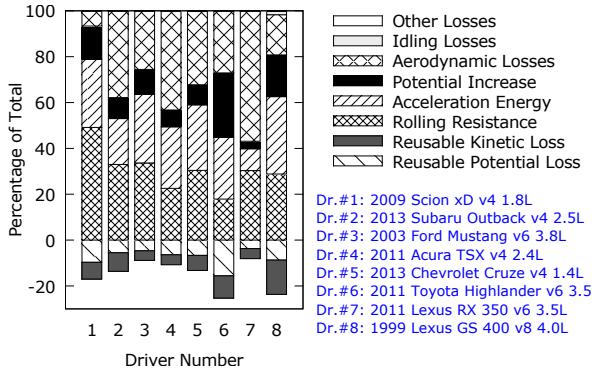


Figure 17: Drivers can also use our application to build up customized long-term driving profiles that average out trip-level dynamics such as traffic, road, and weather conditions.

Analyzing all the data from a given car can reveal further insights specific to driving behavior. Figure 17 plots the per-component contributions for the eight drivers who contributed the most data; we ignore the others for clarity. Consider driver #6, who has a large hybrid SUV. Her daily commute involves climbing a steep hill near her residence; consequently she spends the most fuel in going uphill (potential increase). We see that she is able to make more use of the loss in kinetic energy (KE) because her car explicitly recaptures what would otherwise be lost as heat upon braking to charge the battery instead. Driver #8 does not have a hybrid but appears to be by far the gentlest user of brakes; instead reducing his KE by making it work against the other losses. Consider driver #7, who also has a large SUV but primarily uses it to commute from a suburb to the city on a major highway. We see that rolling resistance and aerodynamic losses dominate; this is expected because most of his driving occurs at higher speeds, involves long distances and his car has a large frontal area. Comparatively, he spends less fuel in increasing KE (acceleration energy), hinting that most of his drives are at relatively steady velocity. Not much change in PE either, because his trips, in the midwest, are on flat roads. Consider driver #4, who has a smaller wagon and also mostly commutes on a congested highway. His component breakdown is similar to that of driver #7 except for a larger contribution due to increasing KE. Perhaps congestion causes him to change speed often. In contrast driver #1 uses a sub-compact for a long commute and some errands on surface streets. We see that rolling resistance is dominant for her; aerodrag is small due to the slower speeds and small frontal area.

5.4.2 How Much Training Data Do We Need?

To answer this question, we vary the sizes of training data. Per size, we pick a random subset to be the training data, learn the car-specific parameters from this set and apply these parameters to the rest of the data to estimate per-component contributions. We repeat this for 100 random subsets. When training is done on *enough* data, we would expect that the parameters learnt from the different training

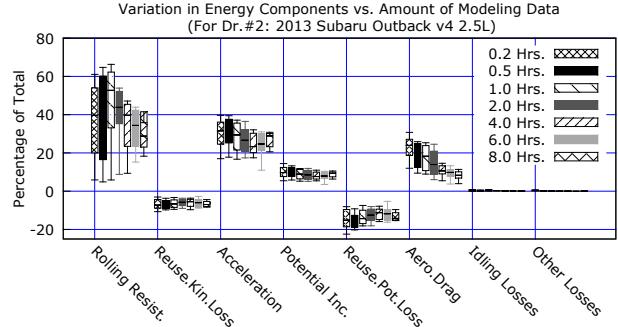


Figure 18: For driver 2, to build a reliable long-term profile, we need to collect about four hours of data and build the regression model.

subsets to be similar, i.e., they report similar per-component contributions. However, when trained on too little data, the per-component contributions could be very different.

Figure 18 plots the quartiles, min and max of the per-component contributions for driver #2 given different amounts of training data. We report results for this driver because he had the most data; however other drivers yielded similar results. We see that some components can be estimated correctly with fewer data than others. When too little data is used, most components exhibit variability. Because, the model may be influenced by the specific roads and traffic conditions present in the training data. Such models are likely to have little predictive value. However when more than four hours of data is used for training we find that most components are stable. These training sets are perhaps large enough to be representative of typical driving conditions. This leads us to conclude that about four hours of training data should suffice for most drivers.

6 RELATED WORK

A driver cares about two aspects of vehicular fuel use: (1) how much fuel would a trip use and (2) what factors impact fuel efficiency?

The conventional metric for fuel efficiency in the United States (US) is miles per gallon (MPG). While MPG is adequate to compare vehicles, we saw that it neither helps predict fuel use on a trip nor explains the factors impacting fuel efficiency [21]. The environmental protection agency (EPA) is tasked with determining MPG estimates and publishes an annual document outlining its methodology [5]. In response to widespread criticism—estimates lacked real-world testing, and were of very limited scale *i.e.*, city or highway—EPA updated its rating system in 2008. The new system considers things like faster speeds, acceleration, air conditioner use, and colder outer temperatures [13]. While an improvement, they still do not suffice for the above goals. In particular, MPG estimates are inadequate at capturing the variable traffic and road conditions [2, 14, 4]. To remedy this, much focus has gone into gathering and collecting real-world user data re: fuel efficiency [12, 27]. Unfortunately, the user-reported numbers exhibit substantial variability [12]

and lack the context that may help explain the variability.

Concluding that static MPG estimates are unreliable, focus has shifted towards dynamic models of fuel efficiency. One class of work empirically determines an MPG estimate per driving regime such as constant speed, high acceleration, peak-traffic times, highway or city *etc.* [9, 7, 17]. While more accurate, these estimates do not explain factors that impact fuel economy nor can they predict fuel consumption accurately. Another class of work develops dynamic fuel-estimation models using several parameters. Some of these approaches require elaborate instrumentation to measure parameters such as exhaust-gas composition and engine-cylinder displacement [1]. The more practical approaches use OBD information available in modern automobiles [3, 16, 15]. These methods collect OBD-data from individual drivers and build fuel-estimation models. GreenGPS [16] is the best example here.

However, these approaches still have a few drawbacks. First, they can analyze fuel use post-facto but cannot predict fuel use before a trip, especially if the road segments have not been driven on before. Second, lacking the ability to extrapolate, they require drivers to continually use an OBD device. Third, when using the models to extract per-component contributions, we find that errors in the models (*e.g.*, using average values, not separating loss of energy terms) leads to mis-attributions. The approach presented here addresses these shortcomings. It allows users with a smartphone device to obtain accurate estimates of MPG values and offers insight into the factors affecting fuel economy. And, it can predict fuel use before a trip occurs; fuel-prediction also distinguishes us from other mobile participatory sensing systems that use smartphone devices to generate traffic advisories [20, 19] or manage parking [11].

7. CONCLUSION

We describe a phone+car+cloud system that has the potential to transform many vehicular use cases. Our core technical contribution is the concept of inference remapping which allows us to compute inferences even when the ideal inputs are missing or impossible to obtain *a priori*. For *e.g.*, given readings from a smartphone, we can estimate fuel used by a vehicle. Further, given some crowd-sourced data, we can predict fuel used along a route for whom no data readings have been collected. At first blush, both seem impossible. Yet, remapping makes this possible. Because, we are able to exploit underlying correlations between readily available information (*e.g.*, maps), training data, and the desired information that captures vehicular fuel use (*e.g.*, dynamic changes in speed, stop durations, car parameters etc.) Careful engineering was required to compose disparate time-series and to not overwhelm phone battery. Much work remains, in particular, Sparc's predictions remain 1.8X away from the post facto estimates. Also, whether remapping is broadly applicable remains an open question.

8. REFERENCES

- [1] K. Ahn. *Microscopic fuel consumption and emission modeling*. PhD thesis, Virginia Polytech. Inst. and State Univ., 1998.
- [2] W. M. Al-Momani and O. O. Badran. Experimental investigation of factors affecting vehicle fuel consumption. *Int. J. Mech. and Materials Eng.*, 2007.
- [3] F. An and M. Ross. Model of fuel economy with applications to driving cycles and traffic management. *Transportation Research Record*, 1993.
- [4] S. T. Anderson et al. Automobile fuel economy standards: Impacts, efficiency, and alternatives. *Rev. Environmental Economics and Policy*, 2010.
- [5] A. Atabani et al. A review on global fuel economy standards, labels and technologies in the transportation sector. *Renewable and Sustainable Energy Rev.*, 2011.
- [6] Automatic. Available Online at: <https://www.automatic.com/>.
- [7] J. Bandeira et al. A comparative empirical analysis of eco-friendly routes during peak and off-peak hours. In *Ann. Meet. Transportation Research Board*, 2012.
- [8] C. Baumgarten. *Mixture formation in internal combustion engines*. 2006.
- [9] M. Ben-Chaim, E. Shmerling, and A. Kuperman. Analytic modeling of vehicle fuel consumption. *Energies*, 2013.
- [10] Cambridge Mobile Telematics. Available Online at: <http://www.cmtelematics.com>.
- [11] V. Coric and M. Gruteser. Crowdsensing maps of on-street parking spaces. In *DCSS*, 2013.
- [12] EPA shared MPG estimates. Available Online at: <http://www.fueleconomy.gov/mpg/MPG.do?action=browseList>.
- [13] EPA fuel economy guide. Available Online at: <http://www.fueleconomy.gov/feg/printGuides.shtml>.
- [14] E. Ericsson. Variability in urban driving patterns. *Transportation Research Part D: Transport and Environment*, 2000.
- [15] Fuely. Available Online at: <https://www.fuely.com/>.
- [16] R. K. Ganti et al. GreenGPS: A participatory sensing fuel-efficient maps application. In *MobiSys*, 2010.
- [17] Gas buddy. Available Online at: <http://www.gasbuddy.com/>.
- [18] J. S. Greenfeld. Matching gps observations to locations on a digital map. In *Ann. Meet. Transportation Research Board*, 2002.
- [19] S. Hu et al. Poster abstract: Smartroad: A crowd-sourced traffic regulator detection and identification system. In *IPSN*, 2013.
- [20] E. Koukoumidis et al. SignalGuru: Leveraging mobile phones for collaborative traffic signal schedule advisory. In *MobiSys*, 2011.
- [21] R. P. Larrick and J. B. Soll. The MPG illusion. *Science* 20, 2008.
- [22] A. Mai and D. Schlesinger. A business case for

- connecting vehicles executive summary. CISCO Internet Business Solutions Group Report, Apr. 2011.
- [23] Metromile. Available Online at: <https://www.metromile.com/>.
 - [24] Mojio. Available Online at: <https://www.moj.io/>.
 - [25] S. Nath. ACE: exploiting correlation for energy-efficient and continuous context sensing. In *MobiSys*, 2012.
 - [26] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Mobisys*, 2010.
 - [27] M. Satyanarayanan. Mobile computing: The next decade. *ACM Mobile Comp. and Comms. Rev.*, 2011.