Participatory Classification in a System for Assessing Multimodal Transportation Patterns

Kalyanaraman Shankari
Mogeng Yin
Randy H. Katz
David E. Culler

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-8
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-8.html

February 17, 2015

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
15 FEB 2015		**00-00-2015 to 00-00-2015**

4. TITLE AND SUBTITLE
Participatory Classification in a System for Assessing Multimodal Transportation Patterns

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of California at Berkeley,Electrical Engineering and Computer Sciences,Berkeley,CA,94720

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
There has been an increasing trend of performing inference on data collected by smartphones to provide context-aware location-based services. When this inference is performed using supervised analysis, these services need ground truth if high accuracies are desired. While accuracy is less of a concern for services targeted at individuals, it is important when individual data is aggregated for semantic analysis of a population. However, traditional techniques for obtaining ground truth such as paid crowdsourcing are challenging in this domain since the ground truth is uniquely available to the user. Therefore, the user needs to be the source of ground truth for these services. This motivates the need for Participatory Classification, a framework that is able to satisfy the need for minimally invasive ongoing, ground truth collection from regular users at scale. We present an architecture that can be used to enable this framework for such services, and evaluate the framework in the context of an end-to-end prototype that we built. The prototype minimizes the burden on the user while classifying trips by travel mode, and uses the classified trips to generate a personalized carbon footprint for the user and aggregate data such as commute mode share, for use by urban planners. With this prototype, we collected 7439 labelled sections from 44 unpaid volunteers over a total period of 3 months.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**15**	

Acknowledgement

This work was supported in part by the Jim Gray Fellowship, in part by the NSF ActionWebs CPS-0931843, in part by NSF CISE Expeditions Award CCF-1139158, LBNL Award 7076018, DARPA XData Award FA8750-12-2-0331, and gifts from Amazon Web Services, Google, SAP, The Thomas and Stacey Siebel Foundation, Adobe, Apple, Inc., Bosch, C3Energy, Cisco, Cloudera, EMC, Ericsson, Facebook, GameOnTalis, Guavus, HP, Huawei, Intel, Microsoft, NetApp, Pivotal, Splunk, Virdata, VMware, and Yahoo!.
We would like to thank Akshay Vij for suggesting approaches for automated inference and for his insight into prompted recall. We would also like to thank Shanthi Shanmugam for all her help with the design,

and for the initial MovesConnect OAuth implementation. And to Thomas Raffill, for all the reviews.

Participatory Classification in a System for Assessing Multimodal Transportation Patterns

Kalyanaraman Shankari
EECS Department
UC Berkeley
shankari@eecs.berkeley.edu

David Culler
EECS Department
UC Berkeley
culler@cs.berkeley.edu

Randy Katz
EECS Department
UC Berkeley
randy@cs.berkeley.edu

Mogeng Yin
Institute of Transportation Studies
UC Berkeley
mogengyin@berkeley.edu

ABSTRACT

There has been an increasing trend of performing inference on data collected by smartphones to provide context-aware location-based services. When this inference is performed using supervised analysis, these services need ground truth if high accuracies are desired. While accuracy is less of a concern for services targeted at individuals, it is important when individual data is aggregated for semantic analysis of a population. However, traditional techniques for obtaining ground truth such as paid crowdsourcing are challenging in this domain since the ground truth is uniquely available to the user. Therefore, the user needs to be the source of ground truth for these services.

This motivates the need for Participatory Classification, a framework that is able to satisfy the need for minimally invasive, ongoing, ground truth collection from regular users at scale. We present an architecture that can be used to enable this framework for such services, and evaluate the framework in the context of an end-to-end prototype that we built. The prototype minimizes the burden on the user while classifying trips by travel mode, and uses the classified trips to generate a personalized carbon footprint for the user and aggregate data such as commute mode share, for use by urban planners. With this prototype, we collected 7439 labelled sections from 44 unpaid volunteers over a total period of 3 months.

1. INTRODUCTION

As mobile systems using smartphone technology have matured, we have seen the emergence of two distinct and complementary fields of study - in **participatory sensing**, observations from a large number of lay users are aggregated to map environmental parameters (e.g. air quality and potholes) for urban regions and in **individual activity classification**, an individual user's activities (walk, bike and drive, sleeping, eating, etc.) are inferred by extracting semantic analysis from smartphone sensors. The main difference between the two fields is in their focus - the first is at a societal level and the second is at the individual level.

In this work, we are interested in trip mode classification across a broad population in order to use the **aggregate observations** to improve the transportation system at urban scale, similar to participatory sensing; as well as to provide a personalized record of the user's **individual travel mode** history and carbon footprint, which is similar to individual activity classification.

We are interested in accurate classification from many individuals with the least burden on the user in order to maximize participation. We investigate how to engage the individual effectively in performing classifications that can both drive the learning process and provide enough accuracy to generate a reliable aggregate picture. Therefore, we combine the two fields and develop a framework for "participatory classification" in the context of sustainable land use and transportation planning.

Practical considerations typically dictate the following procedure for activity classification: 1) an initial phase of supervised learning (training), 2) a static set of classifiers, and 3) ongoing collection of unlabelled data only for prediction (e.g. [10], or [14]).

However, this method presents several shortcomings when it comes to model estimation accuracy in our domain: 1. *Variability across users is hard to reconcile into a single generic model* for example, different people bicycle at different speeds, 2. *Context sensitive classifiers are needed when sensing data is insufficient for disambiguation* for example, sensing data cannot distinguish carpools from single occupant vehicles, and 3. *The same user might have different characteristics at different times* for example, a user may ride at her child's pace while dropping off at school.

The development of user-specific, context-sensitive classifiers for basic activity sensing ([15], [16]) has addressed these problems, but has introduced a new requirement for user-specific ground truth to train these user-specific classifiers.

In addition, if applications are purely targeted towards providing user level feedback, their accuracy is less important since the user is the consumer of the information, and can ignore errors while making changes. But, intuitively, having high accuracy data is important at the aggregate level to avoid compounding of individual errors. This is particularly important if the aggregate data is to be used for semantic analysis, for example,[22] shows that low accuracy rates can introduce significant bias if the detected trips are used for travel demand models. Collecting ground truth on an ongoing basis can increase the accuracy of the data available for aggregation.

Finally, cold start is a potential problem on new users. The aggregated classifiers can be used to bootstrap at the initial stage of the classification.

However, ongoing collection of user-specific ground truth is subject to the following constraints. a) labels can only be assigned by the user. One standard technique to generate labels for large amounts of unlabelled data is to use crowdsourcing by paid humans who cannot access the ground truth. However, that technique cannot be used to accurately infer user intent. b) assignment of labels can cause a substantial burden, which should be mitigated. This means that techniques that require the user to manually trigger the entry of ground truth, or to visit a website later, are not sustainable at large scale. So the primary challenge for participatory classification is that of minimally invasive ground truthing at scale.

1.1 Contributions

Our contributions are all related to addressing the challenge above.

1. We introduce the notion of Participatory Classification, which is a framework to explore ideas around engaging the individual effectively in performing classifications that can both drive the learning process and provide enough accuracy to generate a reliable aggregate picture.
2. We explore the use of prompted recall directly on the smartphone to collect ground truth for a large number of trips, highlighting low confidence trips to reduce the burden on the user. While there have been prior projects that have collected large-scale GPS traces, and prior projects that have worked on activity classification, their traces were collected without ground truth, and/or the participants were compensated. We have built an end-to-end prototype with apps in both the android and iPhone stores and have deployed it to collect 7439 labelled trips from 44 unpaid volunteers for a period of roughly 3 months.
3. We aggregate individual user information to perform aggregate analysis (e.g. heatmaps, arrival times at work).

The paper outline is as follows: in section 2, we compare our solution to related work, sections 3, 4 and 5 describe the system architecture, functionality and design choices, section 6 is a brief evaluation, section 7 outlines the future work, and section 8.

2. RELATED WORK

The related work falls into 4 main categories, each of which is described below. We focus on the individual activity classification category, which is closest to our domain. A visual representation of the space is shown in 1.

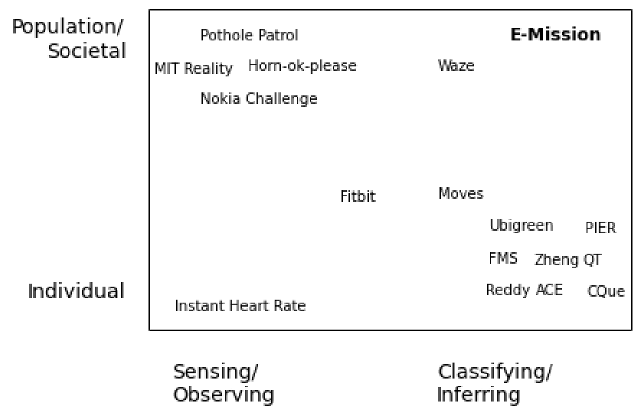

Figure 1: Space of related work

2.1 Individual sensing

This includes display of data about a single individual, for e.g. the Instant Heart Rate Sensing app [1].

2.2 Participatory sensing

We think that large scale datasets such as [4] and [11] that contain unlabelled GPS data, fall into this category since they can be used to generate heatmaps of human activity. This also includes citizen science or urban sensing applications such as [5] and [20].

2.3 Individual activity classification

Since this section is most related to our application domain, we provide a brief review of papers around individual activity classification in Table 1. Note that the table does not include commercial applications such as Waze [23] or Moves [13], which are very similar to our work, but are closed-source commercial applications whose architecture and evaluation is unknown. A short feature comparison is provided here instead.

Waze Waze uses participatory sensing to determine traffic flow, but it only works for automobiles and does not perform any mode classification. It does allow users to provide information about incidents, but the incidents are user reported and not learned from sensed data.

Moves We use moves for our data sensing and initial classification. However, it is an individual activity classifier since it does not provide aggregate results. Further, it does not distinguish between motorized modes.

To summarize, we distinguish ourselves from the related work because:

Recall We allow users to correct our classifications by prompting them to confirm their trip modes directly on the phone. This has allowed us to build a large set of GPS traces labelled by user confirmed transportation mode.

Aggregate We aggregate individual user data in order to obtain an aggregate overview of temporal information, such as the distribution of arrival and departure times at work, and spatial information, such as the most popular bike and car routes.

Sensors We perform the tracking using GPS data collected from smartphone sensors at relatively coarse granularity, instead of a separate GPS device with fine granularity.

Modes We automatically distinguish between motorized modes (car, bus, train, air) in addition to non-motorized modes such as walking and cycling.

Carbon We provide users with their personalized, automatically detected transportation carbon footprint, and compare it to their peers and emission reduction goals.

3. SYSTEM ARCHITECTURE

The system architecture diagram is shown in Figure 2. The various components are briefly described below. The glossary (Sec. 3.1) might be useful. The areas with a dark background are currently implemented, while those with a light background are planned for the future.

This is a distributed system in which three sets of data are exchanged via three independent sync mechanisms (**Trips**, **Incidents** and **Results/Incentives**). We have chosen one-way sync as the data transfer technique since our requirements are for timely, but not real-time communication, sync transfers allow power efficient scheduling [24] and are robust to connectivity issues in the mobile environment. The three flows are largely independent, but come together to inform two sets of analyses - **mode classification/inference of trips**, and **generation of information/results** for the user.

Each flow has corresponding databases on each side, and data flows through the database based on object state. We illustrate this with the example of the trip flow, and how it enables participatory classification.

1. Trips that are sensed on the phone have an *preliminary*, coarse classification (walk/bike/transport only). This also allows for more power efficient trip location sensing. Completed trips with a preliminary classification are stored in the *queued* table until the next sync.
2. The next sync pushes the newly *sensed trip* to the server, where it is stored in the unestimated table.
3. The next time the classifier is run, it generates a *proposed* mode for the trip, and the trip is moved to the *proposed* table.
4. The next sync pushes the proposed trip to phone where it is stored in the unconfirmed trip table.
5. The next time the user launches the app, the trip is displayed as part of the confirmation screen, and is confirmed by the user. The trip then moves to the *confirmed* table.
6. The next sync moves the trip back to the server, where it is stored in the confirmed table and can be used in classifying other incoming trips, and in other analytics.

3.1 Glossary

1. **Trips and sections:** The data received from the phone is pre-segmented into trips, each of which consists of one or more sections. A trip is a logical transition from one location to another, and may consist of multiple sections. For each section, we receive the start time, end time, GPS tracking points approximately every 30 seconds if there is signal, and a coarse, preliminary inference of the mode.

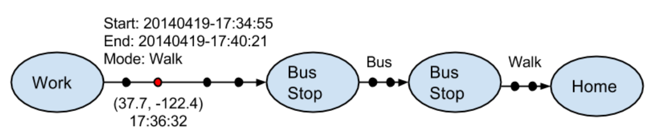

Figure 3: Examples of a trips and its sections

2. **Unclassified sections:** Trip sections that were detected using phone sensors but have not yet been confirmed by the user.
3. **Classified sections:** Trip sections that have been displayed to the user and confirmed as accurate or inaccurate.
4. **Predicted mode:** Mode predicted by our inference algorithm.
5. **Confirmed mode:** Mode confirmed by the user.

4. SYSTEM FUNCTIONALITY

We have built an end-to-end system prototype, with apps in both the android and iOS app stores, that we

Table 1: **Summary of individual activity classification focusing on features relevant to this paper**

	UbiGreen [8]	PIER [14]	Reddy (2010) [19]	Zheng (2010) [25]	FMS [2]	PhD Thesis [9]	QT [10]	CQue[16]	ACE[15]	E-Mission
Device	separate	phone	phone	separate	phone	phone	phone	phone	phone	phone
Modes	walk + cycle + transport	walk + cycle + transport	walk + cycle + transport	walk + cycle + bus + car	walk + cycle + bus + subway + motorbike + car	walk + cycle + transit + car	walk + cycle + bus + train + car	walk + drive + still + with friends + home + work	walk + cycle + drive + at home + in office	walk + cycle + bus + train + car + air
Recall	on phone	python script	offline	web	web	manual	web, optional	phone??	user initiated on phone	phone
Carbon	green/non green	Estimate from CARB	N	N	N	N	Y	N	N	Estimate from [12]
Upload	N	auto w/ manual trigger	manual	unspecified	auto w/ manual trigger	manual	auto	N/A	N/A	auto
Collection with ground truth	14, 1-4 wks	5, 1 day	16 for 7.50 hrs,1 for 4 weeks,16 for 1 day	65, 3 mos	27, 5 days	6	not reported	7, 2 weeks	10, 2 weeks	60, 6 mos
Collection without ground truth	N/A	30, 6 mos	N/A	N/A	34, 2 weeks	6	135, 3 weeks	35, > 10 wks	95, > 2 weeks	N/A
Notes on ground truth	auto sense, trigger, or manual enter	ground truth only for "driving" mode	collection over 4 weeks was "when possible"	...	Target metric was validating 50% of the data	Ground truth was reported in person to researcher	not reported	Unlabelled data was from reality mining dataset	Unlabelled data was from reality mining dataset	Reported using phone
Accuracy	not reported	30% - 90% for drive only	88% - 96% (overall)	61% - 83%	not reported	80% - 90%	not reported	75 - 99%	not reported	62% - 95%
Compensation	$100 to $300	Researchers	Unsure	based on labelled distance	SGD 30 (USD 25)	Researchers	$15/hr	Researchers	Researchers	non researchers, no money

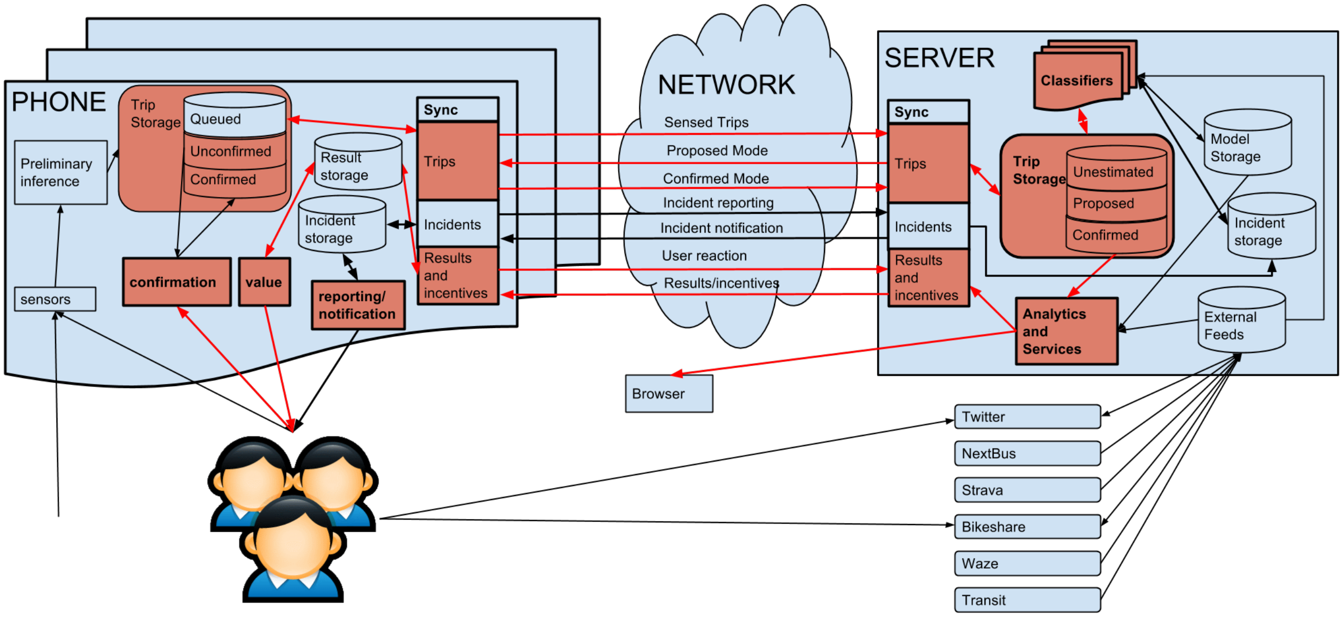

Figure 2: System architecture

have used to explore ideas around participatory classification. This section describes the prototype functionality, and Section 5 describes the design choices behind the prototype.

The prototype has three main components, two of which are user visible.

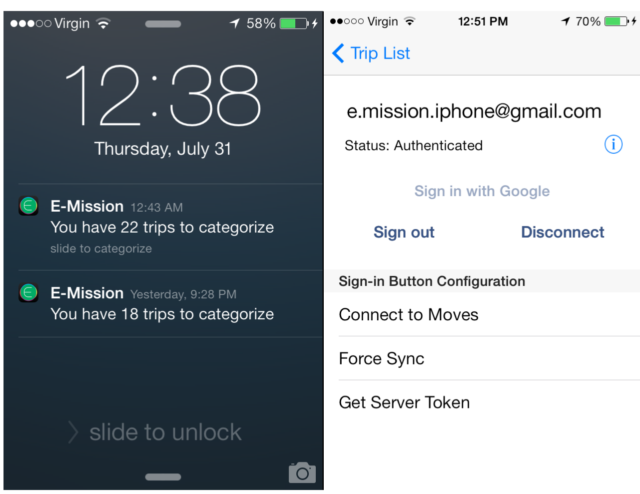

Figure 4: Example of trip notification and authentication screen

4.1 Phone app

We have developed phone apps for both the android and iPhone platforms. These are available for general install using the app stores and have been designed so that no interaction with the researchers is necessary for install and ongoing use. The apps have 4 main functions.

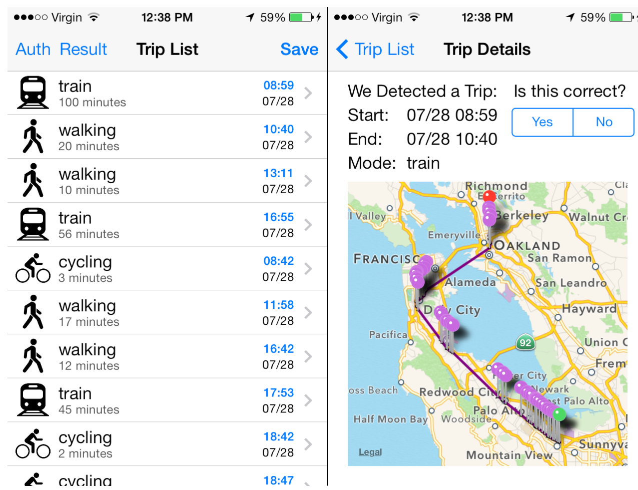

Figure 5: Sample list of trip sections and the detail of one section showing the route taken

1. Display a set of onboarding screens that describe the system and obtain consent from the user.
2. Obtain authentication to access to the data collected by the Moves app installed on the same phone (Fig. 4).
3. Display a notification prompting the user to classify all the unconfirmed trips (Fig. 4) from the past week. Since we use a sync mechanism in the trip confirmation flow, the notification does not need to be responded to immediately, and unconfirmed trips will simply be included in the next notification. We can see this in Fig. 4, where the 18 trips at 9pm have not been confirmed, and are included

in the notification at midnight.

4. When the app is launched, display a list of unconfirmed trips and allow the user to confirm them (Fig. 5).

4.2 Web app

The web app is responsible for exposing a REST API that provides access to the data in several forms. It is a fairly lightweight process that primarily reads data directly from a MongoDB instance and does not perform significant postprocessing. A complete list of the current API methods is provided in Table 2. In addition, the webapp exposes a visualization UI for the aggregate functions that is built using Javascript and NVD3 [17], a lightweight wrapper on top of D3, invoking the REST API for the data. Selected screenshots of the web UI are shown in Figure 6.

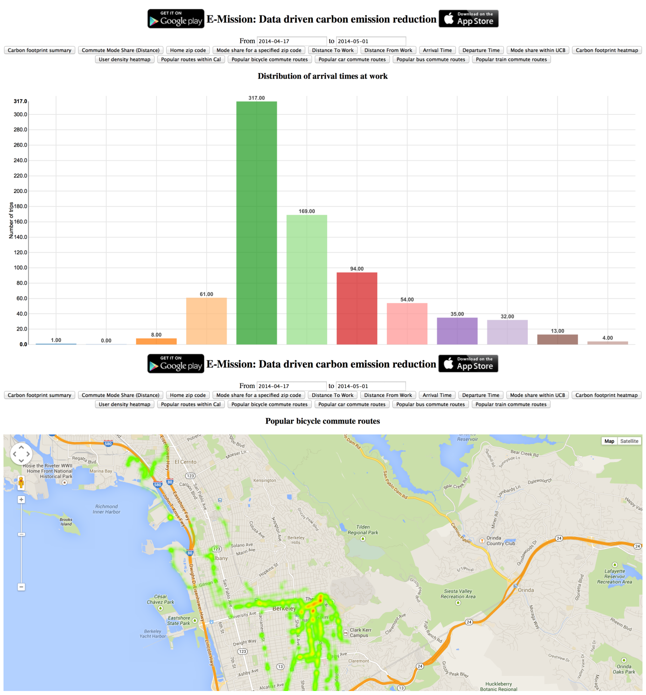

Figure 6: Arrival times at work and popular bicycle commute routes during the last two weeks of Apr 2014 (2014-04-17 to 2014-05-01)

4.3 Analysis

To have a responsive interface, we perform the bulk of the processing offline in batch mode. The results of the offline processing are stored in the database for easy access by the webapp layer. We sketch the algorithms used in the analysis here - a more detailed description is available in the Technical Report [21].

1. **GPS trace retrieval** We currently read GPS traces using the Moves app, which also conveniently breaks up the traces into trips and sections. As we integrate with other sources, we may need to incorporate trip detection algorithms here as well.
2. **Home and work location** Once we have the raw trip sections for each user, we detect home and work locations automatically. We make the assumption that the first trip section made after 5 am each day has a high probability of originating from home. We define the place that a user spends most of the time in a day (except home) as his/her work location.
3. **Commute mode sections** To support statistics on commute behaviour such as the arrival time at work, we classify trip sections as commute and non-commute. For the "to" commute trip, we do this by finding the first trip segment that a user made after 5am from home, and iterating over subsequent trip segments until we find one that ends at work. We use a similar heuristic for the "from" commute trip. All other trips are labelled "non-commute".
4. **Mode inference** We use several features generated from the GPS data in order to generate a predicted mode classification for unclassofied trips. This includes not just non-motorized modes such as walk and bike, but also, uniquely, motorized modes such as car, bus, train and air. We originally attempted to use the general (G) and advanced (A) features from [25]. Since our readings were obtained from smartphones, their granularity was coarse, the feature calculations were not very accurate, and the resulting accuracies, specially for motorized transport, were low. In response, we added the following spatiotemporal features, which allowed us to increase the accuracy for motorized modes by around 30%.
   (a) **Bus/Train (B):** Determine bus and train station locations by looking at the start and end points of bus and train trip sections, and using the DBSCAN [6] from the `scikit-learn` library [18] algorithm to cluster them into stations.
   (b) **Location (L):** Add the lat/lng coordinates of the start and end points of the trip sections.
   (c) **Time (T):** Add the hour of the trip as a feature. This allows us to avoid overfitting with the location parameter.

5. SYSTEM DESIGN CHOICES

This section explores the design choices and lessons learned from the development of the prototype. It focuses on design choices that are key to the framework of pervasive classification - the interested reader is referred to the associated Tech Report [21] for additional details.

To recap, our primary challenge is that of minimally

Table 2: List of current API methods

API name	PII?	Method	Description
`/result/commute.modeshare.distance`	N	GET	Distance travelled by each mode in commute trips
`/result/internal.modeshare.distance`	N	GET	Distance travelled by each mode inside the UC Berkeley campus
`/result/commute.modeshare/zipcode/`zc	N	GET	Number of trips in each mode for a particular zip code
`/result/commute.distance.to`	N	GET	Distance travelled during commute *to* work
`/result/commute.distance.from`	N	GET	Distance travelled during commute *from* work
`/result/commute.arrivalTime`	N	GET	Time at which users arrived at work
`/result/commute.departureTime`	N	GET	Time at which users left work
`/result/heatmap/carbon`	N	GET	Carbon intensity of various zip codes
`/result/heatmap/pop.route/cal`	N	GET	Popular routes within the UC Berkeley campus
`/result/heatmap/pop.route/commute/`$selMode$	N	GET	Popular routes for a particular commute mode
`/result/carbon/all/summary`	N	GET	Aggregate transportation carbon footprint
`/tripManager/getUnclassifiedSections`	Y	POST	The list of sections that a user needs to classify
`/tripManager/setSectionClassification`	Y	POST	User confirmed ground truth
`/compare`	Y	POST	The personalized carbon footprint for a particular user
`/movesCallback`	Y	POST	Moves auth code that is exchanged for an access token

invasive ground truthing at scale. The design choices that we used to address that challenge fall into three main categories.

5.1 Motivate users

In order to motivate unpaid volunteers to give us ground truth, we used the following techniques from behavioral economics [7] and mapped them to our app as follows:

1. **Trigger:** The trigger is an internal (*That's cool!* $\rightarrow$ *let me launch instagram*) or external (*You've got mail!*) event that catches the user's attention. Our app has an external trigger - it uses the smartphone notification mechanism to prompt users with trips to confirm.
2. **Action:** The trigger functions as a reminder to perform an action. In our case, the action that we want is for the user to confirm their trips.
3. **Variable Reward:** Since the user just completed a task for us, we need to offer her a reward. Offering a monetary reward is not scalable for ongoing data collection, so we offer information. We display the user's carbon footprint (Figure 7), and comparisons to both the average of other users, and to an optimal value. This makes the information both personalized and actionable. Since this information is refreshed based on the confirmed data, it is constantly changing, and the variable reward increases engagement. This is clearly a reward that primarily appeals to environmentally conscious users, and we are exploring other reward techniques in our ongoing work.
4. **Investment:** In our case, the confirmation is the only investment we require. This is an area for future improvement to further increase user engagement.

5.2 Reduce burden

Participatory Classification techniques need to be minimally invasive, since the motivation is also slight. Here are some techniques that we used to reduce the user burden.

1. the app is publicly available in the standard app stores. In our experience, this is critical for widespread adoption. Before we put the apps in the public stores, we found that it was challenging to get non-technical users to install the apps - installing apks on android was cumbersome, but the process to install beta test iOS apps was so onerous that we ended up physically connecting user phones to our laptops for the install.
2. the trips are confirmed directly on the phone. In our experience (Section 6.1), even with prompting on the phone, user engagement reduces over time. Given the higher cognitive load to remember to access a website using a browser without prompting, our intuition is that browser based confirmation methods will see an even sharper drop off.
3. we use the mode inference algorithm from Section 4.3 to pre-populate a predicted mode, so that the user can typically confirm with one click.

In spite of these efforts, we saw a drop-off in the percentage of trips confirmed over time, which we have tried to address through reducing the effort for confirmation (see Section 6.1).

5.3 Consider privacy

Since our data is privacy sensitive, we have classified the methods that expose it into two groups - ones that expose Personally Identifiable Information (PII) and ones that don't. As we can see from Table 2, all methods that expose PII are HTTP POST methods, and require a JSON Web Token (JWT) for authentication. These are currently accessed from the phone apps, where we generate the JWT by authenticating with Google.

We perform two levels of authentication. We use OAuth to authenticate the user account. This allows the same user to access their data from multiple devices. We also use OAuth to authenticate with our GPS trace provider (Moves) - this gives us the permission to read the list of trips and sections that they have collected.

6. EXPERIMENTAL RESULTS

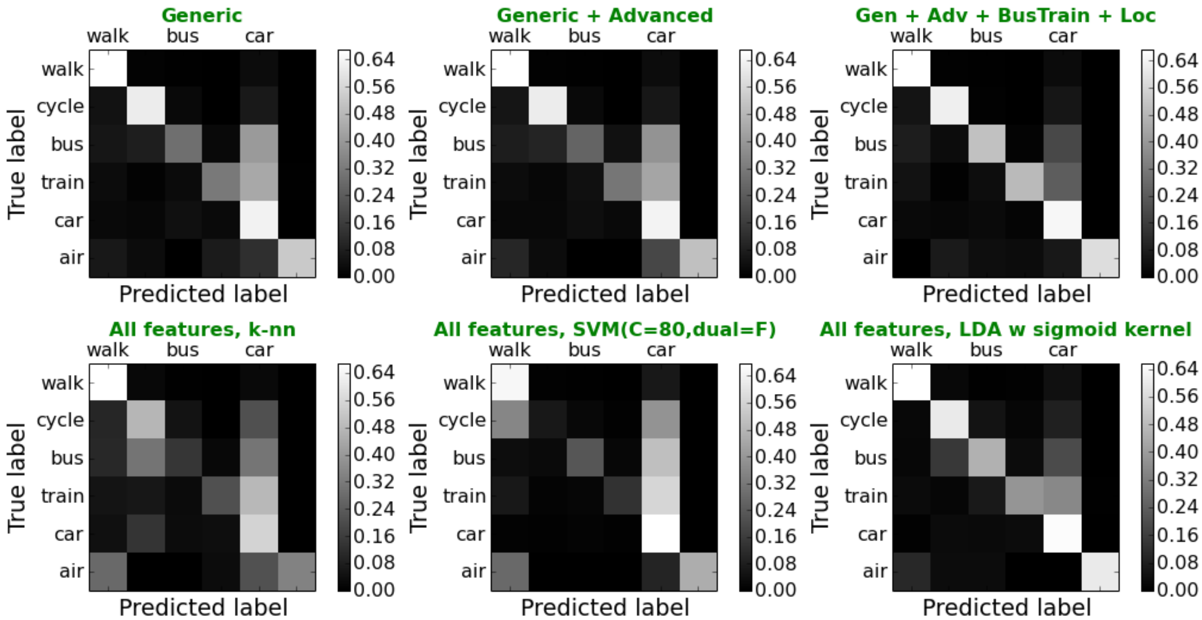

Figure 8: Confusion matrices for different combinations of features and models

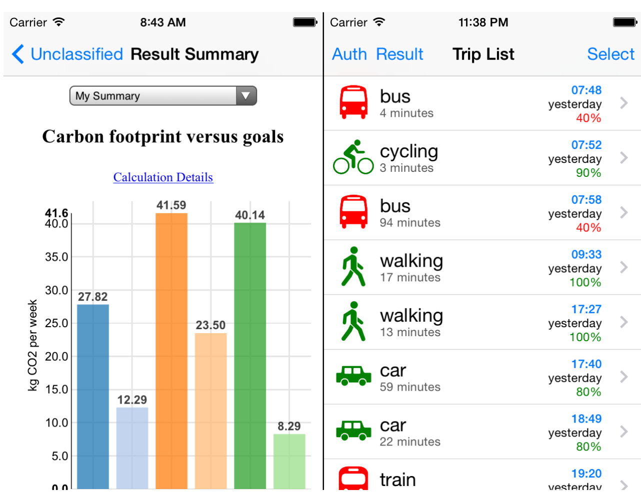

Figure 7: Personalized carbon footprint and redesigned confirmation screen

This section sketches the characteristics of the data that was collected, computes the accuracy of the automated inference algorithm, and evaluates the option of using user models.

Although users consented to our privacy policy [3] by downloading the apps from the app stores, they did not provide explicit consent to having their data used for research. So this paper will not include an analysis of the detected travel patterns.

6.1 Behavioral Evaluation

We were able to collect 7439 trip sections from 44 users in the San Francisco Bay Area for a period of roughly 3 months (2014-04-12 to 2014-07-18). Since the data collection was not part of an official study and participants were not paid, participants started and stopped collection at various times. We can infer user acceptance of the system by looking at the distribution of trip sections over time and across users (Figure 9). We can see that the total number of sections detected was relatively constant, but the number of confirmed sections went down every month. Further, the distribution across users indicates that there were different responses - *disengaged* (uninstalled the app), *tolerant* (continued data collection but didn't bother to confirm), and *engaged* (confirmed trips religiously).

In order to lower barriers to confirming, we have redesigned the UI to highlight low confidence trips, allow confirmation from the list view, and enable confirmation of multiple trips at one time (Figure 7). The redesigned UI was deployed roughly a month ago, and informal feedback has been uniformly positive. We hope to report those results over a longer time frame in the future.

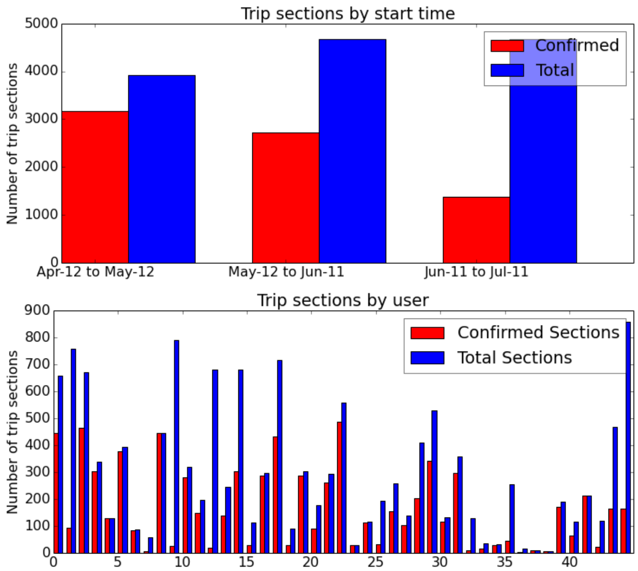

Figure 9: Number of confirmed trip sections per month and per user. Users 36-39 are disengaged, 10 and 13 are tolerant, and 6 and 9 are engaged

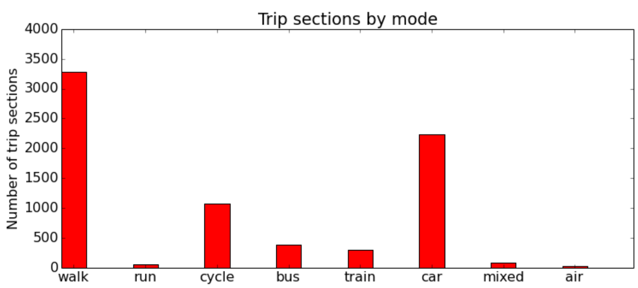

Figure 10: Number of confirmed trip sections per mode

6.2 Mode inference using an aggregate model

6.2.1 Evaluation metrics

As we can see from Figure 10, the distribution of trip modes is skewed, and so the overall accuracy might be a misleading metric. If the class specific accuracies are not uniform, the overall accuracy may simply reflect the proportion of high accuracy classes in the dataset. So we evaluate the accuracy of our learning methods separately for each mode. We do this by generating a confusion matrix using stratified 5-fold validation, as shown in Algorithm 1.

6.2.2 Feature and model selection

```
for (train, test) ∈ kFolds do
    model = algo.fit(X[train], y[train]);
    yPred = model.predict(X[test]);
    cmRaw = confusion_matrix(y[test], yPred);
    // [610 12 1];
    rptSum = repeat(sum).reshape();
    // [623 623 623];
    thisCm = cmRaw / rptSum // [98 2 0];
    sumCm = sumCm + thisCm // [188 10 2];
end
avgPctCm = sumPctCm / kFolds
```

Algorithm 1: Stratified k-fold confusion matrix computation

Table 3: Accuracy per mode with different sets of features

Feature set	walk	cycle	bus	train	car	air
Generic	95	85	34	37	88	69.0
G+A	95	85	30	36	89	65.0
G+A+L	96	88	48	55	92	83.0
G+A+B	95	85	63	49	89	74.0
G+A+B+L	96	88	66	63	91	79.0
G+A+B+L+T	95	88	71	62	91	83.0

There are several potential sets of features and models that we can choose from. We used the `scikit-learn` [18] library to evaluate the use of various combinations of models and features. Based on the work done in [25], we started with random forests as the model and explored various feature sets, and then we picked one feature set and validated the choice of model. We present a summary of our results here. For more details on the experimental evaluation of different models, please refer to the associated technical report [21].

Figure 8 and Table 3 show that while the accuracy of walk and bike modes is uniformly high, the addition of geospatial information doubled the accuracy of bus and train modes. Therefore, we select the **G+A+B+L+T**(4.3) feature set for further analysis.

After selecting features, we evaluated the use of other learning algorithms. In [25], the other algorithms evaluated were primarily parameteric, and did not perform well. Figure 8 shows that we were able to reproduce this result using a linear SVM in which the parameters were tuned using grid search. We also tried a different non-parametric method (k-nn). which was better than the parametric method, but worse than random forests.

Since the bad performance of linear models may be due to the fact that the data is not linearly separable, we also explored the use of non-linear kernels (`rbf`, `poly`, `sigmoid`) with linear models (`SGD`, `LDA`, `SVM`) [21]. However, the best results with parametric models are still worse than the random forest result, especially for the `train` mode.

6.3 Mode inference using user specific models

As described earlier in Section 1, we think that a new learning paradigm of building user specific models can help improve accuracy. In order to test this hypothesis, we took all users who had more than 150 confirmed

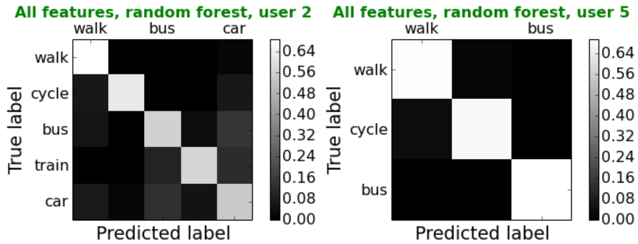

Figure 11: Confusion matrices for high accuracy user models

trips, and built user-specific models, in which we considered only the prior trips for that user.

Using these models, we were able to find users for whom mode inference accuracy is high although the percent of motorized trips is fairly high. We pick two of these and plot confusion matrices for them.

The results are shown in Figure 11. As we can see, the motorized mode accuracies are higher than a combined model. This indicates that this is a promising area to explore.

6.4 Computer system scalability

Our initial prototype system ran on an Amazon AWS micro instance with 1 vCPU and 1 GiB of RAM.

The metrics that we used to evaluate performance and scalability are shown in Table 4. Note that some of these metrics are generated from scripts that are run periodically using cron jobs, so we can measure the total run time in addition to the time taken for each operation.

As we can see Figure 12, the mongo DB `sections.bson` file grows linearly with the growth in the number of sections. However, the other metrics don't fare as well. We see a dramatic increase in run times around the end of Jun, a couple of weeks after we turned on our machine learning pipeline. The script to determine which sections are commute sections, for example, ended up taking a whole day, while even the mean response time to return results was in the minutes. Note that, similar to other work, most of the time spent in running our pipeline is in reading the data and generating the feature matrix. Additional performance charts are in [21].

To work around these issues, we have moved from the micro instance to an x-large instance (4 vCPU, 15 GiB, SSD storage). We have also simultaneously switched to collecting data with explicit consent for research, so the number of trips (< 2000) is not sufficient to stress the old system, let alone the new one. We will revisit this issue once the size of the collected data approaches the initially collected data.

7. FUTURE WORK

Our primary focus for future work will be on improving the phone layer, the web layer, and the analytics.

7.1 Phone layer

The primary challenge at the phone layer is to motivate people to share their travel behavior. We need to do this by both reducing the work, and increasing the rewards. We can reduce effort by improving phone app design further, and increase rewards through some form of gamification. In addition, although the Moves team is working on optimizing power consumption, the increased power drain is still noticeable. We should consider our data needs and see if it is possible to write our own data gathering that is more optimized to our workload.

7.2 Web layer

The primary challenge here is that of data access and visualization. The current web app displays a subset of data that we believe will be useful at the aggregate level. However, we can easily imagine that there might be other queries that might also be interesting to other researchers. How do we change the web app to support richer visualizations, and have the option for them to be open ended? Do we support a rich, scriptable query language for even more powerful access? How do we do so without sacrificing privacy?

7.3 Analytics

Finally, we want to run additional analytics to recommend actions that users and planners can take to reduce carbon emissions. We need to think of these potential recommendations, and then implement the code to detect them using external data sources. We also need to improve the carbon emission calculation to take into account more complex factors such as carpooling, fuel efficiency and so on.

8. CONCLUSION

We have motivated the need for Participatory Classification, a framework for minimally invasive, large scale collection of ground truth from regular users. We have presented an architecture that can be used to implement this framework, and described how we have used this architecture to build and evaluate an end-to-end system that has collected labelled trip patterns for 44 users in the San Francisco Bay Area over 3 months. In order to reduce the burden on the user, we generate a proposed classification, along with a confidence, which allows users to quickly confirm the trip if the classification is correct. Our accuracies for this proposed classification, are 60-95% using a set of speed and spatiotemporal features modelled using a random forest. We are able to perform aggregated analysis of travel patterns and generate results such as popular routes by

Table 4: List of scalability metrics

Metric	Invocation	Description
DB size	N/A	Size of the exported `sections.bson` file, in MB
Data retrieval	*/2, */4	Script that connects to Moves, retrives trip data for each user, and saves it to the database. Sleeps for 2 minutes after reading data for every 10 users in order to stop overwhelming Moves. Originally ran every two hours, switched to every 4 hours when the classification pipeline was enabled.
Commute sections	7	Script that reads sections for a user, determines home and work locations and commute trips, and saves the commute flag back to the database
Pipeline	*/4	Script that reads the confirmed sections as the training set and auto-classifies unclassified and unconfirmed sections
getUnclassifiedSections	N/A	API call to read the sections that need to be classified by this user
compare	N/A	API call to read the carbon footprint results for this user

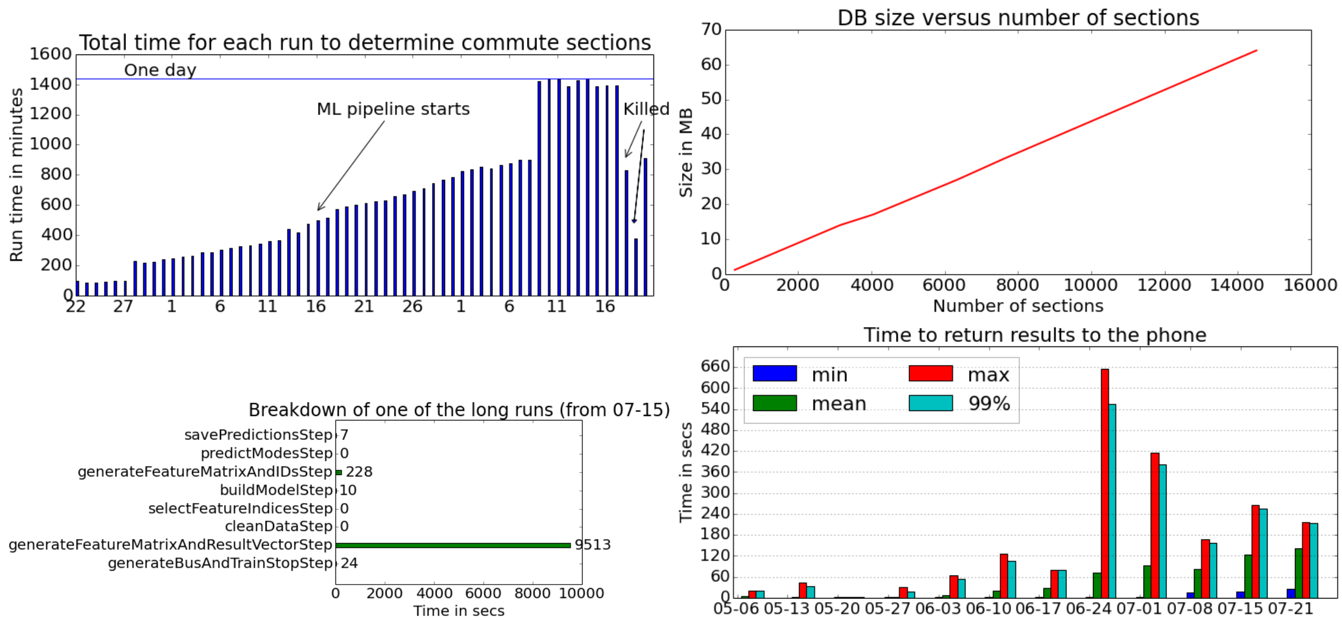

Figure 12: Changes in various performance metrics over time on the micro instance

mode and arrival times at work.

9. ACKNOWLEDGEMENTS

This work was supported in part by the Jim Gray Fellowship, in part by the NSF ActionWebs CPS-0931843, in part by NSF CISE Expeditions Award CCF-1139158, LBNL Award 7076018, DARPA XData Award FA8750-12-2-0331, and gifts from Amazon Web Services, Google, SAP, The Thomas and Stacey Siebel Foundation, Adobe, Apple, Inc., Bosch, C3Energy, Cisco, Cloudera, EMC, Ericsson, Facebook, GameOnTalis, Guavus, HP, Huawei, Intel, Microsoft, NetApp, Pivotal, Splunk, Virdata, VMware, and Yahoo!.

We would like to thank **Akshay Vij** for suggesting approaches for automated inference and for his insight into prompted recall. We would also like to thank **Shanthi Shanmugam** for all her help with the design, and for the initial `MovesConnect` OAuth implementation. And to **Thomas Raffill**, for all the reviews.

9.1 References

10. REFERENCES

[1] Azumio. Instant heart rate.

[2] Caitlin D. Cottrill, Francisco Camara Pereira, Fang Zhao, Ines Ferreira Dias, Hock Beng Lim, Moshe Ben-Akiva, P. Christopher Zegras. The future mobility survey: Experiences in developing a smartphone-based travel survey in singapore. *Transportation Research Record: Journal of the Transportation Research Board*, (2354):59–67, 2013.

[3] E-Mission. E-mission: Data driven carbon emission reduction.

[4] N. Eagle and A. (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, May 2006.

[5] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 29–39. ACM, 2008.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A

density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, page 226âĂŞ231, 1996.

[7] N. Eyal. How to manufacture desire: An intro to hooks.

[8] J. Froehlich, T. Dillahunt, P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. A. Landay. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1043âĂŞ1052, 2009.

[9] J. Jariyasunant. *Improving Traveler Information and Collecting Behavior Data with Smartphones.* PhD, University of California, Berkeley, Berkeley, 2012.

[10] Jerald Jariyasunant, Maya Abou-Zeid, Andre Carrel, Venkatesan Ekambaram, David Gaker, Raja Sengupta. Quantified traveler: Travel feedback meets the cloud to change behavior. Technical Report UCTC-FR-2013-06, University of California Transportation Center, Sept. 2013.

[11] J. K. Laurila, D. Gatica-Perez, I. Aad, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen, and others. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*, 2012.

[12] Mikhail Chester, Arpad Horvath. Environmental life-cycle assessment of passenger transportation: A detailed methodology for energy, greenhouse gas, and criteria pollutant inventories of automobiles, buses, light rail, heavy rail and air. UCB-ITS-VWP-2008-2, UC Berkeley, Berkeley, 2008.

[13] Moves. Moves: Activity diary for iphone and android.

[14] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, page 55âĂŞ68, 2009.

[15] S. Nath. Ace: exploiting correlation for energy-efficient and continuous context sensing. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 29–42. ACM, 2012.

[16] A. Parate, M.-C. Chiu, D. Ganesan, and B. M. Marlin. Leveraging graphical models to improve accuracy and reduce privacy risks of mobile sensing. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 83–96. ACM, 2013.

[17] N. Partners. Nvd3.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks*, 6(2):1–27, Feb. 2010.

[20] R. Sen, B. Raman, and P. Sharma. Horn-ok-please. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 137–150. ACM, 2010.

[21] K. Shankari, M. Yin, S. Shanmugam, D. E. Culler, and R. H. Katz. E-mission: Automated transportation emission calculation using smart phones. Technical Report UCB/EECS-2014-140, EECS Department, University of California, Berkeley, Aug 2014.

[22] A. Vij and K. Shankari. When is big data big enough? implications of using gps-based surveys for travel demand analysis. Technical Report UCB/EECS-2014-141, EECS Department, University of California, Berkeley, Jul 2014.

[23] Waze. Free community-based mapping, traffic & navigation app.

[24] F. Xu, Y. Liu, T. Moscibroda, R. Chandra, L. Jin, Y. Zhang, and Q. Li. Optimizing background email sync on smartphones. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 55–68. ACM, 2013.

[25] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma. Understanding transportation modes based on GPS data for web applications. *ACM Transactions on the Web*, 4(1):1–36, Jan. 2010.