Crowdsourcing Biking Times

Mingsheng Wu
College of Mathematical and
Natural Sciences
University of Maryland
wumingsheng@umd.edu
Lingzi Hong
College of Information Studies
University of Maryland
lzhong@umd.edu
Vanessa Frias-Martinez
College of Information Studies
University of Maryland
vfrias@umd.edu

PURBA'15, Osaka, Japan.

DOI string from ACM form confirmation

Abstract
Urban cyclists often rely on Google's biking directions to consult routes and times. However, cyclists have reported that those estimates can sometimes be inaccurate [1]. In this paper, we explore the accuracy of Google biking times using a crowdsourced approach. Specifically, we use real biking data from a bike sharing system as ground truth and evaluate the automatic computation of Google's biking times. We analyze similarities and differences between the two as well as the role that measurable factors such as trip distance or slope might play in the temporal differences. Finally, we propose a predictive model based on a set of measurable factors that improves the accuracy of Google's biking time computations by 5%.

Author Keywords
urban computing;bike sharing systems;predictive analytics

Introduction
In the last ten years, there has been a large increase in bike use specially among young urbanites. Only in the US, commuting by bike has grown by 60% over the past decade [2]. Bike sharing systems have partially contributed to this surge by allowing cyclists to easily borrow bikes for short trips in cities such as New York

(Citi Bikes), Chicago (Divvy), San Francisco (SFBikeShare) or Washington D.C. (Capital Bikeshare).

There exists a wide range of mobile applications that allow cyclists to map biking routes and to share them with others including *Ride with GPS*, *Map my Ride*, *Veloroutes* or *Routeslip*. For any existing route, cyclists can download maps from other cyclists and check specific features such as directions or the length, duration or elevation of the route. However, most of these applications are used by leisure cyclists who focus on long, rural bike trips or on touristic urban trips. Regular urban cyclists and commuters willing to plan their daily routes typically resort to the biking directions in Google Maps. As opposed to the mobile applications, where maps are created by cyclists, Google Maps automatically generates biking routes and times using a set of algorithms based on map information and average biking speeds, among other features. As a result, urban cyclists sometimes report inaccuracies with the biking times and routes proposed by Google [1].

In this paper, we analyze the accuracy of google biking times using crowdsourced data from thousands of urban bike routes generated by the users of a bike sharing system. We take advantage of the open data initiatives present in many bike sharing programs that allow to access individual trip data and use that information as ground truth. Our contributions are twofold: (i) we analyze the differences between google biking times and the crowdsourced data, and evaluate the role that certain measurable factors such as trip distance or slope might play; and (ii) based on the previous analysis, we propose a predictive model that incorporates the measurable factors into current google's computations to improve the accuracy of their biking times when compared to the crowdsourced data (ground truth). Since bike sharing systems are not present in all cities, such predictive models would allow Google to enhance their biking time computations worldwide using crowdsourced data from just a few cities.

Related Work

Our work is based on the idea that crowds can act as passive or active sensors of human behavior [18]. Active sensing behaviors or participatory sensing empowers individuals to gather and analyze data from their own surroundings with the objective of sharing local knowledge [17, 6]. On the other hand, passive or opportunistic sensing behaviors require less user involvement and include scenarios such as location sampling without explicit action from the user [12, 8, 9, 7, 5]. There exist multiple mobile systems that focus on active sensing to improve biking experiences [4]. Our research is framed within opportunistic sensing as we use biking data passively gathered by bike sharing programs from the interactions of thousands of users with the systems.

From an analytical perspective, there exists a large number of papers studying the performance of bike sharing systems. Some of the most important issues are balancing, or how to guarantee that there are available bikes in all stations at all times [10]; predicting bike usage between pairs of stations [16] or trip intention [19]; understanding the impact and effect of bike sharing systems on other means of transportation [15] and general system analytics [13, 11]. However, to the best of our knowledge, our approach is novel in proposing the use of bike sharing data as a ground

truth source to improve the automatic computation of biking times at large-scale in platforms such as Google Biking Directions in Google Maps.

Data and Pre-processing

Bike Sharing Data. Bike sharing systems allow users to borrow bikes from specific locations (stations), use them for a short period of time and return them to any station in the system. Typically, there exist two types of memberships: subscribers and casuals. Subscribers are frequent users who pay monthly or yearly fees while casuals can get daily or weekly passes. The pricing scheme usually offers the first 30 to 45 minutes free after which additional fees are charged based on type of subscription. For this paper, we collected as ground truth the trip history data from the Capital Bikeshare system in Washington, D.C. throughout 2013 ($1.5M$ trips). Trip history data contains the following information for each trip between any pair of stations: start and end date and time, start and end station, duration, bike id and membership type. Additionally, the geographic coordinates for each station in the system (316) are also provided. Unfortunately, trip history data does not contain information regarding the nature/intention of the trip *i.e.,* we don't know whether a given trip was a direct, non-stop trip between two stations or whether the user made some brief stops along the way without docking the bike into a station until its final destination.

Google Biking Data. Google Maps offers the possibility of retrieving biking information between any two given locations through the Google Directions API. The output includes, among other variables, information about the route, the elevation, the distance or the time it would take to make the trip on a bike. We collected these variables from Google's API for all pairs of stations in the Capital Bikeshare system for which exists at least one trip. Additionally, the biking time between any pair of stations might change depending on the direction of the trip: going from station A to station B might require biking up a high slope whereas going in the opposite direction would be a downhill. For that reason, we collect google's biking information between pairs of stations for both directions.

Filters. Google does not fully disclose how biking times are computed. However, they claim to assume direct, non-stop trips between origin and destination when computing trip durations. On the other hand, trips in bike sharing systems will reflect different types of behaviors: from direct, non-stop trips whose duration represents the actual biking time to *wandering trips* that might include stops along the way without re-docking the bike which will increment the total trip duration loosing accountability for the actual biking time. To be able to compare google biking times with crowdsourced times from a bike sharing system, we need to exclusively focus on direct, non-stop trips which represent actual biking times. However, since information about the nature of the trip is not provided in the bikeshare dataset, we propose to focus our analysis on what we call *commuting trips*. We define commuting trips as biking trips in the bike sharing system that go from *residential* to *work* stations during weekday mornings. Our assumption is that people commuting from home to work areas will highly probably take direct, non-stop routes to get to work as soon as possible.

To detect commuting trips in a bike sharing dataset, we propose a two-step process. First, we filter out trips

from casual members who might be more prone to show wandering behaviors since they are not frequent users of the system and exclusively focus on subscribers' trips. Second, we select all morning trips that happen from a residential station to a work station. The final set of trips will be the commuting trips whose durations we will compare against google's biking times. To be able to apply this two-step process we need to determine which stations in the bike sharing system can be defined as residential or work stations. The intuition is that residential stations should observe a large number of outgoing trips in the morning and work stations a large number of outgoing trips in the afternoon from people commuting to and from work respectively. To identify such behaviors, we propose to compute for each station in the bikeshare dataset an activity vector that represents the hourly average of incoming trips and the hourly average of outgoing trips throughout all trip data gathered. Formally, the activity vector for station i is defined as $a_i(t), t = \{1, ..., 48\}$ where $t = \{1, ..., 24\}$ represents the hourly average number of incoming trips to station a_i and $t = \{25, ..., 48\}$ the hourly average number of outgoing trips from station i. We normalize the activity vectors for all stations with the Z-score transformation and use k-means to cluster them and infer the best distribution of biking behaviors (clusters) with the Davies-Bouldin index.

Figure 1 shows the best clustering results ($k = 3$) using the Capital bikeshare dataset. Each line represents the activity vector of a station and the color its cluster membership: blue, red or green. The x-axis represents 48 hours: the first 24 are for hourly averages of the number of incoming trips and the last 24 for outgoing trips' hourly averages. We observe three clearly

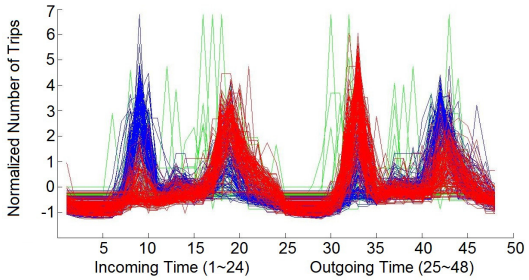

Figure 1: Clusters identifying residential and work locations.

differentiated behaviors: the blue stations show a large number of incoming trips in the mornings($[6am - 10am]$) and a large number of outgoing trips in the afternoon ($[5pm - 10pm]$). We associate this cluster to work stations. The red cluster shows the opposite behavior: a large number of incoming trips in the afternoon and a large number of outgoing trips in the mornings. We associate this cluster to residential stations. The third cluster (green) is a mix of both without identifiable behaviors. It is important to mention that these results are consistent with the analysis of other bike sharing systems to understand types of trips [14, 3]. With these results in hand, we select as commuting trips all subscribers' trips in the bikeshare dataset that happen between 6am and 10am, whose origin station is labelled as residential and whose destination station is labelled as work.

Analysis

In this analysis, we want to evaluate the accuracy of google biking times using data from a bike sharing system. Our assumption is that the crowdsourced

information gathered from bike sharing systems represents the ground truth as opposed to the biking times computed by Google which are based on a set of algorithms and assumptions regarding biking conditions. As stated in the previous section, since the nature of the biking trips in a bike sharing system can include wandering behaviors, we exclusively focus on morning commuting trips and compare these against their Google counterpart.

Figure 2 shows the distribution of the crowdsourced biking times versus google's times for all trips. Specifically, for each trip and direction, we report its crowdsourced duration and the google biking time extracted from Google Directions' API. Since many pairs of google's and crowdsourced times may overlap, we present the final plot as a heat map where each square (of size $20x20sec$) is colored based on the number of trips that share a given pair of biking times within that range. The black line $y = x$ separates trips with longer crowdsourced biking times (above) from trips with longer google biking times (below). We can observe that approximately 20% of the trips show crowdsourced biking times shorter than the ones computed by Google. For those trips, the crowdsourced biking time is, on average, $1.49min$ shorter than the biking times reported by Google. On the other hand, the remaining trips share equal or longer crowdsourced times than the times reported by Google which are, on average, $2.10min$ shorter. Both differences were statistically significant with $p < 0.01$ using the Mann-Whitney U test. As a result, it is fair to say that there might exist certain factors that make google biking times be a little bit off with respect to the real biking times extracted from the bike sharing system. Next, we focus our analysis on the potential

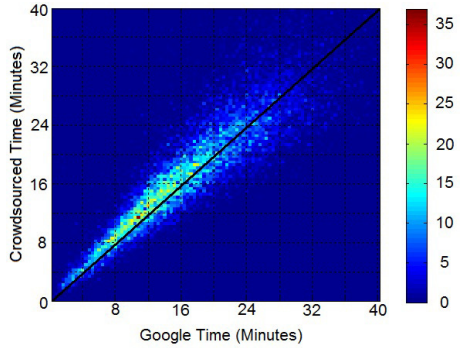

Figure 2: Crowdsourced biking times versus Google biking Times.

impact of two factors: distance and slope. Our end objective is to disentangle whether these factors might play a role in the biking time differences observed between google times and crowdsourced times.

Distance. To understand the role that distance might play in the duration of a biking trip, we represent the distribution of google and crowdsourced biking times with respect to distance and analyze similarities and differences between the two. Figure 3 shows the heat maps for google biking times (gt) (a) and crowdsourced biking times (ct) (b) for each trip and direction with respect to distance. Each square represents the number of trips within a $.1mix20sec$ range.

The Figure shows that as distances increase, the spread of the time distribution is much larger for the crowdsourced times than for google biking times. For example, at $2mi$ google trips show biking times in the range $[10 - 15min]$ ($\bar{gt} = 12min$) while the crowdsourced trips report times between $[9 - 17min]$

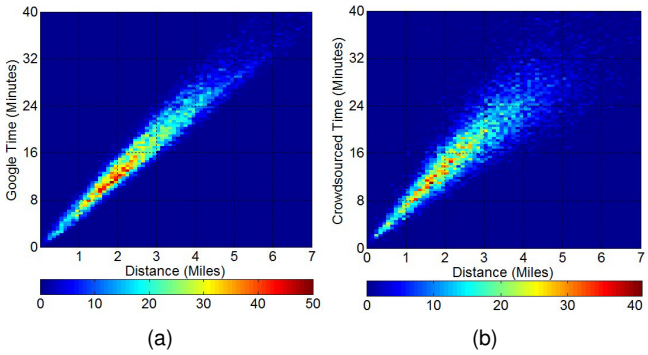

Figure 3: Google and Crowdsourced Biking times versus Distance.

($\bar{ct} = 13min$); and while at $3mi$ all google biking times are between $[15 - 21min]$ ($\bar{t} = 18min$), the crowdsourced times vary from $14min$ to $24min$ ($\bar{t} = 20min$). This trend in the spread becomes more accentuated for longer trip distances ($d \geq 4mi$) where the time spread is not only larger (with $\bar{gt} = 24min$ vs $\bar{ct} = 26.5min$), but also looses its linear relationship with the distance as shown by the cloud of points at the tail of the plot 3(b). As distance increases, google appears to be assuming slightly higher biking speeds than the real ones and thus computing shorter biking times on average. Additionally, statistical tests showed that the differences between google and crowdsourced times were significant at $p < 0.01$. These results show that google does not do a bad job at approximating the crowdsourced biking times with average differences within $\approx 2min$. However, the longer the biking distances, the more difficult it is for google to capture the variance of biking times gathered in the crowdsourced data.

Slope. To understand the impact that slope might have on trip biking times, we define two slopes: the ascent slope and the descent slope. The ascent slope measures the total upward slope of a route while the descent slope measures the total descent. To compute these two slopes, we use the waypoints provided by Google Directions API and the elevations provided by Google Elevation API. Specifically, for each pair of stations and direction in our dataset, we retrieve all locations (waypoints) in Google's suggested route and its individual elevations. We compute the ascent slope (as) for each pair of stations and direction as the sum of all the upward slopes in the route: $as_{st_i,st_j,dir} = \sum_1^Z \frac{\triangle y}{\triangle x}$ where Z is the number of upward slopes in the route between stations st_i and st_j with trip direction dir; y represents the change in elevation between two locations in the route and x the horizontal distance between them. Similarly, we define descent slope for each pair of stations and direction $ds_{st_i,st_j,dir}$ as the sum of all the downward slopes in the route.

Figure 4 shows the heat maps for google (a) and crowdsourced biking times (b) for different descent slope values (slope values are scaled as $10/slope$ for clarity purposes, which means that smaller x-axis values are associated to larger slopes). Focusing on the steepest descent slopes ($ds \geq 5ft$, x-axis ≤ 2) we observe that the largest volume of trips (denser areas in the heat map) have google biking times in the $[5 - 16min]$ range ($\bar{gt} = 10min$) whereas crowdsourced biking times expand across a larger time range $[7 - 22min]$ ($\bar{ct} = 15min$). These differences were statistically significant with the Mann-Whitney U test at $p < 0.01$. As a result, it appears that Google might be modeling lower biking times than the real ones for steep descents. Moving on to flatter descent slope

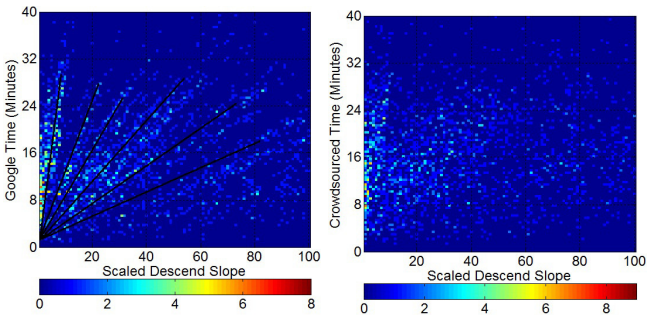

Figure 4: Biking times versus Descent Slope.

values, google biking times appear to be approximating some type of linear relationships between the slope and the times *i.e.,* smaller descent slopes are associated to longer biking times with varying slope coefficients (see the fitted black lines). However, such relationship is not observed in the crowdsourced biking times which appear to be much more chaotic, reflected as a pattern-less cloud of points. On the other hand, the relationship between biking times and ascent slopes did not reveal any relevant observation except for the fact that for extreme ascent slope values ($as \geq 5ft$), google biking times show average values of $\bar{bt} = 18min$ whereas the values are slightly lower for the real crowdsourced times with $\bar{ct} = 16min$. This result indicates that for extreme ascents, Google is computing slightly higher biking times than the ground truth.

Predicting Biking Times

The analysis presented shows that although Google algorithms tend to capture well crowdsourced biking times, there are certain scenarios such as long-distance trips or extreme slopes which are more prone to temporal differences. Next, we explore predictive models to enhance current Google biking time computations taking into account the features explored in the previous section: distance, ascent and descent slope. The final objective is to improve Google's biking times worldwide using crowdsourced datasets from existing bike sharing systems.

Given the non-linear nature of some of the features, we explore two predictive models: Random Forests (RF) and Support Vector Regression (SVR). We define as baseline the simplest model: use Google times (gt) to predict the crowdsourced biking times (ct). And explore how features such as distance or slope might improve the prediction accuracy of the crowdsourced biking times by incorporating them in a more complex predictive model. Each trip in our training and testing sets is defined as $(ct, gt, distance, ascent, descent)$ where ct is the crowdsourced time to be predicted, gt is the google biking time, and distance, ascent and descent slope are the specific features for that trip and direction. We divide the dataset into randomly selected training (80%) and testing sets (20%) and report the average correlation between real and predicted values (r) and the mean-square errors (MSE) across ten runs. For the RF, we use 5-fold CV to adjust the number of predictors explored at each step; and we use a 10-fold CV grid search approach to tune for the SVR parameters epsilon (ϵ) and cost (C).

Table 1 shows the predictive accuracy for the baseline and for the model with the additional features. The first important observation is that the crowdsourced times can be predicted quite well simply using Google biking

times ($r = 0.93$). However, adding the other predictors to the baseline (distance and slope) improves the predictive power of the model by 5%: from $r = 0.93$ to $r = 0.98$ for RFs and from $r = 0.56$ to $r = 0.66$ for SVR. Focusing on the best model (RF), we analyze the importance of the predictors in the trees using the permutation test. In our case, the importance of the features (in order) were: Google time (0.96), distance (0.90), descent (0.014) and ascent (0.009). Interestingly, the importance confirms the results discussed in the previous section: distance probably allows to incorporate the long-distance trips that Google failed to approximate in the baseline model; and similarly, although to a minor extent, descent and ascent slopes might explain the extreme slope cases that the baseline did not cover. As a result, the final model improves the predictive power of the baseline by 5%. Finally, the correlation between the crowdsourced and the predicted times is not perfect ($r = 0.98$), which might reveal that there exist other features that could be playing a role in the final crowdsourced biking times. These features can be measurable such as the weather or more latent like local knowledge e.g., "knowing that on a given road it's safe to turn with a red light". Future work will explore these and other potentially predictive features.

Discussion and Future Work

Our work shows that platforms such as Google Biking Directions are doing a good job at approximating real, crowdsourced biking times through automatic computations. However, we have revealed that there are certain scenarios such as longer trips or steep slopes that are harder to model through Google's formulas and heuristic rules. To solve that, we have proposed a predictive model that enhances current google's biking times computations with respect to real, crowdsourced biking times extracted from bike sharing systems. At its core, the model incorporates distance, ascent and descent slope as predictors and increases the predictive power of the baseline by 5%. Future work will expand the analysis to multiple cities and will evaluate the impact that other measurable and latent features such as weather might have on improving the prediction of the real biking times.

	RF		**SVR**	
Predictors	r	MSE	r	MSE
gt (baseline)	0.93	6689.7	0.56	60812.5
gt,d,as,ds	0.98	2856.47	0.66	50363.1

Table 1: Prediction results for the crowdsourced biking times: baseline model (google time, gt) and model with gt, distance, ascent and descent.

REFERENCES

1. 2014. How accurate are Google Maps Cycling time estimates? http://www.betterbybicycle.com/2014/09/how-accurate-are-google-maps-cycling.html. (2014).
2. 2015. League of American Bicyclists. www.bikerleague.org/content/resources. (2015).
3. E. Come and L. Oukhellou. 2014. Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the VélibSystem of Paris. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 39.
4. S. Eisenman and et al. 2009. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN)* 6, 1 (2009).

5. E. Frias-Martinez, G. Williamson, and V. Frias-Martinez. 2011. An agent-based model of epidemic spread using human mobility and social network information. *International Conference on Social Computing* (2011).
6. Vanessa Frias-Martinez, Sae-Tang Abson, and Enrique Frias-Martinez. 2014. To Call, or To Tweet? Understanding 3-1-1 Citizen Complaint Behaviors. *International Conference on Social Computing* (2014).
7. Vanessa Frias-Martinez, Cristina Soguero, and Enrique Frias-Martinez. 2012a. Estimation of Urban Commuting Patterns Using Cellphone Network Data. *Workshop on Pervasive and Urban Applications, PURBA* (2012).
8. Vanessa Frias-Martinez, Victor Soto, Heath Hohwald, and Enrique Frias-Martinez. 2012b. Characterizing Urban Landscapes using Geolocated Tweets. *International Conference on Social Computing* (2012).
9. Vanessa Frias-Martinez, Victor Soto, Jesus Virseda, and Enrique Frias-Martinez. 2012c. Computing Cost-Effective Census maps from cell phone traces. *Workshop on Pervasive and Urban Applications, PURBA* (2012).
10. C. Fricker and N. Gast. 2014. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *EURO Journal on Transportation and Logistics* (2014).
11. J. Froehlich, J. Neumann, and N. Oliver. 2009. Sensing and Predicting the Pulse of the City Through Shared Bicycling. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence (IJCAI'09)*. 7.
12. R. Ganti, F. Ye, and H. Lei. 2011. Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE* 49, 11 (2011).
13. A. Kaltenbrunner and et al. 2010. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing* 6, 4 (2010).
14. N. Lathia, S. Ahmed, and L. Capra. 2012. Measuring the impact of opening the London shared bicycle scheme to casual users. *Transportation research part C: emerging technologies* 22 (2012).
15. Z-C. Li and et al. 2015. Modeling the Effects of Public Bicycle Schemes in a Congested Multi-Modal Road Network. *International Journal of Sustainable Transportation* 9, 4 (2015).
16. D. Singhvi and et al. 2015. Predicting Bike Usage for New York Citys Bike Sharing System. In *AAAI 2015 Workshop on Computational Sustainability*.
17. J. Sprake and P. Rogers. 2014. Crowds, Citizens and Sensors: Process and Practice for Mobilising Learning. *Personal Ubiquitous Computing* 18, 3 (2014).
18. M. Vukovic, S. Kumara, and O. Greenshpan. 2010. Ubiquitous Crowdsourcing. In *ACM International Conference on Ubiquitous Computing*.
19. C. Wang, G. Akar, and J. Guldmann. 2015. Do your neighbors affect your bicycling choice? A spatial probit model for bicycling to The Ohio State University. *Journal of Transport Geography* 42 (2015).