Smart Parking Pricing: A Machine Learning Approach

Eran Simhon, Christopher Liao and David Starobinski
College of Engineering, Boston University

***Abstract*—Crowded streets are a major problem in large cities. A large part of the problem stems from drivers seeking on-street parking. Cities such as San Francisco, Los Angeles and Seattle have tackled this problem with smart parking systems that aim to maintain the on-street parking occupancy rates around a target level, thus ensuring that empty spots are spread across the city rather than clustered in a single area. In this study, we use the San Francisco's SF*park* system as a case study. Specifically, in each given parking area, the SF*park* uses occupancy rate data from the previous month to adjust the price in the current month. Instead, we propose a machine learning approach that predicts the occupancy rate of a parking area based on past occupancy rates and prices from an entire neighborhood (which covers many parking areas). We further formulate an optimization problem for the prices in each parking area that minimize the root mean squared error (RMSE) between the predicted occupancy rates of all areas in the neighborhood and the target occupancy rates. This approach is novel in that 1) it responds to a predicted level of occupancy rate rather than past data and 2) it find prices that optimize the total occupancy rate of all neighborhoods, taking under account that prices in one area can impact the demand in adjacent areas. We conduct a numerical study, using data collected from the SF*park* study, that shows that the prices obtained from our optimization lead to occupancy rates that are very close to the desired target level.**

I. Introduction

In congested downtown areas, drivers often have a difficult time locating available parking spots. Public on-street parking is often the most coveted option because of its low prices and proximity to popular destinations. When on-street parking is priced uniformly in all locations, the demand may outweigh the supply for spots closest to popular destinations, while spots in other areas are left underutilized.

This problem causes drivers to slow down and circle around their destination for extended periods of time, waiting for a convenient spot to become available (a behavior know as "cruising"). Increased congestion, pollution, safety issues, and driver frustration can be reasonably attributed to this behavior. The work in [1] compiles results from ten studies in eight cities that surveyed drivers in congested downtown areas and found that the average estimate of the percentage of drivers cruising for parking was 34%. It is also stated in [1] that public parking has historically been priced ineffectively, and that demand-based (smart) pricing of metered parking could alleviate some of the congestion caused by cruising for parking.

Fig. 1 summarizes how smart parking pricing systems typically work: 1) A central parking authority adjusts parking prices based on occupancy rate data and pricing constraints; 2) Drivers use apps or other means to plan for a suitable parking area. A variety of mobile apps, such as ParkMe [2], Parker [3], ParkWhiz [4], and SpotHero [5] allow drivers to reserve and pay for lot and garage parking spots in advance; 3) Some parking apps also include navigation capabilities to help drivers find their chosen parking spots; 4) Smart parking meters and/or sensor networks collect occupancy rate data; 5) This data is transmitted to the central authority's data centers, where occupancy rate and pricing data are stored and used to compute price adjustments. Our approach assumes that the infrastructure described in Fig. 1 exists, and we focus only on step 1, the pricing mechanism.

Our approach shares the same objective as smart parking systems that have been implemented across several U.S. cities since 2011, including San Francisco, Los Angeles, and Seattle. The main goal of these pricing mechanisms is to keep the occupancy rate of each parking area at a target level. The target level should allow enough spots to be available in each parking area at all times, so that drivers who want to park close to their destinations can park immediately upon arrival. Drivers that are more sensitive to price can choose to park in less crowded areas at a lower price. Thus, the need for cruising is reduced.

In current systems, deployed in the aforementioned cities, the price in each parking area (e.g., a street or a block) is adjusted periodically based on the occupancy rate in the previous period. Moreover, the price in each area is set independently to those in other areas. This method has two major drawbacks: (1) it ignores the fact that changing the price in one area impacts the demand in adjacent areas, which can lead to prices cycling (2) the price changes are based on a simple heuristic which does not take advantage of the data accumulated over time, which can lead to sub-optimal pricing.

In this paper, we propose a demand-based parking pricing mechanism that can easily be implemented by public parking authorities to reduce the amount of cruising for parking that occurs in their cities. Our method employs machine learning methods to predict the occupancy rate of individual parking areas. Using this prediction model, we propose an optimization method that determines prices for each parking area such that the predicted occupancy rate in all areas meet a target level. In our solution, when predicting the demand for a given parking area, we take into account not only the price of that area but also prices of adjacent areas. By doing so, we provide a system-wide solution which prevents cycling.

The remainder of the paper is organized as follows. Section II summarizes related work. Section III presents the proposed

Fig. 1: Smart Parking Pricing System Overview
Image 2 Source: [6]
Image 4 Source: [7]

price optimization model. Section IV describes the results of a case study performed using real-world parking data. Section V contains concluding remarks.

II. Related Work

In recent years, much research has been performed on smart parking pricing systems. Several papers have utilized data from the SF*park* [8] and LA Express Park [9] smart parking pilot programs.

In [1], the authors evaluate how the price changes enacted during SF*park*'s first year affected on-street parking occupancy rate. They found a surprisingly wide range of price elasticities at the block level, suggesting that many factors other than price affect the parking demand, including location, time of day, day of week, initial price, size of the price adjustment, and the amount of time that has passed since the beginning of the program. The authors argue that because these factors vary greatly on a block-by-block basis, a theoretical model for predicting the prices that would achieve the target occupancy rate is not the best approach. Instead, parking authorities should adjust parking prices on a block-by-block basis based on observed occupancy rates. During each price update, SF*park* changes each block's prices in response to the occupancy rate from the previous price adjustment period. The authors suggest that SF*park* should now shift towards predicting the occupancy rate for the upcoming pricing period.

Using data from the SFpark pilot, [10] shows that for blocks where prices rose, occupancy rates lowered overall. However, the fraction of time that the blocks had at least one available spot did not increase. The authors suggest that in order to reduce cruising, prices may need to be adjusted based on minimum vacancy rather than average occupancy rate, and price changes may need to be larger.

Many studies, including ours suggest an approach that focuses on meeting a target occupancy level for every parking area. Ref. [11] presents the public parking pricing algorithm that has been implemented as part of the LA Express Park program. The emphasis of this algorithm is on reducing the amount of time that each block is underused or over-congested. Ref. [12] proposes a parking pricing system that utilizes a PID (Proportional Integral Derivative) feedback controller to achieve a target occupancy level. The controller attempts to minimize the parking area's occupancy error, the difference between the target occupancy rate and the current occupancy rate. Ref. [13] presents a bi-level parking pricing optimization method. The model's upper level is the parking authority's objective function. The primary objective is to minimize the absolute difference of each parking area's number of occupied parking spots from its target level. The lower level of the model is a Nash equilibrium problem describing the drivers' parking choice.

Other relevant smart parking studies include [14], [15], [16], [17], [18], [19] and [20]. Ref. [14] proposes a stochastic control approach for optimal parking pricing, incorporating stochasticities in the parking demand and the drivers' value of time. In [15], the authors show that in most areas of LA, there is usually availability at lower-priced blocks within walking distance of more popular ones, so drivers have a financial incentive to park a bit further. Ref. [16] describes the negative impacts that common mis-pricing practices have on cities and argues the importance of charging market prices for on-street parking. Refs. [17], [18], and [19] formulate competitive parking games and present pricing algorithms that aim to convert the Nash equilibrium parking assignment to the system optimal assignment, in which a total parking cost is minimized. In [20] the authors present a mechanism to match available parking spaces and drivers in an optimal manner.

Our approach is unique because it employs machine learning to develop an occupancy rate prediction method that utilizes historical parking data. In the prediction step, our method also incorporates the effects of surrounding blocks' prices on each block's occupancy rate.

III. Model Formulation

A. Method Overview

Our approach uses least-squares regression methods to formulate an occupancy rate prediction function for each parking area. These prediction functions are then used to select optimal prices such that an occupancy error metric is minimized. Machine learning is an appealing approach to solve this problem because there is no simple way to formulate a purely theoretical model that effectively characterizes the demand of all parking areas. This is because each block's parking demand is influenced by many different factors, including price, location, time of day and day of the week. We incorporate these factors into our learning model, as well as the prices of surrounding blocks, since these relationships can affect drivers' decision making. Our case study results indeed show that including factors other than price in the learning model lowers the error of the demand prediction. This method can easily be incorporated into existing systems such as SF*park* [8], the smart parking system currently in effect in San Fransisco. Now that SF*park* has released several years-worth of parking occupancy rate and pricing data, we argue that the next logical step for smart parking systems is to process this data using machine learning and incorporate the obtained insights into the price optimization model.

B. The Parking Network

We consider a city in which a central authority controls the prices of public parking. The city has a set of neighborhoods N. A neighborhood $n \in N$ contains a set B of public parking areas, which we call *blocks*. A block $b \in B$ can represent any type of parking area, such as an on-street block face, a parking lot, or a parking garage.

C. Block Pricing

The blocks in B are priced on an individual basis. All parking spots within a block b have the same price. Because parking demand generally varies based on the time of day and the day type (e.g. weekend or weekday), the block prices are divided into a set of pricing periods D. A block $b \in B$ has one current price p_b^d per pricing period $d \in D$.

Within a pricing period, the expected occupancy rates of all blocks should be roughly constant. An example of a pricing period used by SF*park* is (weekday, 12pm-3pm), see Section IV.A [21]. Separate pricing periods could also be defined for special events that affect the neighborhood's parking demand.

We denote the set of current prices of all blocks in B for pricing period d by P_B^d, such that $p_b^d \in P_B^d$. The prices in P_B^d are updated based on the optimization problem presented in the next section.

D. Occupancy Prediction

The expected occupancy rate o_b^d for block b during pricing period d is defined as follows:

$$o_b^d = f\left(b, d, P_B^d\right). \tag{1}$$

The occupancy rate prediction function f is a function of b, d, and P_B^d. Thus, the prediction of the occupancy rate of block b is affected by the prices of the other blocks in its neighborhood. Given adequate historical parking pricing data, this function can be learned via least-squares regression methods, with the block ID, pricing period ID, and the prices of all blocks in the neighborhood as features and the average occupancy rate of the block during the sample time period as the response.

E. Optimization Problem

We denote the set of expected occupancy rates of all blocks in B for pricing period d by O_B^d, such that $o_b^d \in O_B^d$. We assume that all blocks in set B share a target occupancy rate of o_T. The set of optimal prices P_B^d and the corresponding set of expected occupancy rates O_B^d are computed according to the following objective function:

$$\min_{P_B^d} \sqrt{\sum_{b \in B} \frac{\left(o_T - o_b^d\right)^2}{|B|}} \tag{2a}$$

s.t.

$$p_b^d \geq p_{min}, \forall b \in B, \tag{2b}$$

$$p_b^d \leq p_{max}, \forall b \in B. \tag{2c}$$

Eq. ((2a)) minimizes the root-mean-squared occupancy error, where the occupancy error is the absolute difference between the target and expected occupancy rates. This minimization function is subject to a minimum price constraint (2b) and a maximum price constraint (2c). Other pricing constraints, such as the maximum amount that the price can change per adjustment, can be incorporated into this model as well.

IV. Case Study

A. SFpark Pricing Mechanism

As an example of how a current pricing mechanism works, we next describe the SF*park* on-street parking rate adjustment method [21]. Each block-face is priced separately, and prices are set differently for different pricing periods (time-of-day and day-type). For most blocks, the time-of-day periods are open-12pm, 12pm-3pm, and 3pm-close. The day types are weekend and weekday. Prices are updated every couple of months (e.g., they have been updated 17 times from July 2011 through December 2015). The target occupancy rate range is 60-80%. If a block's measured occupancy rate since the last price update falls within this range, the price is not changed. If the measured occupancy rate is within 80-100%, the hourly rate is raised by \$0.25. If the measured occupancy rate is within 30-60%, the hourly rate is lowered by \$0.25. If the occupancy rate is below 30%, the hourly rate is lowered by \$0.50. Minimum and maximum hourly rate constraints are \$0.25 and \$6.00, respectively.

The algorithms implemented in the LA Express Park and Seattle pricing systems have some differences, but they follow the same basic principle of adjusting the price of an individual parking area by a fixed amount when its observed occupancy metric falls outside of a target range.

B. Data Description

The SF*park* website [22] provides meter rate adjustment spreadsheets containing data regarding the 17 on-street parking price adjustments that have taken place from July 2011 through December 2015. We use data from these spreadsheets to compile a sample data set for the experiments described in this section. Each sample describes the following information: pilot area (neighborhood), street/block name, day type, time-of-day period start time, occupancy rate since the last price update, price of the block since the last price update, and prices of all other blocks in the neighborhood since the last price update. The data set that we compile contains 3,068 samples from the 44 blocks in the South Embarcadero neighborhood.

Figure 2 summarizes the SFpark performance from 2011 to 2015 in the South Embarcadero neighborhood. The figure includes five metrics related to occupancy rate for each of the block/time/day-type combinations. As can be seen in the

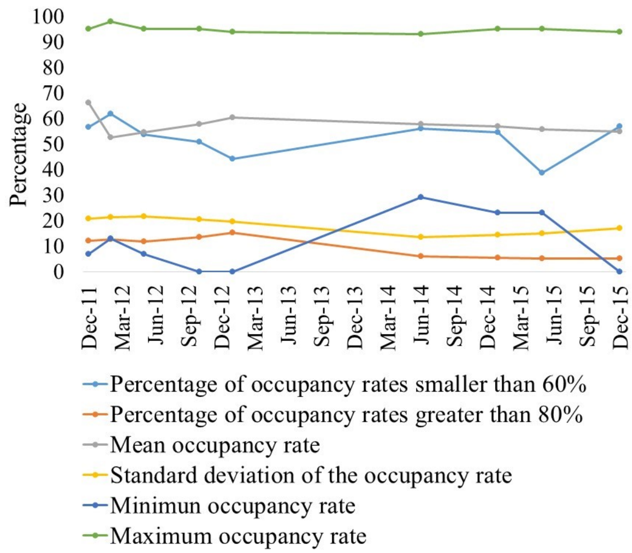


Fig. 2: SFpark performance summary

figure, the system yields modest or almost no improvement during its years of operations. For instance, during all those years there have been block/time/day-type combinations with almost 100 percent occupancy rate.

C. Demand Learning

In our method, each block has its own unique occupancy rate prediction function that can be learned using real data. These functions need not be mathematically nor computationally complex. Our case study results show that elastic-net regression methods result in sufficiently low prediction errors. Furthermore, with least-squares regression occupancy rate prediction functions, we can utilize gradient-based optimization methods to select optimal prices, which may be more reliable and efficient than non-gradient-based methods for our problem.

A software implementation is developed in R [23] to carry out demand learning. The goal of this software is to fit the linear occupancy rate prediction function as defined in Eq.((1)) to the sample data. The `caret` package [24] is used to carry out least-squares and elastic-net regression.

A five-fold cross-validation is performed in order to evaluate the training and test errors. First, the entire dataset is randomly partitioned into five sub-data sets. For a sub-dataset $S_i, i \in [1,5]$, the training set includes all samples in $S_j, j \in [1,5], j \neq i$, while S_i is used as the test set. In this way, the dataset is trained and tested five times.

The `train` function from the `caret` package is used to train a `glmnet` model [25] on each of the five data sets. The `glmnet` package performs an elastic-net regularized generalized linear modeling. Elastic-net models have two parameters, the mixing percentage $0 \leq \alpha \leq 1$ and the regularization parameter $0 \leq \lambda \leq \infty$. Note that $\alpha = 0$ is the ridge/L2 penalty, $\alpha = 1$ is the lasso/L1 penalty, and $0 < \alpha < 1$ is a mixture of the two. The parameter λ determines the penalty strength [26]. When both α and λ are set to zero, the model is a least-squares regression without penalization/regularization.

TABLE I: Training Metrics with $\alpha = 0$ & $\lambda = 0$, all block prices Included

Training Set	RMSE	R^2
1	10.350	0.745
2	10.289	0.746
3	10.541	0.735
4	10.341	0.744
5	10.414	0.737
Mean	10.387	0.741

TABLE II: Training Metrics with $\alpha = 0$ & $\lambda = 0$, All block prices not included

Training Set	RMSE	R^2
1	16.838	0.324
2	16.883	0.315
3	16.929	0.313
4	17.057	0.308
5	16.780	0.323
Mean	16.897	0.317

We consider two settings. In the first one, the input of the prediction function o_b^d includes the prices of all blocks in the neighborhood P_B^d, while in the second, the input only includes the price of block b. Comparing the performance of those two settings allows one to determine whether the demand of a given block does indeed depend on prices of surrounding blocks.

First, we set $\alpha = 0$ and $\lambda = 0$ (non-penalized least-squares regression) and compare the training error with all block prices versus without all block prices. Table I shows that the RMSE and R^2 (which is the ratio between the variation explained by the model and the total variation) values obtained when training on the five training sets with $\alpha = 0$, $\lambda = 0$, and all block prices terms included in the occupancy rate prediction function.

When performing the same training procedure with the same five datasets without all block prices, the training performance metrics are significantly worse. Table II shows the RMSE and R^2 values obtained when training on the five training sets with $\alpha = 0$, $\lambda = 0$, and all block prices not included in the occupancy rate prediction function. Comparing Table II to Table I, we observe that including interaction terms in the prediction model lead to lower values for RMSE and higher values for R^2, confirming that including all block prices help improve the fit in a significant way. Fig. 3 illustrates this improvement. The data points in Fig. 3a are closer to the dotted line than in Fig. 3b, showing that predictions match observations more closely when all block prices are incorporated. That implies that the demand in a given block is indeed affected by prices of surrounding blocks.

For the remainder of the case study, all block prices are

TABLE III: Best Parameters

Training Set	α	λ
1	0.015	0.045
2	0.04	0.045
3	0.015	0.045
4	0.03	0.045
5	0.03	0.045
Mean	0.026	0.045

TABLE IV: Training Metrics with Best Parameters

Training Set	RMSE	R^2
1	9.660	0.781
2	9.541	0.783
3	9.829	0.773
4	9.703	0.778
5	9.906	0.768
Mean	9.729	0.776

included in the occupancy rate prediction function. The `train` function is set to tune the α and λ parameters, given a "tuning grid" of values in intervals of 0.005. The parameter values resulting in the lowest training RMSE are chosen as the "best" parameters. Table III shows the best parameters computed for the five training datasets. Table IV shows the RMSE and R^2 values obtained when training on the five training sets with their best parameters. It can be observed from Table IV that the training error is lower and R^2 is higher compared to the results in Table I, indicating that the fit is improved when including the penalties.

After training, the five test sets were tested with their corresponding fit models. The results are shown in Table V. The mean RMSE is smaller than 10 percent. Given that the average occupancy rates range between 0 and 100 percent, we believe that the prediction function is accurate enough to be used for optimizing prices.

D. Finding Optimal Prices

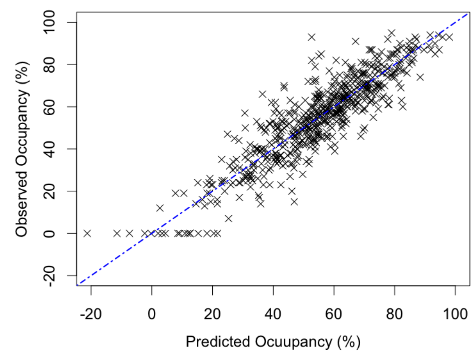
Given a set of prediction functions, the next task is to find the optimal price for each block. For the case study we set the target level $O_T = 80\%$ and we choose the day-type to be weekday and day-time to be open-12pm. We solve the optimization problem defined in Eq. (2) using the Augmented Lagrangian Algorithm [27] which is available in the R package `nloptr`. The output of the optimization problem is a set of optimal prices ranging from \$0.6 to \$5.2. By plugging those prices in the prediction function, we get that the occupancy rates of all blocks are between 79.6% and 80.5% with average of 80.3% (recall that the target level is 80%). Therefore, the case study shows that there exists a set of prices that leads to occupancy rates that are very close to the target level.

V. Conclusion

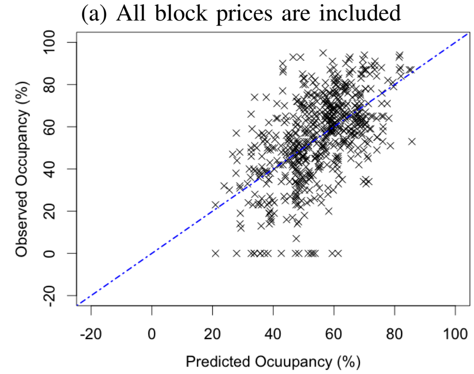
In this paper, we present a machine learning approach for predicting the occupancy rates for a set of parking areas in a neighborhood. We use this learning model to obtain parking prices. These prices are the solution of an optimization problem, which aims to achieve occupancy rates close to a certain target level in each parking area. Our experimental results indicate that taking into account prices of surrounding blocks significantly improves the accuracy of the machine learning model in terms of predicting occupancy. We also show how to set the parameters of the the `glmnet` model in R in order to optimize performance (i.e., minimize the RMSE and R^2 values). The actual implementation and evaluation of the pricing algorithm presented in this study in a real city environment represent interesting areas for future work.

TABLE V: Test Error with Best Parameters

Test Set	RMSE
1	9.188
2	9.323
3	9.515
4	9.413
5	9.581
Mean	9.404



(a) All block prices are included



(b) All block prices are not included

Fig. 3: Observations vs. predictions for occupancy rate. The dashed line represents the 45 degree reference line. Data points are taken from the test set.

Acknowledgment

This research was supported in part by the US National Science Foundation under grants CNS-1117160 and CNS-

1409053. The authors thank Carrie Cramer for her contributions to the initial phases of this research.

References

[1] G. Pierce and D. Shoup, "Getting the prices right," *Journal of the American Planning Association*, vol. 79:1, pp. 67 – 81, Winter 2013.
[2] ParkMe. [Online]. Available: https://www.parkme.com/
[3] Parker. [Online]. Available: http://www.streetline.com/find-parking/parker-mobile/
[4] ParkWhiz. [Online]. Available: https://www.parkwhiz.com/
[5] SpotHero. [Online]. Available: http://spothero.com/
[6] SF*park* Android App Screenshots with Phone. [Online]. Available: http://sfpark.org/resources/sfpark-android-app-screenshots-with-phone/
[7] "SDOT Annual Report 2015 On-Street Paid Parking Occupancy," Seattle Department of Transportation, Tech. Rep., Summer 2015. [Online]. Available: http://www.seattle.gov/transportation/parking/docs/SDOTAnnualReport2015parking.pdf
[8] Sfpark. [Online]. Available: http://sfpark.org/
[9] LA Express Park. [Online]. Available: http://www.laexpresspark.org/
[10] D. G. Chatman and M. Manville, "Theory versus implementation in congestion-priced parking: An evaluation of SF*park*, 2011–2012," *Research in Transportation Economics*, vol. 44, pp. 52–60, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0739885914000067
[11] O. Zoeter, C. Dance, S. Clinchant, and J.-M. Andreoli, "New algorithms for parking demand management and a city-scale deployment," *KDD '14 Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1819–1828, 2014. [Online]. Available: http://doi.acm.org/10.1145/2623330.2623359
[12] F. Li, H. Hindi, D. H. Greene, and Y. A. Sun, "Parking pricing system," U.S. Patent Application 13/666,064, May 1, 2012. [Online]. Available: http://www.google.com/patents/US20140122375
[13] D. Mackowski, Y. Bai, and Y. Ouyang, "Parking space management via dynamic performance-based pricing," *Transportation Research Procedia*, vol. 7, pp. 170 – 191, 2015, 21st International Symposium on Transportation and Traffic Theory Kobe, Japan, 5-7 August, 2015.
[14] Z. S. Qian and R. Rajagopal, "Optimal occupancy-driven parking pricing under demand uncertainties and traveler heterogeneity: A stochastic control approach," *Transportation Research Part B: Methodological*, vol. 67, pp. 144 – 165, Sep. 2014.
[15] S. Clinchant, C. Dance, T. de Ruijter, P. Ghent, and O. Zoeter, "Using analytics to understand on-street parking: the impact of special permit use and the benefit of demand based rates over zones," *22nd ITS World Congress, Bordeaux, France*, Oct 2015. [Online]. Available: http://www.simplesend.com/simple/uploadedimages/000536/files/ITSCongress_smartparking_2015.pdf
[16] M. Manville, *Parking Pricing*. Emerald Group Publishing Limited, 2014, ch. 12, pp. 137–155. [Online]. Available: http://www.emeraldinsight.com/doi/abs/10.1108/S2044-994120140000005019
[17] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, and J. Lin, "Parking slot assignment games," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '11. New York, NY, USA: ACM, 2011, pp. 299–308. [Online]. Available: http://doi.acm.org/10.1145/2093973.2094014
[18] D. Ayala, O. Wolfson, B. Xu, B. DasGupta, and J. Lin, "Pricing of parking for congestion reduction," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '12. New York, NY, USA: ACM, 2012, pp. 43–51. [Online]. Available: http://doi.acm.org/10.1145/2424321.2424328
[19] F. He, Y. Yin, Z. Chen, and J. Zhou, "Pricing of parking games with atomic players," *Transportation Research Part B: Methodological*, vol. 73, pp. 1 – 12, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0191261514002203
[20] Y. Geng and C. G. Cassandras, "New smart parking system based on resource allocation and reservations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1129–1139, 2013.
[21] SF*park* On-Street Pricing Policy. [Online]. Available: http://sfpark.org/resources/sfpark-on-street-pricing-policy/
[22] SF*park* Rate Adjustment Spreadsheets. [Online]. Available: http://sfpark.org/resource-category/adjustments-resources/
[23] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015. [Online]. Available: https://www.R-project.org
[24] caret. [Online]. Available: https://github.com/topepo/caret/
[25] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010. [Online]. Available: http://www.jstatsoft.org/v33/i01/
[26] T. Hastie and J. Qian. (2014) Glmnet Vignette. Stanford. [Online]. Available: http://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html
[27] A. R. Conn, N. I. Gould, and P. Toint, "A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Numerical Analysis*, vol. 28, no. 2, pp. 545–572, 1991.