SAE INTERNATIONAL®

Smart On-Street Parking System to Predict Parking Occupancy and Provide a Routing Strategy Using Cloud-Based Analytics

2017-01-0087
Published 03/28/2017

Jiaqi Ma, Erin Clausing, and Yimin Liu
Ford Motor Company

CITATION: Ma, J., Clausing, E., and Liu, Y., "Smart On-Street Parking System to Predict Parking Occupancy and Provide a Routing Strategy Using Cloud-Based Analytics," SAE Technical Paper 2017-01-0087, 2017, doi:10.4271/2017-01-0087.

Abstract

It is estimated that up to 30% of traffic in cities is due to drivers searching for parking. Research suggests that drivers spend an average of 6-14 minutes looking for an available space in London. This increases individual stress levels as well as congestion and pollution. Parking Guidance Systems provide an effective way to reduce parking search time by presenting drivers with dynamic information on parking. An accurate prediction and recommendation analytics algorithm is the key part of the system combining real time cloud-based analytics and historical data trends that can be integrated into a smart parking user application. This paper develops a prediction algorithm based on transient queuing theory and Laplace transform to predict parking occupancy thus predicting open parking locations.

Introduction

Many cities face the challenges of an increasing population and an increasing rate of private vehicle ownership. The resulting rise in congestion, lack of parking, and pollution cause complicated problems for city managers. For example, in Beijing there are 5,705,000 vehicles but only 1,992,822 parking spots in 2016 [1]. In Beijing average congestion time in 2015 increased to 3 hours from just under 2 hours in 2013 [2]. But there are also many opportunities to leverage modern technologies to help city officials manage parking and traffic efficiently. In the future, smart cities can integrate private and public transportation with smart parking systems to help citizens to plan their journeys. This paper will first discuss a parking guidance system which integrates the front-end and back-end together. The system will use predictive algorithms to recommend locations where a driver is likely to find open parking spaces. The core of the system is the analytics engine that predicts parking availability based on historic and real time data sources. Second, this paper will discuss the queuing methods applied in the engine to estimate demand for parking space by location, bay type, parking rules, and time. (In London sections of on-street parking are called bays. A bay usually contains 4-8 parking spaces.) Finally, the paper will further explore a routing strategy which gives the best route to check available bays and while optimizing both travel time and customer preferences. A simulation method is applied to evaluate the system performance and robustness.

A number of studies have focused on real-time parking availability estimation. Most of the studies describe an infrastructure-based method of detecting parking availability. Some of the parking solutions use ultrasonic sensors mounted on vehicles [3] and some, such as San Francisco and Westminster, use sensors embedded in roads to capture the real time parking occupancy[4][5]. These systems can generate accurate high-quality data to evaluate the effectiveness of parking solutions, but there are huge costs involved in the installation and maintenance. The vehicular ad hoc networks (VANETs) approach provides a vehicle information exchange network and then broadcasts parking availability. However, this approach has a huge latency issue. Also, the success of VANETs technology depends on whether a large percentage of users in a given market adopt the system [6][7]. Wireless technology (WSNs) and smart phone based technology also have huge data cleaning and signal reliability issues[8][9].

In contrast, mobile phone based app payment systems are becoming more and more popular in many markets. Almost every parking meter has the technology to enable the user to pay for parking using a smart parking app. For example, the Park Right app can be used to pay in the Westminster borough of London [5], the Pay by Phone app can be used to pay in the Islington borough of London [10], and Park Detroit can be used to make payments in the Detroit, Michigan area. [11] Since the deployment and maintenance costs of these systems are low, it is reasonable to assume that in the future the majority of urban parking will be managed using parking apps [8]. This paper proposes a parking guidance algorithm which uses only parking payment data. The system can provide real time parking probability and occupancy level estimates.

Queuing theory has a long history. It has been used in a large number of applications in many different industries such as telecommunication, transportation, logistics, communication, and manufacturing. For example, queuing theory is used to determine how many servers are optimal for computer networks [12]. But there are few applications that use queuing theory to optimize parking systems. For the real time parking estimation, queuing transient behavior can explain the continuous system behaviors under a wide variety of conditions and can integrate different types of data sources into the system.

For the proposed parking guidance system, we aim to give customers the best parking solution in terms of parking availability, parking time, driving time, walking time, price and so on. The salesman method described in the paper is the best solution to optimize checking for available parking.

Parking Guidance Systems

The parking guidance system includes these features:

1. Business Logic Server – will assist the resident in finding a suitable parking space based on their profile preferences, the real-time parking situation in their target locale, and their GPS location.
2. Smart Parking Algorithm – based on a request from a user to park at a given destination the engine will compare both real time and historical data inputs from that location to select a 'best fit' recommendation for the driver.
3. Smart Parking App – will allow drivers to poll for available spaces at their chosen destination. While driving the application will keep them apprised of the parking situation and dynamically re-route them if spaces become occupied.

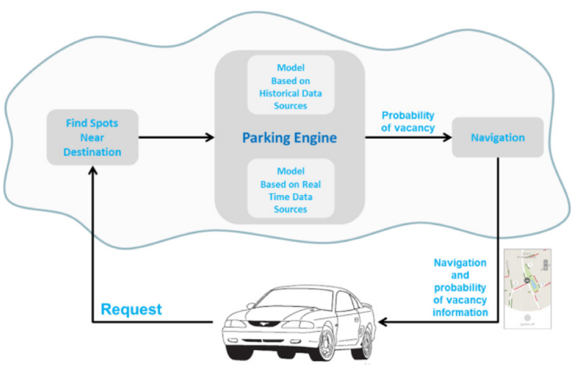

Figure 1. Parking guidance system

Method

In this section two methods are described. First, a queuing theory model based on a transient probability model was developed to predict the probability to park as well as the future occupancy level. Given the historical arriving rate and parking rate for each bay, the system will predict future availability. Although the probability of finding a parking space is important to the customer, the driving time from the origin to the parking spots, the walking time from parking spots to the final destination and unit cost of parking are also very important. So in the second part, the optimization of all of these factors was added to the system to obtain the optimal routing solution algorithm. The recommendation of parking bays as well as the optimal order in which bays should be checked for availability will be outputs from the proposed method.

Parking Availability Prediction Algorithm

Queuing systems can be described as entities (servers) that are called upon by particular entities (users) who ask for services. In each serving system we can specify the arrival process, the service process and one or more service stations or servers. There are many papers that have developed a Markov queueing probability model for a specific problem.

The parking problem can be defined as queuing problem with the FIFO (first in, first out) rule. Car driving-in behavior will add one element to the queue, while car driving-out behavior will take away one element. The capacity of the queue is the capacity of the bay. The occupancy of the parking bay is the number of vehicles in the bay at a specific time. Figure 2 shows an example of bay with a capacity of 6. During this time period a car driving into the bay increases the occupancy by 1, whereas a car driving out the bay decreases the occupancy by 1.

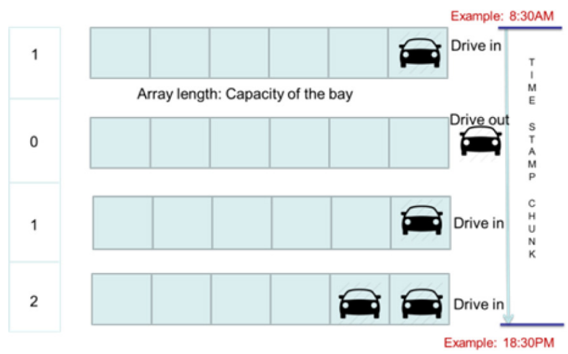

Figure 2. Parking queueing explanation

Let us suppose that drivers are parking at the times $\tau_1, \tau_2, \ldots, \tau_n, \ldots$, where the inter-arrival times $\theta_n = \tau_n - \tau_{n-1}$ ($n = 1,2, \ldots ; \tau_0 = 0$) are identically distributed, independent, positive random variables with the distribution function

$$P\{\theta_n \leq x\} = F(x) \tag{1}$$

There are m available parking spots in each bay. The parking event is successful if the driver finds an available spot. If all of the parking spots are occupied, any incoming drivers will be unsuccessful. It is assumed that the parking times are identically distributed, independent, positive random variables with distribution function $H(x)$ and independent of the input process $\{\tau_n\}$.

Let us denote by $\zeta(t)$ the number of occupied spots at the instant t. The system is said to be in state E_k at the instant t if $\zeta(t) = k$. Define $\zeta_n = \zeta(\tau_n - \tau_0)$, that is ζ_n is the number of occupied spots immediately before the arrival of the n^{th} car. Let $P\{\zeta_N = k\} = P_k^{(n)}$, $P\{\zeta(t) = k\} = P_k^t$

If we assume that the driver parking times are distributed exponentially, the CDF is

$$H(x) = \begin{cases} 1 - e^{-\mu x} & if\ x \geq 0 \\ 0 & if\ x < 0 \end{cases} \tag{2}$$

And if the arrival process assumed to be Poisson with density λ

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & if\ x \geq 0 \\ 0 & if\ x < 0 \end{cases}$$

(3)

Model Explanation

Continuous-Parameter Markov Chains

Considering a continuous-parameter Markov chain {X(t), tϵT} for T = {t: 0 ≤ t < ∞}, for any time s > t > u ≥ 0 and state i and j:

$$p_{ij}(u, s) = \sum_r p_{ir}(u, t)p_{rj}(t, s)$$

(4)

Where $p_{ij}(u, s)$ is the probability of moving from state i to j in the time beginning at u and ending at s, and the summation is over all states in the chain.

$$\frac{dp_{ij}(t)}{dt} = -q_j p_{ij}(t) + \sum_{r \neq j} p_{ir}(t)q_{rj}$$

(5)

$$p'(t) = p(t)Q$$

(6)

Where p(t) is the vector ($p_0(t)$, $p_1(t)$, $p_2(t)$, … .), p′(t) is the vector of its derivatives.

The intensity matrix of the Markov chain is:

$$Q = \begin{pmatrix} -q_0 & q_{01} & q_{02} & \cdots \\ q_{10} & -q_1 & q_{12} & \cdots \\ q_{20} & q_{21} & -q_2 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

(7)

Since the state space of the queue model is composed of non-negative integers (representing the number of cars present), the problem can be categorized as a continuous-parameter (time) Markov chain. In some queue models it also has additional birth-death property that the net change across an infinitesimal time interval can never be other than -1, 0, or +1. So

$$q_{n,n+1} = \lambda_n$$

(8)

$$q_{n,n-1} = \mu_n$$

(9)

$$q_{rj} = 0 \quad \text{(elsewhere)}$$

(10)

$$q_n = \lambda_n + \mu_n$$

(11)

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & \cdots & \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & \cdots & \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \cdots & \\ \cdots & \cdots & \cdots & \cdots & \end{pmatrix}$$

(12)

If the interarrival time and service times are distributed, then $\lambda_j = \lambda$ and $\mu_j = \mu$, where the average interarrival time is $\frac{1}{\lambda}$ and the average service time is $\frac{1}{\mu}$

The birth-death process equations become

$$\frac{dp_n(t)}{dt} = -(\lambda + \mu)p_n(t) + \lambda p_n(t) + \mu p_{n+1}(t)$$

(13)

$$\frac{dp_0(t)}{dt} = -\lambda p_0(t) + \mu p_1(t)$$

(14)

Steady State M/M/c/C Model

In the parking problem, we assume the arrivals follow a Poisson process, parking time is exponentially distributed, and there are c servers (c parking spots for each bay). It is basically a M/M/c/C multi-server queueing mode which is a homogeneous Markov model with exponentially distributed interarrival and parking times. Arrivals occur at rate λ according to a Poisson process moving the process from states *i* to *i+1*. In this case, the arrival rate is the customer parking arrival rate. Vehicle parking times are service times which have an exponential distribution with parameter μ. And parking bay capacity is denoted as c servers which serve from the front of the queue. If there are less than c jobs, some of the servers will be idle meaning the bay is not full. Also for parking problem there is no space to wait in the line. At any time the number allowed in the system is equal to the parking bay capacity.

The intensity matrix for Markov chain in M/M/c/C model is as follows:

$$Q = \begin{Vmatrix} -\lambda & \lambda & & & & \\ \mu & -(\lambda+\mu) & \lambda & & & \\ & 2\mu & -(\lambda+2\mu) & \cdots & & \\ & & \cdots & & & \\ & & & & -(\lambda+(c-1)\mu) & \lambda \\ & & & & c\mu & -c\mu \end{Vmatrix}$$

(15)

For example, for a bay of capacity 6 there are 7 possible states. The Markov model of the 7 possible system states is as follows

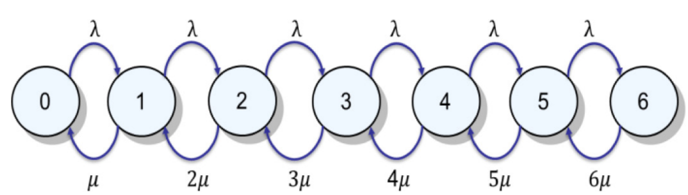

Figure 3. Markov chain of the 6 possible bay capacities

Steady state multi-server probabilities can be solved by setting the derivative from Eq. (1) equal to 0, denoted as

$$p_n = \frac{(\lambda/\mu)^n}{n!} \sum_{i=0}^{c} \frac{\lambda^i}{\mu^i i!} (0 \leq n \leq c)$$

(16)

Erlang's loss formula which is the probability of a full system at any time in the steady state:

$$p_c = \frac{\frac{r^c}{c!}}{\sum_{i=0}^{c} \frac{r^i}{i!}} \; (r = \lambda/\mu)$$
(17)

Transient Model of the Parking Problem

If we know that there are i vehicles at time t = 0 in the parking bay, we can predict the probability of k spots occupied at time t, $\zeta(t) = k$. The probability is denoted as $P_{ik}(t)$

$$P_{ik}(t) = P\{\zeta(t) = k | \zeta(0) = i\}$$
(18)

$$Let\ P(t) = \| P_{ik}(t) \| \qquad (i, k = 0,1, \dots, c)$$
(19)

$$and\ Q = \| q_{ik} \| \quad (i, k = 0,1, \dots, c)$$
(20)

$$P'(t) = QP(t) = P(t)Q$$
(21)

The Laplace Transform of Equation $P_{ik}(t)$ is denoted as $\pi_{ik}(s)$

$$\pi_{ik}(s) = \int_0^\infty e^{-st} P_{ik}(t) dt \qquad (\Re(s) > 0)$$
(22)

Where s is a complex number frequency parameter in the Laplace Transform and

$$\Pi(s) = \| \pi_{ik}(s) \| \; (i, k = 0,1, \dots, c)$$
(23)

$$s\Pi(s) - I = Q\Pi(s) = \Pi(s)Q$$
(24)

$$\Pi(s) = [sI - Q]^{-1}$$
(25)

$$|sI - Q| = (-1)^{c+1} \sum_{j=0}^{c} \binom{c}{j} \lambda^{c-j} s(s+\mu) \dots (s + j\mu)$$
(26)

If the initial condition is 0, that is $\zeta(0) = 0$, then the probability there are k vehicles in the spot at time t is given by the Laplace transform of $P_k(t) = P\{\zeta(t) = k\}$ $(k = 0,1, \dots, c)$ which is given in Takács's book as

$$\int_0^\infty e^{-st} P_k(t) dt = \sum_{r=k}^{m} (-1)^{r-k} \binom{r}{k} \beta_r(s)$$
(27)

$$\beta_r(s) = \frac{\frac{1}{s + r\mu} \sum_{j=r}^{c} \binom{c}{j} \frac{(s + r\mu) \dots (s + j\mu)}{\lambda^{j+1-r}}}{\sum_{j=0}^{c} \binom{c}{j} \frac{s(s+\mu) \dots (s + j\mu)}{\lambda^{j+1}}}$$
(28)

If there are c vehicles in the spot, the bay is fully occupied and the Laplace transform of the probability is

$$\pi_{0c}(s) = \frac{\lambda^c}{\sum_{j=0}^{c} \binom{c}{j} \lambda^{c-j} s(s+\mu) \dots (s + j\mu)}$$
(29)

$$P_{ik}(t) = \mathcal{L}^{-1}\{\pi_{ik}(s)\}(t) = \frac{1}{2\pi i} \lim_{T \to \infty} \int_{r-iT}^{r+iT} e^{st} \pi_{ik}(s) ds$$
(30)

In this section, we consider the parking problem as a closed-loop continuous system. The response depends on the initial value and will fluctuate as time changes, but the system will stabilize eventually. If, at any time, the system receives information about the real time occupancy of the parking bay, different initial values will be input into the system and the probabilities will be recalculated to reflect the updated information. In this way the system dynamics problem is solved in analytical way but yet is flexible enough to utilize real time information.

Parking Bay Checking Order Optimization

Assume a customer departs location *O* and would like to park at destination *d*. There are *k* bays available within certain radius $\{b_1, b_2, \dots, b_k\}$ of d and the probability of being able to park in those bays is $\{p_1, p_2, \dots, p_k\}$. The customer needs to check the parking bays in an optimal order given the probability of vacancy, the walking distance from the destination and other customer preferences. The need to check bays for vacancy in an optimal order is most important in parking areas with a low probability of vacancy. The optimized checking order and routing optimization algorithm will not only save time and energy but can also be customized to individual preferences. The objective function is to minimize the cost function that the customer cannot park at the first selected spot and must move on to second one and so on until a full trajectory has been formed. The cost function is subject to several constraints such as minimizing walking time, trip time and/or the number of stops. The cost function is defined as follows:

$$\begin{aligned} c(o, b_1, t) = t_{ob_1} + p(t_{ob_1}) \\ \times [t_{b_1 d} \times w1 + P(b_1) \times t_{stay} \times w2] \\ + [1 - p(t_{tot})] * D \end{aligned}$$
(31)

where $p(\mathrm{t}_{\mathrm{ob}_1})$ is the probability that the spot is still available after time $\mathrm{t}_{\mathrm{ob}_1}$, $w1$, $w2$, $w3$ are weight factors of walking time, price and driving time, t_{stay} is the user-defined staying time and $P(b_1)$ is the unit price of bay b_1, and D is a time penalty taken if parking bay b_1 is not available.

Figure 4 depicts a customer who will depart from location O and would like to park at destination d when there are three bays available. There are six possible combinations of orders the bays could be checked for availability :{b1, b2, b3}, {b2, b1, b3} and so on. Here is the example of one of the possible checking order

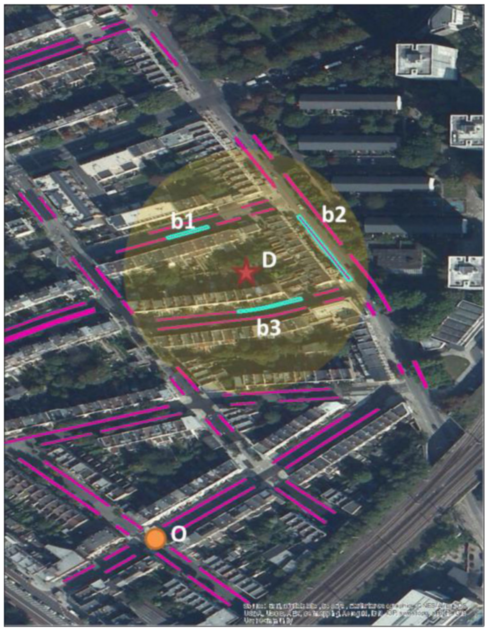

Figure 4. Example of origin and destination with 3 bays

Possibility 1: b1 is checked first and b1 is free after time t_{ob1}

$$c(o, b_1, t) = t_{ob_1} + p(t_{ob_1}) \times [t_{b_1 d} \times w1 + P(b_1) \times t_{stay} \times w2] \quad (32)$$

Possibility 2: b1 is checked first but b1 is not free after time t_{ob1} and therefore bay b2 is checked and after time

$t_{ob1} + t_{b_1 b_2}$ b2 is free.

$$c(b_1, b_2, t) = [1 - p(t_{ob1})] \times \{t_{b_1 b_2} + p(t_{ob_1} + t_{b_1 b_2}) \times [t_{b_2 d} \times \mathbf{w1} + P(b_3) \times t_{stay} \times \mathbf{w2}]\} \quad (33)$$

Possibility 3: b1 and then b2 are checked but neither is available, so b3 is checked.

$$c(b_2, b_3, t) = [1 - p(t_{ob1})] \times [1 - p(t_{ob_1} + t_{b_1 b_2})] \times \{t_{b_2 b_3} + p(t_{ob_1} + t_{b_1 b_2} + t_{b_2 b_3}) \times [t_{b_3 d} \times \mathbf{w1} + P(b_3) \times t_{stay} \times \mathbf{w2}]\} \quad (34)$$

It is assumed that people wouldn't want to check more than 3 available parking bays. Also the computation time is too long if there are a large number of parking bays near the destination. To solve this problem the parking bays are clustered.

The K-means clustering method is applied in the experiment. Each cluster contains a number of available parking bays. Travel time is calculated from each cluster centroid instead of from each individual bay.

Application

The last section will describe the queuing methods applied by the system. The following case studies will describe the two cases that the system will encounter. Case 1 is the case that the parking bay occupancy is 0 at the beginning. The case study gives an analytics solution. The system dynamics of the parking vacancy changes over time are shown. Case 2 is for a real-time online parking solution. In this case the parking occupancy does not start from 0. Updates of the actual parking occupancy can be observed by many different methods any may be input into the system at any time. Using updated information in addition to the historical data will enable the system to give more accurate predictions in the near future.

Case 1

For example, suppose there is a bay with a capacity of 2, an average interarrival time of 20 minutes, and an average parking time of 30 minutes. If the spot is empty at the beginning of the day, what are the probabilities 0.5 hours, 1 hour, 2 hours and 3 hours later that the bay is fully occupied? What is the steady state probability?

In this case, the parameters in the system are $c = 2, \lambda = 3, \mu = 2$

The steady state solution is

$$p_c = \frac{\frac{r^c}{c!}}{\sum_{i=0}^{c} \frac{r^i}{i!}} \left(r = \lambda/\mu\right) = 0.3103 \quad (35)$$

$$\pi_{0c}(s) = \frac{\lambda^c}{\sum_{j=0}^{c} \binom{c}{j} \lambda^{c-j} s(s+\mu) \dots (s+j\mu)} \quad (36)$$

$$\pi_{0c}(s) = \frac{9}{s^3 + 12s^2 + 29s} \quad (37)$$

The inverse Laplace transform $P_{0c}(t) = L^{-1}\{\pi_{0c}(s)\}(t)$

$$\frac{9}{46}\left(-7e^{(-6-\sqrt{7})t} + 6\sqrt{7}e^{(-6-\sqrt{7})t} - 7e^{(\sqrt{7}-6)t} - 6\sqrt{7}e^{(\sqrt{7}-6)t} + 14\right) \quad (38)$$

Table 1. Probability at different times

T(hour)	Probability P_{0c}
t = 0.5	$P_{0c}(0.5) = 0.2201$
t = 1	$P_{0c}(1) = 0.2927$
t = 2	$P_{0c}(2) = 0.3097$
t = 3	$P_{0c}(3) = 0.3103$

$$\mathcal{L}\{P_k(t)\} = \sum_{r=k}^{m} (-1)^{r-k} \binom{r}{k} \beta_r(s) \quad (39)$$

$$\beta_r(s) = \frac{\frac{1}{s+r\mu}\sum_{j=r}^{c}\binom{c}{j}\frac{(s+r\mu)\dots(s+j\mu)}{\lambda^{j+1-r}}}{\sum_{j=0}^{c}\binom{c}{j}\frac{s(s+\mu)\dots(s+j\mu)}{\lambda^{j+1}}} \tag{40}$$

The probability that there are 0 cars observed as a function of time is given by:

$$\pi_0(s) = \frac{s^2+9s+8}{s^3+12s^2+29s} \tag{41}$$

$$\mathcal{L}^{-1}\{\pi_0(s)\} = (21*exp(-6*t)*(cosh(7\text{^}(1/2)*t) + (13*7\text{^}(1/2)*sinh(7\text{^}(1/2)*t))/49))/29 + 8/29 \tag{42}$$

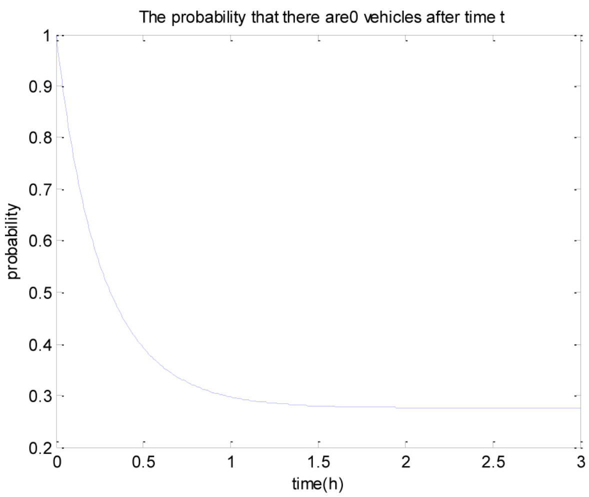

Figure 5. Probability of 0 vehicles versus time

The probability that there is 1 car observed as a function of time is given by:

$$\pi_1(s) = \frac{3(s+4)}{s^3+12s^2+29s} \tag{43}$$

$$\mathcal{L}^{-1}\{\pi_1(s)\} = 12/29 - (12*\exp(-6*t)*(\cosh(7\text{^}(1/2)*t) - (5*7\text{^}(1/2)*sinh(7\text{^}(1/2)*t))/28))/29 \tag{44}$$

The probability that there are 2 cars observed as a function of time is given by the following equations. Since the capacity of the example bay is 2, in this case, the bay is fully occupied

$$\pi_2(s) = \frac{9}{s^3+12s^2+29s} \tag{45}$$

$$\mathcal{L}^{-1}\{\pi_2(s)\} = 9/29 - (9*exp(-6*t)*(cosh(7\text{^}(1/2)*t) + (6*7\text{^}(1/2)*sinh(7\text{^}(1/2)*t))/7))/29 \tag{46}$$

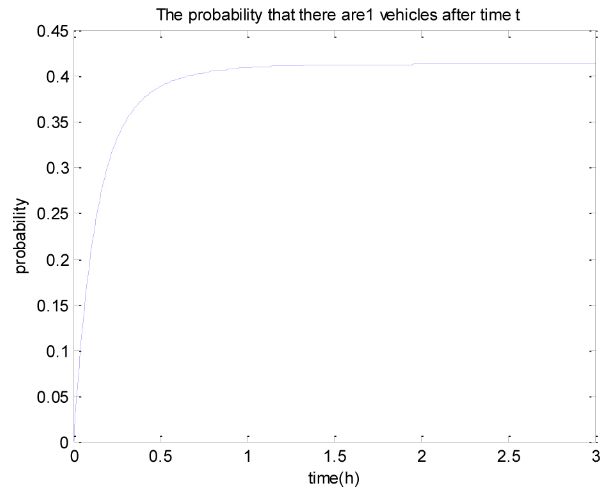

Figure 6. Probability of 1 vehicle versus time

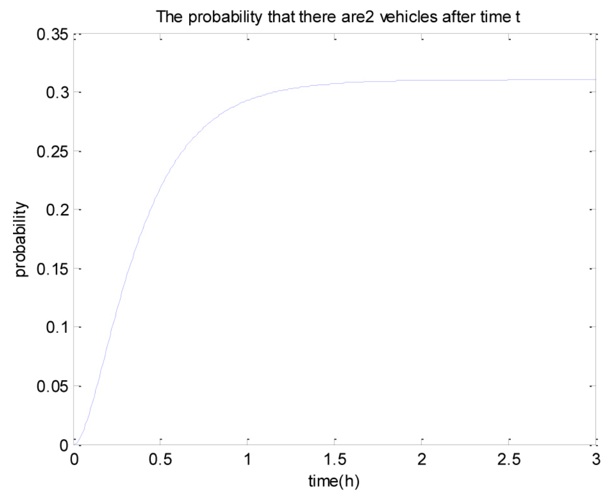

Figure 7. Probability of 2 vehicles versus time

Case 2

The initial condition is the occupancy is not 0. There are i vehicles in the bay at time t (for a bay with a capacity of 2, i can be 0, 1, or 2). What is the probability of k vehicles (k = 0, 1, 2) after time t?

In this case the Q matrix is

$$Q = \begin{bmatrix} -3 & 3 & 0 \\ 2 & -5 & 3 \\ 0 & 4 & -4 \end{bmatrix} \tag{47}$$

$$sI - Q = \begin{bmatrix} s+3 & -3 & 0 \\ -2 & s+5 & -3 \\ 0 & -4 & s+4 \end{bmatrix} \tag{48}$$

$$\Pi_{ik}(s) = [sI-Q]^{-1} = \begin{bmatrix} \frac{s^2+9s+8}{s^3+12s^2+29s} & \frac{3(s+4)}{s^3+12s^2+29s} & \frac{9}{s^3+12s^2+29s} \\ \frac{2(s+4)}{s^3+12s^2+29s} & \frac{(s+3)(s+4)}{s^3+12s^2+29s} & \frac{3(s+3)}{s^3+12s^2+29s} \\ \frac{8}{s^3+12s^2+29s} & \frac{4(s+3)}{s^3+12s^2+29s} & \frac{s^2+8s+9}{s^3+12s^2+29s} \end{bmatrix} \tag{49}$$

The Laplace transform of probability if there are i vehicles initially and k observed at time t is the corresponding element in the matrix $\Pi_{ik}(s)$

For example, if there is 1 vehicle at time 0, the probability of 2 vehicles at time t will be as follows

$$\mathcal{L}^{-1}\left\{\frac{3(s+3)}{s^3+12s^2+29s}\right\}(t) = 9/29 - (9 * exp(-6 * t) * (cosh(7\hat{}(1/2) * t) - (11 * 7\hat{}(1/2) * sinh(7\hat{}(1/2) * t))/21))/29 \tag{50}$$

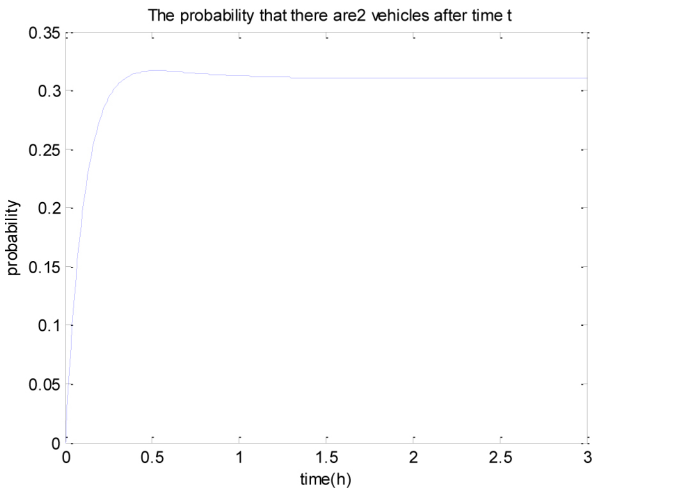

Figure 8. Probability of 2 vehicles versus time when initial number of vehicles is 1

Simulation

Most on-street parking systems accept cash, credit cards, or mobile payment. Very few parking systems use installed sensors to detect availability because of the equipment and maintenance costs. However, as smart phone mobility products such as parking apps become more popular, not only do customers have the opportunity to pay using a smart phone, but cities can collect more accurate and more real time parking on-street parking data. In the simulation, we want to predict the likelihood of parking availability as well as occupancy level using only the time and duration paid for by each person using their phone.

The parking data used in this is study is from Westminster, a borough located in central London. There are around 8000 bays total in Westminster. In this study we are only focusing on a small area of Westminster which contains 785 bays. Figure 9 shows the geographical location of the bays. The red points are the centroids of bays used in the simulation. The size of bubble represents the capacity of the bay.

Figure 9. Geographical locations of Westminster bays used in the study

In the Westminster on-street parking system a sensor was embedded in each parking location. The sensor records the availability every 6 minutes which gives us valuable ground truth data to evaluate the system. For training data, we didn't use the availability data directly. Since we are simulating a phone payment system, the arriving rate is calculated and the maximum stay time is used to estimate parking time to represent data from a payment system. We assume the arriving process follows a Poisson distribution and that the parking time is exponentially distributed.

In this way we simulate historical payment data to predict future vacancy. A year of data was divided into two categories. The 9 months from January to September is used as training data. The three months from October to December is used as testing data. The average occupancy as well as the probability that the bay is fully occupied are predicted.

The Figure 10 shows the different arrival rates and availability for each hour of the day and for each day of the week. Saturdays and Sundays have different patterns than weekdays, so we are use different categories for different hours and whether the day is a weekday or not. The top graph in Figure 10 shows the average interarrival time of parking events. Early morning has longer interarrival time which translates to a slower parking rate. The count of interarrival time represents the car arrival rate. Evening is the peak time for arrival rate.

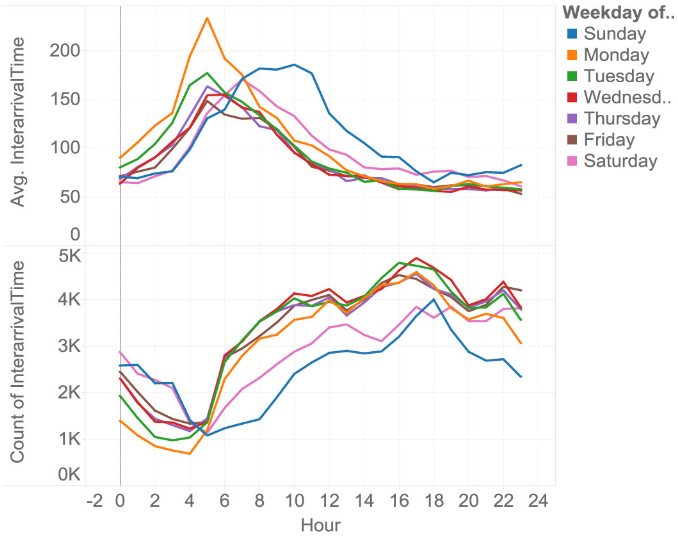

Figure 10. Interarrival time for different hours and days

Figure 11 shows the average availability for different hours and different days of the week. Figure 12 shows an aggregated form of the same information on a map. The lowest parking availability is at noon time. Weekdays have a similar pattern, whereas Saturday and Sunday are quite different.

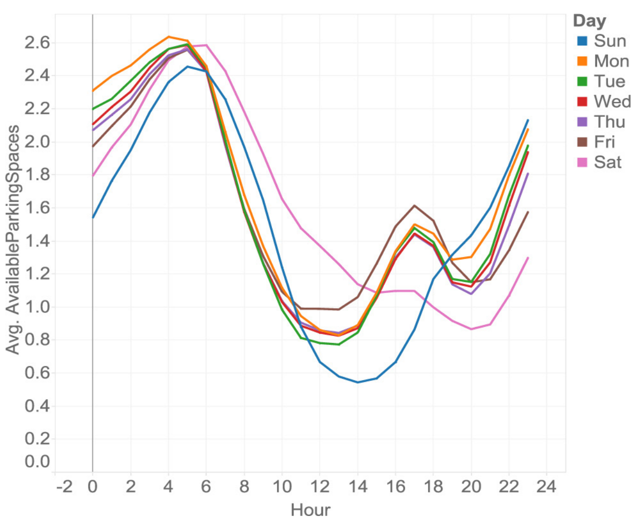

Figure 11. Availability based on hour and day

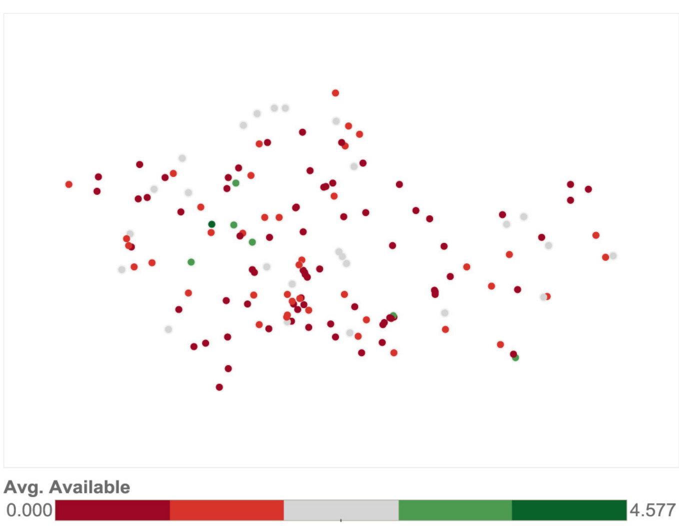

Figure 12. Average availability

Result

In the first experiment, we used 9 months data for training and the rest of the data for testing. In the figure below the upper graph shows the testing occupancy for different days and different hours, and the lower graph shows the prediction results, green for the weekend and orange for weekdays. The whole testing experiment has mean absolute error of 0.9725 spaces.

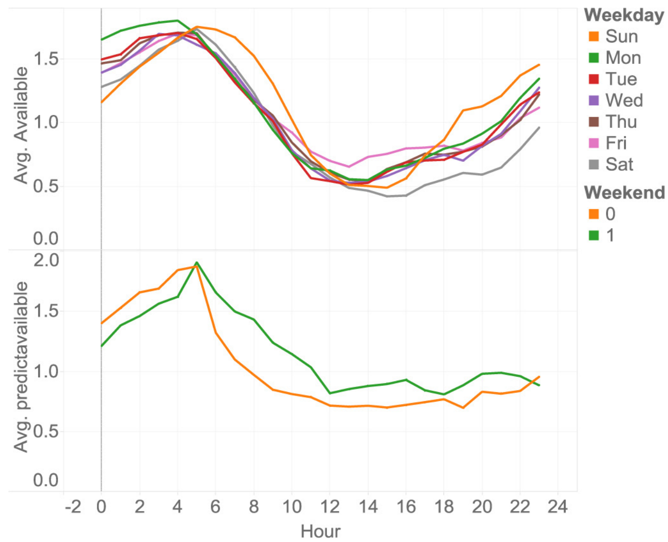

Figure 13. Availability on different hour and weekday

If we only consider the probability of full occupancy, we define a bay as full if the probability exceeds a threshold (50% for example). The system has an accuracy of 69.24% in correctly identifying all the fully-occupied bays in the testing data. Figure 14 shows the probability of full occupancy when the actual availability is 0.

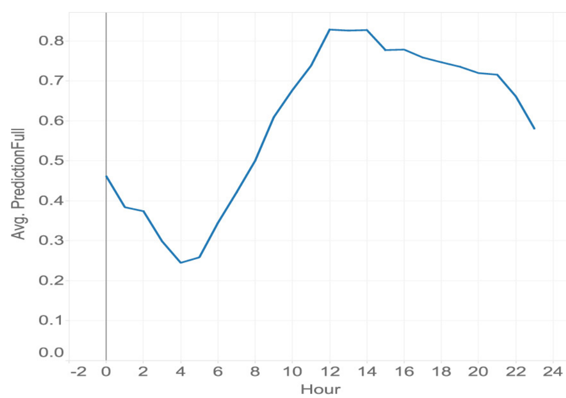

Figure 14. Availability for different hours

In the simulation, we only use maximum allowed stay time to estimate the customer parking duration because we don't have the parking time for each individual. In the real payment situation, we would have additional indicators of paid parking duration. This information will improve system accuracy. The system also predicts the parking probability in near future which is necessary for a real-time parking solution. Given current occupancy, the system is able to predict the occupancy in couple of hours and give the customer a more accurate result. Even without a sensor system, the current occupancy might be observed by a parking enforcement officer, an automatic number plate recognition system, crowd sourcing, or other means. The system has the ability to integrate with other data sources to provide a complete real-time solution. We developed a system combining customer parking data such as payments with parking bay data such as occupancy measurements to develop a robust parking strategy for any real time parking problem.

Summary/Conclusions

Many Smart Cities are starting to integrate new smart mobility services with traditional transportation modes. This paper introduced a parking guidance system based on a Cloud-Based Optimization Engine. The analytics engine applied a Markov queueing probability model to estimate demand for parking space by location, bay type, parking rules and time. In two applications, the algorithm recommended the optimal places to park as well as the best routes to get there. The simulation utilized parking sensor data from a borough of London to evaluate the algorithm and validate its effectiveness. This system provides a better tool for city parking management as well as giving the consumer access to tools that save time, money and stress.

Parking data, especially detailed on-street parking data is not easy to obtain. Once a complete parking data in different payment modes is available, the algorithm can be improved and validated further.

References

1. Beijing Municipal Commission of Transport “Beijing parking spots statistics” http://www.bjysj.gov.cn/xxgk/hygl/tcssgl/201611/t20161107_163733.html, accessed Nov, 2016
2. Beijing-China “Average congestion is 3 hours on weekday in 2015” *Beijing Daily, 2016-01-02*
3. Park, Wan-Joo, . “Parking space detection using ultrasonic sensor in parking assistance system.” *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008.
4. SFpark http://sfpark.org/how-it-works/the-sensors/ accessed Feb, 2016
5. City of Westminster https://www.westminster.gov.uk/parking-bay-sensors accessed Feb, 2016
6. Caliskan, Murat, . “Predicting parking lot occupancy in vehicular ad hoc networks.” *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*. IEEE, 2007.
7. Verroios, Vasilis, Vasilis Efstathiou, and Alex Delis. “Reaching available public parking spaces in urban environments using ad hoc networking.” *2011 IEEE 12th International Conference on Mobile Data Management*. Vol. 1. IEEE, 2011.
8. Nawaz, Sarfraz, Christos Efstratiou, and Cecilia Mascolo . “Parksense: A smartphone based sensing system for on-street parking.” *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013.
9. Basu, Prithwish, and Thomas DC Little . “Networked parking spaces: architecture and applications.” *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*. Vol. 2. IEEE, 2002.
10. PayPoint Company https://www.paypoint.com/en-gb accessed Feb, 2016
11. Park Detroit http://www.parkdetroit.us/ accessed Feb, 2016
12. Wikipedia “Queueing theory” https://en.wikipedia.org/wiki/Queueing_theory accessed Nov, 2015
13. Gross, Donald, and Carl Harris. "Fundamentals of queueing theory." (1998).
14. Takács, Lajos, . *Introduction to the Theory of Queues*. Vol. 584. New York: Oxford University Press, 1962. 174–187
15. White, John A. *Analysis of queueing systems*. Elsevier, 2012.
16. Abatew, J., and Whitt W. "Calculating transient characteristics of the Erlang loss model by numerical transform inversion." *Communications in statistics. Stochastic models* 14.3 (1998): 663–680.

Contact Information

Jiaqi Ma
Data Scientist, Global Data Insight and Analytics
Ford Motor Company
15031 Commerce Drive South
Dearborn, MI 48120
Work phone: (313) 3229330
jma35@ford.com

Acknowledgements

The smart mobility experiment was supported by Ford Motor Company. We thank our colleague Mr. Douglas Nicoll and Mr. Sagie Evbenata for their great support on data acquiring and business communication. Also thanks to our colleague Hao Zhou for making this paper possible.

The Engineering Meetings Board has approved this paper for publication. It has successfully completed SAE’s peer review process under the supervision of the session organizer. The process requires a minimum of three (3) reviews by industry experts.

ISSN 0148-7191

http://papers.sae.org/2017-01-0087