CS221 Fall 2013 Project Report [p-final]

SUNet ID: [markcx, jimywang]
Name: [Xiao Chen, Jim Wang]

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

1 Introduction

According to the Department of Parking and Traffic, San Francisco has more cars per square mile than any other city in the US [1]. The search for an empty parking spot can become an agonizing experience for the city's urban drivers. A recent article claims that drivers cruising for a parking spot in SF generate 30% of all downtown congestion [2]. These wasted miles not only increase traffic congestion, but also lead to more pollution and driver anxiety. In order to alleviate this problem, the city armed 7000 metered parking spaces and 12,250 garages spots (total of 593 parking lots) with sensors and introduced a mobile application called SFpark, which provides real time information about availability of a parking lot to drivers. However, safety experts worry that drivers looking for parking may focus too much on their phone and not enough on the road. Furthermore, the current solution does not allow drivers to plan ahead of a trip.

We alleviate the parking problem by predicting parking availability in the future. A driver should be able to plan where she is parking with high degree of certainty ahead of her trip. We also add parking price prediction because the price varies with demand and a driver may have a certain budget constraint. This report summarizes our work in (i) prediction of future parking and price availability, (ii) parking recommendation given user input about future travel plans, (iii) and simulation of a driver following our recommendation and accounting for uncertainty in parking space availability.

2 Literature Review

Caliskan *et al.* [4] predict parking lot occupancy based on information exchanged among vehicles. They build a mathematical model based on queueing theory and uses a continuous-time homogeneous Markov Model. A simulation was developed in VISSIM, a traffic simulation software, to implement their model. Their prediction algorithm is most effective for prediction times car arrival rate less then 15 minutes.

Felix *et al.* also did real-time prediction based on live user requests [5]. This methodology consists of three subroutines to allocate simulated parking requests, estimate future departures, and forecast parking availability. They use aggrecated approch to make prediction. It yielded small average error availabilities (less than 3% in the case of 1 hour of anticipation).

It is noteworthy that most parking papers emphasize parking prediction. For our project, we place an initial emphasis on prediction, but want to also focus on building a recommendation product and simulating user parking behaviour based on predicted available lots set.

3 Task Definition

We divide our task into three phases: prediction, recommendation, and simulation. First, we predict the number of available spots and the parking price in the future. Specificly, we use July-Aug data (2 months) as training data, and we wish to predict future parking availability from 6AM-10PM in September. Second, we use the results from the first task to help a driver plan her trip by recommending parking spots based on her preferences. These preferences include estimated time of arrival, final destination, max walking distance, max price willing to pay, preference for short walking distance or cheap parking price. Third, we simulating a single driver making parking decisions as a Markov Decision Process (MDP), again following the list of recommended lots (not necessarily in the order listed in Task 2). In particular, there are two types of uncertainties: (i) parking spot not being available, (ii) the driver may decide to visit the next closest lot or the next cheapest lot.

4 Data

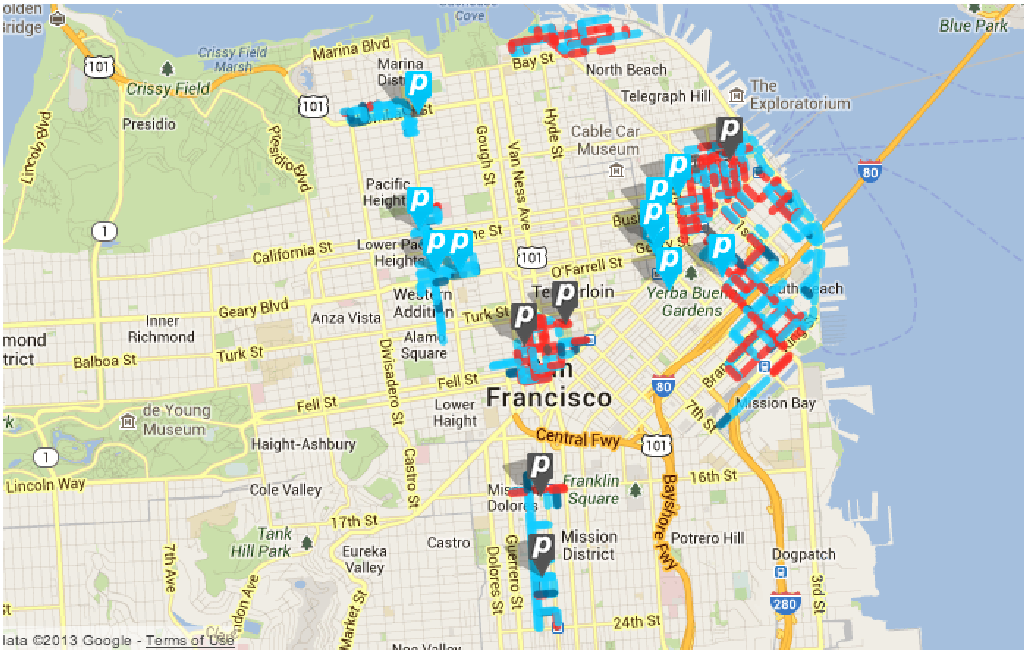


Figure 1: map of parking place

Data has been collected between July 2013 and September 2013 from San Francisco's SFpark API [3]. We use July-Aug data (2 months) as training data, and we wish to predict future parking availability from 6AM-10PM in September. The city provides hourly data for 593 parking lots (Figure 1). We recorded all the public data from the API including timestamp, geolocation (latitude & longitude), place of interest, capacity, parking price, and lot types (on/off street parking). In addition to these inputs which were already collected, we also collected information on key events: Giants games (49ers did not play in Jul-Aug), big concerts, and races. We recognize that traffic map data may capture other small events by marking whether there is traffic jam due to scheduled or unscheduled events. However, traffic APIs only provide real time data, and we are not able to go back in time to scrap the data.

In order to fit input data to model, we have to first clean the data that has been collected from our Elastic Cloud Computer and Relational Database System instances. The raw data files are named by dates, which contain all parking lot information including timestamp, lot id, occupied spots, total spots, price, lot type, etc. One single file has records on 5 minutes time interval for all recorded parking lots in San Francisco on a particular day. We wrote a script to resort the data files by parking lot id. Also we generated a parking lot location dictionary file that stores parking lot id and geolocation. We create a events schedule dictionary file as well to take the events as a related feature for parking avaliability. Originally the amount of raw data files are more than 1.5 GB, after data cleaning work we trim down to 500MB.

Here is a sample file after data cleaning:

```
"1373295894", "935", "348", "745", "1.00"
"1373296194", "935", "355", "745", "1.00"
"1373296494", "935", "365", "745", "1.00"
"1373296795", "935", "376", "745", "1.00"
"1373297095", "935", "385", "745", "1.00"
"1373297395", "935", "391", "745", "1.00"
```

The first column is timestamp, second column Lot ID, third column number of occupied spots, four column total number of spots, and fifth column the parking price. Each line in this file represent a new data point, from which we extract the input, x, and the output y. More specifically, the input is (time, geolocation), and the output is either number of available spots or price, depending on which variable we are predicting. the output of our model includes number of available spots and price, because we predict both future available spots as well as futher price. In the next section, we discuss how to build features from the input x.

5 Prediction

5.1 Approach

5.1.1 Model

The types of features we identified being highly relavant in predicting parking availability and parking price are day, time, event, distance, etc. For day, we set an indicator function for each day in the week. For time, we discretize time into t-min intervals and set an indicator function for each time interval (t_a, t_b) (inclusive on t_a and exclusive on t_b). We allow for combinations of time intervals in order to make the time features more robust (we created a function which can take any t and generate the corresponding features). In our case, we used $t = 10, 30, 60, 120$. Three event features - sports, concert, and race - are also represented by indicator functions. We mark an event feature as 1 if there is an event either happening at the time of interest or will happen within four hours (to account for people arriving early to an event). The single distance feature is a distance in miles between the lot location and the K-means cluster centroid to which it belongs (Figure 2). The same features are used for training parameters to predict parking availability and to predict parking price.

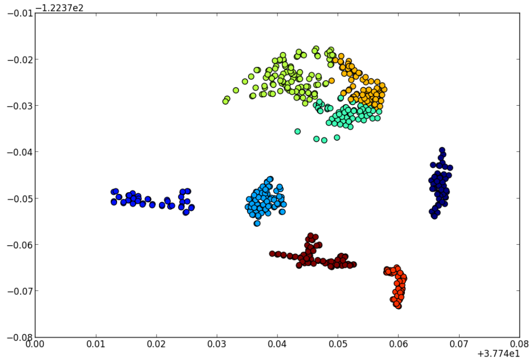


Figure 2: K-mean classify parking lots geo-location

We implement a linear regression on the extracted feature, $\phi(x)$ by stochastic gradient descent. More precisely, we model the number of available spots of parking lot i by

$$h_i(x) = w_i \cdot \phi(x), \tag{1}$$

where w is the weight vector for availability. Likewise, we model parking price by

$$c_i(x) = \theta_i \cdot \phi(x), \tag{2}$$

where θ is the weight vector for availability.

5.1.2 Algorithms

We implement stochastic gradient descent to update the weights for each parking lot:

```
Loop through each parking lot:
  w <-- initialize sparse vector of weights, represented as Counter obj
  Loop through all days Jul 1 through Aug 31:  # each day is a text file
    while line=readLine() # read each line representing a unique timestamp
      phi, y = featureExtractor(line)  # output feature vector and label
```

```
dotProd = dotProduct(phi, w)
for key in phi:
 w[key] += alpha*(y-dotProd)*phi[key]
```

We pick `alpha`, the learning rate, which allows $y - w \cdot \phi(x)$ to approach zero as training continues. If `alpha` is too large, we overshoot and the solution may blow up. If `alpha` is too small, we don't learn fast enough. Currently, we set

$$\texttt{alpha} = \frac{\alpha_0}{t^\beta} \tag{3}$$

. The $\beta \in [0, 1]$, we choose $\beta = 0.5$ and t is iteration steps. A similar analysis can be done for parking price.

5.2 Results

Training on data from Jul-Aug and testing on data from the sample days in the first week in September yielded an average test error of 20% for availability and 2% for price. Error rate at time t is defined as $(y_t - \hat{y_t})/y_t$, where y_t is the actual value and $\hat{y_t}$ is the predicted value. The test error is defined as the average of the error rates. One possisble explanation for the large error rate in availability number is that single lot availability percentage varies wildly: for a 10 spot lot, differing by 1 spot is a 10% error. Nonetheless, while the error rate for the availability number is non-trivial, we decided to focus our attention more on Task 2 and 3 (especially 3) and included uncertainties in our simulation in Task 3 to account, in principle, for the estimate error.

Looking at the computed feature weights, we note that the top features for availability number with positive predictive powers are distance, day of the week, and events. The individual time slot were less predictive. This indicates that (i) there is a regional predictiveness to parking availability (certain regions have more lots available), (ii) major trends or changes happen on a by day basis, possibly with the inclusion of a day-long event.

6 Recommendation

6.1 Approach

We take in the user input information such as estimated time of arrival, final destination, max walking distance, max price willing to pay, preference for shortest distance or cheapest lot. Based on our prediction result we will filter out top 10 best available parking lots. The user preference for distance or price is a factor $P_\alpha \in [0, 1]$, where $P_\alpha = 1$ means user always prefers shortest distance and $P_\alpha = 0$ means user always prefer cheapest lot. We pick the 10 best lots as follows:

```
L <-- set of predicted lots with availability
V = list()
for i in range(10):
  r = rand()
  if r < p:
    V.append(min(L, key=dist))
  else:
    V.append(min(L, key=price))
return V
```

6.2 Results

Here we demonstrate the result for driver looking for parking at either Civic Center or AT& T park, on Sept 3, at 12:30pm. The max distance is set at 0.5 miles and max price at $5. We set $P_\alpha = 0$ or $P_\alpha = 1$. Other results are similar.

```
Destination=Civic Center, pref=price
Hayes St (101-199), Est AvailNum=4, Est Price = $0.08, Dist to dest = 0.38 miles
Mcallister St (500-598), Est AvailNum=6, Est Price = $0.12, Dist to dest = 0.42 miles
Van Ness Ave (201-299), Est AvailNum=1, Est Price = $0.12, Dist to dest = 0.45 miles

Destination=Civic Center, pref=dist
Grove St (2-50), Est AvailNum=2, Est Price = $1.33, Dist to dest = 0.19 miles
Mcallister St (301-399), Est AvailNum=6, Est Price = $1.51, Dist to dest = 0.22 miles
Mcallister St (300-398), Est AvailNum=4, Est Price = $1.51, Dist to dest = 0.24 miles

Destination=AT&T Park, pref=price
3rd St (500-598), Est AvailNum=10, Est Price = $0.00, Dist to dest = 0.34 miles
4th St (501-567), Est AvailNum=4, Est Price = $0.00, Dist to dest = 0.40 miles
4th St (500-598), Est AvailNum=8, Est Price = $0.00, Dist to dest = 0.41 miles

Destination=AT&T Park, pref=dist
King St (100-198), Est AvailNum=1, Est Price = $0.85, Dist to dest = 0.09 miles
3rd St (701-799), Est AvailNum=1, Est Price = $0.46, Dist to dest = 0.11 miles
Townsend St (130-198), Est AvailNum=3, Est Price = $0.42, Dist to dest = 0.11 miles
```

7 Simulation

For the final task, we simulate a driver looking for parking spot as a Markov Decision Process (MDP). Given the driver's input parameters (same as in Section 6), we generate a list of available lots that fit the driver's requirements. Instead of sorting the list based on P_α to provide a recommendation, we simulate a driver visiting lots, where at each lot she can choose to stay or leave. The ultimate goal is to find a lot to stay, but there is some probability (defined later), that she chooses to stay at a lot which turns out to still be full; if this is the case, she is really unhappy and incurs a large cost. On the other hand, she can choose to leave the lot and only incur a small cost. If she chooses to stay and does indeed stay, then she is happy and receives a large reward. The following subsections describe the model in more detail.

7.1 Approach

Let a state capture information about the current lot and the list of lots visited. Then at each state, the driver has two actions, leave and stay, each action leads to a set of successor states (other lots not visited) or the current lot she is parking in. The graphical representation of this model is shown in Figure 3

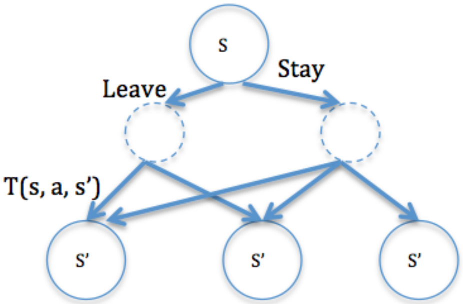


Figure 3: Graph Structure for MDP model

7.1.1 Model

Before describing the MDP model in detail, we define some varibles related to our model formulation.

- P_α is the driver preference which is between $[0, 1]$.
- A is a vector takes binary value on each entry, where $A \in R^n$. It records whether each recommended parking lot is visited or not.
- P_l is the probability that a driver will leave the current parking lot

- d is the index representing current parking lot
- y is a binary variable representing whether the state is a terminal state. 1 means terminal.

To formulate the MDP we have following definition,

- *state* $= (A, d, y)$
- *action* = leave, stay
- *start state* $= (A, d, y)$, where $A = (0, ..., 0), d = -1, y = 0$
- *transition probability*

$$T(s, a, s') = \begin{cases} P_\alpha & \text{if } s' = (A', d', y), \text{where } d' \text{ is the index of the next} \\ & \text{nearest parking lot to user destination, action = leave} \\ 1 - P_\alpha & \text{if } s' = (A', d', y), \text{where } d' \text{ is the index of the next} \\ & \text{cheapest parking lot to user destination, action = leave} \\ 1 - P_l & \text{if action = stay }, s' \text{ is } \textit{end state} \\ P_l P_\alpha & \text{if action = stay, but the driver finds the spot is not available} \\ & \text{and leaves for the next closest lot} \\ P_l(1 - P_\alpha) & \text{if action = stay, but the driver finds the spot is not available} \\ & \text{and leaves for the next cheapest lot} \end{cases}$$

- *rewards*

$$R(s, a, s') = \begin{cases} -1 & \text{if } s' = (A', d', y), \textit{action} = \textit{leave}, y = 0 \\ -10 & \text{if } s' = (A', d', y), \textit{action} = \textit{stay}, y = 0 \\ 100 & \text{if } s' = (A', d', y), \textit{action} = \textit{stay}, y = 1 \end{cases}$$

We give a reward of -1 for leaving because it takes time to move to a new lot, of -10 for staying but being forced to leave (due to full lot) because the driver may be upset after getting her hopes up, and 100 for staying and finding a spot.

An important parameter we need to define in this MDP formulation is P_l. We note that availability number is a Posson process, so we define $P_l = P(X = 0)$ where $P(X) = \frac{\lambda^x e^{-\lambda}}{x!}$ and X is the availability number. So $P_l = e^{-\lambda}$. Intuitively, the higher the predicted availability number, the lower the probability of leaving; the higher the price of the lot, the higher the probability of leaving. Thus, we chose to define

$$\lambda = \max[0, c_1 y_{Avl} - c_2 y_{price}]. \tag{4}$$

We tested different values of c_1 and c_2, and found that $c_1 = 0.5$ and $c_2 = 0.05$ produced intuitive values of P_l. As seen in Figure 4, The probability of leaving depends more heavility on number of available spots than the price. For large number of spots (> 5), the probability is largely independent of price. Further, the probability of leaving decreases fast (exponentially, to be exact), as the number of available spots increases, which makes intuitive sense.

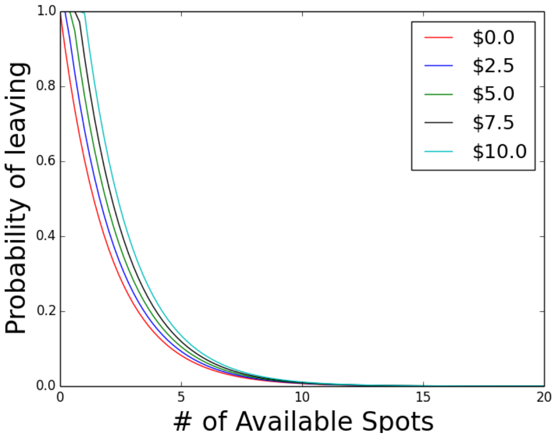


Figure 4: P_l as a function of availability number and price, for $c_1 = 0.5$ and $c_2 = 0.05$.

7.1.2 Algorithms

We use Value Iteration (Bellman, 1957) algorithm to find the optimal value of each state and its corresponding optimal action.

We also analyzed how many states (lots) a driver must visit before she is expected to pick *stay* as the best action. We calculate this expected number is as follows:

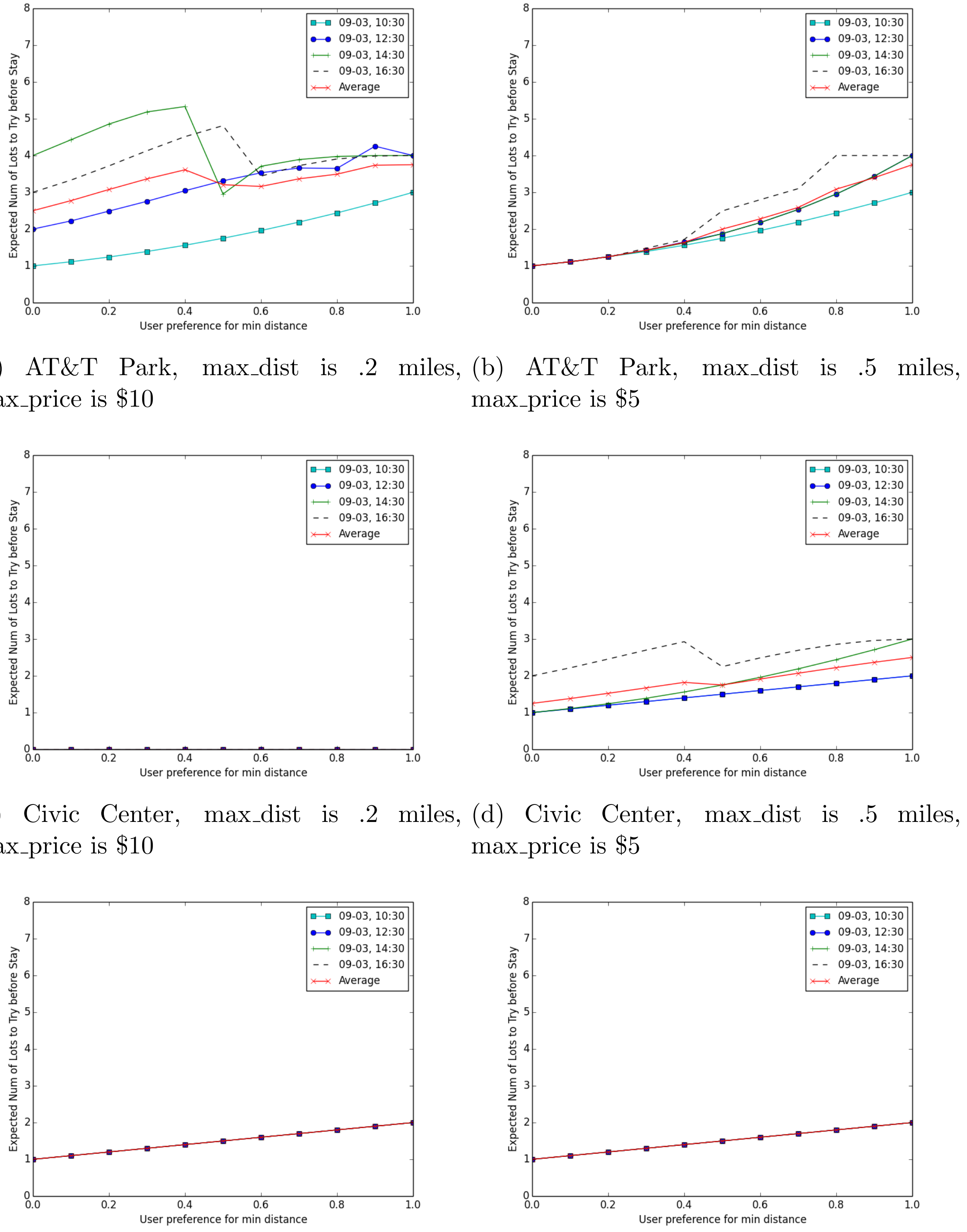
Let $E(s)$ be the expected number of lots to visit, starting at s, before staying.

- If optimal action at s is *stay*, then $E(s) = 0$
- Else $E(s) = 1 + \sum_{s'} T(s, a, s')E(s')$.

If $s' == NULL$, i.e. no more lots to visit, we renormalize all the transition probabilities not accounting for this s'.

7.2 Results

Some sample results are shown in Figure 5. For AT&T Park as user destination (parts (a) and (b)), when the user sets a small distance window and a large price window (max_dist is .2 miles, max_price is $10), we see that the average expected number of visits is around 3. In the morning, it is harder to find parking when one prefers distance. In the afternoon, it is harder to find cheap parking, which makes sense, because (i) the parking price is higher in the afternoon, (ii) the cheap on-street parking are scarce, yet a driver will search for those spots before searching for off-street garage. For this particular input, there are only 12 available lots to search from. The discontinuity observed may be due variations in data for this small data size. Interestingly, for a larger distance window and small price window (max_dist is .5 miles, max_price is $5), the expected number of lots to try goes up, for all



(a) AT&T Park, max_dist is .2 miles, max_price is $10

(b) AT&T Park, max_dist is .5 miles, max_price is $5

(c) Civic Center, max_dist is .2 miles, max_price is $10

(d) Civic Center, max_dist is .5 miles, max_price is $5

(e) Pier 39, max_dist is .2 miles, max_price is $10

(f) Pier 39, max_dist is .5 miles, max_price is $5

Figure 5: Sample example MDP result for various locations, max_dist, and max_price.

times, as P_α increases. This is because when we relax the max_dist requirement, we get 57 more lots than the previous case. Specifically, there are more on-street lots to choose from, and it is more likely that the driver will quickly find her spot, especially when she is not picky about distance. When she is picky, however, we see that the expected number is about the same as in part (a), because she stays at the same lot she would have stayed in part (a).

For Civic Center, in part (c) the expected number always shows 0. This is because for this user input, only 1 lot is available at this time. Thus, the driver either gets the lot or not, either case the expected number is 0, based on the definition given in Section 7.1.2. In part (d), the driver can search among 39 lots. The explanation is similar to part (b). However, there is a kink for the 4:30pm time slot, whch may be explained by not enough data points.

Lastly, for Pier 39, the expected number is the same linear line for both types of user input and for all times. This result is the most surprising. Looking closely at the choices the driver picked, we see that the lots were actually within 0.2 miles and within $5 in price, which explains why parts (e) and (f) are the same.

As seen from these examples, we cannot draw generalizing conclusion about the expected number of visits. The expected number of visits can vary over time, over geography, and over user preferences.

8 Conclusion

In conclusion, we are able to predict *individual* lot with test error below 20% for availability and 2% for price. This is an improvement from our baseline (not discussed here) which was 20% averaged over *all* lots and all time. Given user input information on future travel and personal preferences, we are able to recommend the best available lots. Further, we simulated a driver looking for parking as a Markov Decision Process and analyzed the expected number of visits a driver would have to make before parking.

References

[1] Shoup, Donald. "The High Cost of Free Parking", SFGate Article, published 3 Jun, 2005, accessed 21 Oct, 2013. `http://www.sfgate.com/opinion/openforum/article/The-high-cost-of-free-parking-2630493.php`

[2] Richtel, Matt. "Now, to Find a Parking Spot, Drivers Look on Their Phones", NYTimes Article, published 7 May, 2011, accessed 21 Oct, 2013. `http://www.nytimes.com/2011/05/08/technology/08parking.html?pagewanted=all&_r=0`

[3] SFpark. Last accessed 21 Oct, 2013. `http://sfpark.org/`

[4] Caliskan, M.; Barthels, A.; Scheuermann, B.; Mauve, M., "Predicting Parking Lot Occupancy in Vehicular Ad Hoc Networks," Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th , vol., no., pp.277,281, 22-25 April 2007

[5] Felix Caicedo, Carola Blazquez, Pablo Miranda, Prediction of parking space availability in real time, Expert Systems with Applications, Volume 39, Issue 8, 15 June 2012, Pages 7281-7290, ISSN 0957-4174, http://dx.doi.org/10.1016/j.eswa.2012.01.091.