

# Parking Occupancy Prediction and Pattern Analysis

Xiao Chen  
markcx@stanford.edu

**Abstract**—Finding a parking space in San Francisco City Area is really a headache issue. We try to find a reliable way to give parking information by prediction. We reveals the effect of aggregation on prediction for parking occupancy in San Francisco. Different empirical aggregation levels are tested with several prediction models. Moreover it proposes a sufficient condition leading to prediction error decreasing. Due to the aggregation effect, we would like to explore patterns inside parking. Thus daily occupancy profiles are also investigated to understand travelers behavior in the city.

## I. 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 [3], 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 wish to tackle the parking problem by (i) predicting the occupancy rate, defined as number of occupied parking spots over total number of spots, of parking lots in a zone given a future time and geolocation, (ii) working on aggregated parking lots to explore if there is estimation error reduction pattern in occupancy prediction, (iii) classifying daily parking occupancy patterns to investigate different travel behavior at different region.

## II. PROBLEM STATEMENT

It is more useful to predict occupancy by zone than by individual lots, since drivers generally just want to find parking within a certain proximity (usually less than 10 minutes walking distance is affordable). Thus we combine part (i) and (ii) together for prediction task. We plot the aggregated parking spots occupancy curve for intuitive understanding. It is obvious (Fig 1) that aggregation of parking lots reduces the variability of occupancy ratio illustrate the fact that it is easier to predict occupancy with higher aggregation level. Mean Absolute Percentage Error (MAPE) is used commonly to measure the forecast accuracy. Consider two time series

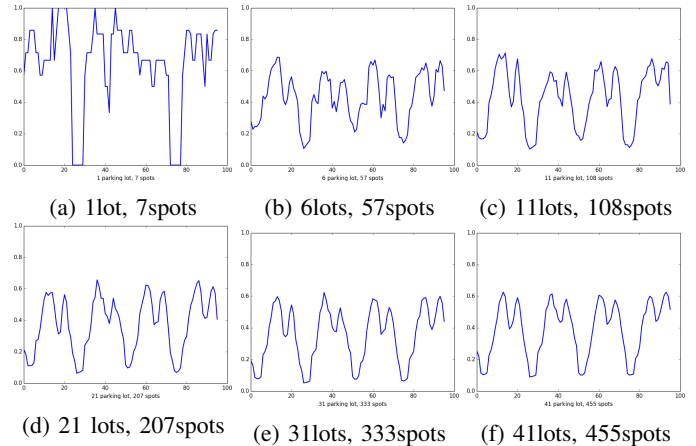


Fig. 1: Hourly parking occupancy ratio for various aggregation levels. A single parking lot with a few spots has little pattern to be exploited. Aggregating more parking lots generates smooth pattern so that it can be more predictable.

$y(t)$  where  $t = 1, \dots, n$  together with its predicted series  $\hat{y}(t)$ . The MAPE is defined as

$$MAPE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n \left| \frac{y(t) - \hat{y}(t)}{y(t)} \right| \quad (1)$$

We use MAPE in this project to measure our prediction performance.

## III. EXPERIMENT SETUP

### A. Data Description

The data used in this paper is collected from the SFPARK between July 1, 2013 and December 29, 2013. Original data is captured every 5 minutes including datetime, geolocation(latitude, longitude), place name, lot capacity, occupied number of spots, parking price, lot type (on/off street parking). For analyzing daily parking occupancy, we take the hourly average occupancy rate as our refined dataset. Therefore occupancy for each day has 24 data points.

### B. Feature Selection

The types of features we identified being highly relevant in predicting parking availability are day, time, event, distance, parking price, etc. For day, we set an indicator function for each day in the week. For time, we discretize time into 24 intervals and set an indicator function for each time interval  $(t_a, t_b)$  (inclusive on  $t_a$  and exclusive on  $t_b$ ). The single distance feature is a distance in miles between the lot location and the cluster centroid to which it belongs (calculated

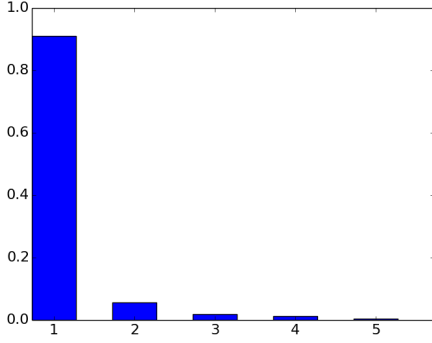


Fig. 2: PCA on timeseries, shown first 5 component

by K-means we will mention in next section). 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). For example, the time stamp corresponds to Monday, 08:00 am. Also suppose that a major race is starting at 10:00 am (within four hours of 08:00 am) that same day. Moreover we check the PCA on timeseries to discover occupancy feature in another space. And we also input the first 3 PCA features because it already explains over 95% variance. Thus, a feature vector for time  $t$  can be expressed as  $v_t = [Monday, 8 : 00 - 9 : 00, distance, race, occupancy(t-1), \dots, occupancy(t-i), PC1, PC2, PC3]^T$  where  $i$  is the time step lag (we pick previous 24 hours occupancies, i.e  $i = 24$ ),  $PC1$  stands for first principle component, similarly for  $PC2, PC3$ .

### C. Model Selection

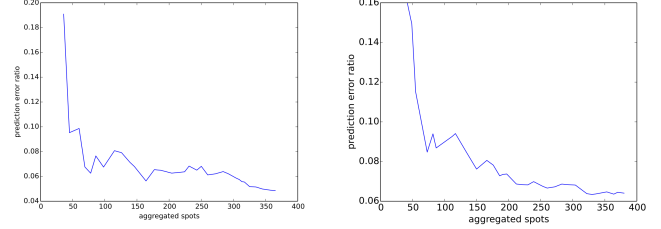
1) *ARIMA*: We implement  $ARIMA(2, 0, 1) \times (1, 1, 0)_{24}$  model to investigate the relationship between prediction error and aggregation level. Here seasonal factor is set to be 24 due to the daily period. We tune the parameters by picking the lowest AIC value among different  $p, d, q, P, D, Q$  combinations [4]. And performance of MAPE are computed in TABLE I. Aggregation prediction error curve is shown in Figure 3a. Originally the prediction error goes over 15%. But it drops significantly after aggregating more than 100 spots, the aggregation effect become flatten out when adding over 160 spots.

2) *Linear Regression (OLS)*: Given  $y(t)$  is the time series, we input previous time step value together with the dummy variable to indicate current time is allocated at what hour in the day, and what day of the week to fit the value at current time. The  $y(t)$  can be expressed as

$$y(t) = c_0 + \sum_{i=1}^{24} \beta_i \mathbf{1}\{HoD(t) = i\} + \sum_{j=1}^5 \theta_j \mathbf{1}\{DoW(t) = j\} + \alpha y(t-1) + \epsilon(t) \quad (2)$$

, where  $HoD()$  is a function takes in time  $t$  and return the hour of the day (from 1 to 24),  $DoW()$  is a function return the day of week (from Monday to Friday).  $\beta, \theta$  are dummy variables,  $\alpha, c_0$  are corresponding parameters. We fit the model by getting least square error of  $\epsilon(t)$

The  $R^2$  value for Linear Regression Model is over 92%. After aggregation of 120 spots, it gives MAPE close to 8%. Figure 3b show the MAPE performance by aggregation.



(a) ATT park region 1 hour ahead, ARIMA(2, 0, 1)  $\times$  (1, 1, 0)<sub>24</sub> (b) ATT park region 1 hour ahead, OLS

Fig. 3: MAPE plot of accumulated parking spots

3) *Support Vector Regression*: Let a vector  $\mathbf{x}_k(t)$  holds all sample data from  $y(t-k+1), y(t-k+2), \dots, y(t)$ , where  $k$  is the vector length (also known as time lags). SVR builds a nonlinear map between  $y(t)$  and  $\mathbf{x}_k(t-1)$  as follows:

$$y(t+1) = w^T \phi(\mathbf{x}_k(t)) + b + \epsilon(t) \quad (3)$$

[5] [6]. The kernel function  $\phi$  is chosen particularly depending on problem. Here the kernel function is radial basis function. We pick previous 24 sample observations to generate current prediction value, which is  $y(t)$  relates to  $y(t-24), y(t-23), \dots, y(t-1)$ . It gives best performance when aggregating about 100 spots, generating MAPE with value of 7.21% for testing.

4) *Feed Forward Neural Network*: It is an alternative way to map nonlinear relation between  $y(t)$  and the vector  $\mathbf{x}_k(t-1)$  which contains  $y(t-k), \dots, y(t-1)$ . [7] We set 3 layers including 2 hidden layer, 12 input nodes (previous 4 occupancies, 3 PCA occupancy features, 3 categorical features, 2 numeric features)<sup>1</sup> as our FFNN setup. The 1st layer has 12 nodes, 2nd layer has 6 nodes. The output node represents forecasting occupancy. We leave 10% data out as testing set. The MAPE is performance towards the aggregation. Noticing that aggregation doesn't help a lot to reduce the error, since the MAPE itself is quite small, which is about 3% as aggregating above 100 parking spots.

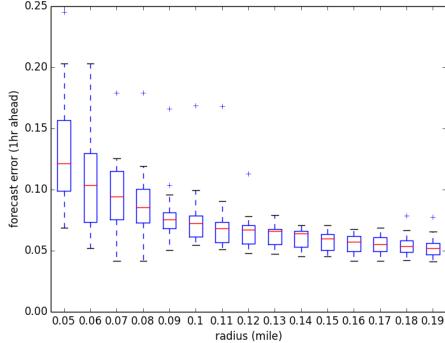
### D. Results

From following results in TABLE I, we find that the Neural Network gives us the best prediction among these models. However it needs longest time for training which is over 90 minutes. Whereas ARIMA, OLS, SVR only needs 39, 12, 20 minutes respectively.

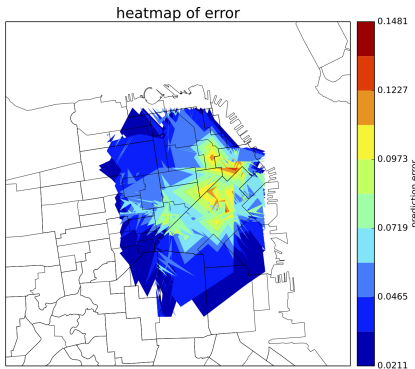
<sup>1</sup>categorical features are day of week, time of day, event; numeric features are distance and price

Models	Traning Error	Test Error
ARIMA(2, 0, 1) × (1, 1, 0) <sub>24</sub>	5.79%	8.09%
OLS	4.98%	7.88%
SVR	4.12%	7.21%
Neural Network (logistic)	0.98%	3.57%

TABLE I: Error Table



(a) ATT park region, 1 hour ahead, OLS model, error boxplot with mile expansion



(b) Heatmap of prediction error, 1 hour ahead, OLS

Fig. 4: MAPE plot of accumulated parking spots

By plotting out heat-map (Figure 4b) of prediction error, we can tell that Financial District and South Market Street Regions have relatively high error ratio. It actually follows the intuition because these areas are close to downtown area. And the parking in-out frequency is higher then other places. Therefore it cause more uncertainty to predict.

#### E. Error Reduction by Aggregation

Above analysis for a specific place (ATT park region) of aggregated lots shows a significant decrease of the prediction error compared to a single lot. It also has similar error reduction pattern based on distance expansion (Fig 4a). We try to explore it from theoretical point of view. Suppose there are two parking lots,  $u$  and  $v$ . The key element associate the geographical lots with the aggregated level of prediction error is the cross-correlation coefficients  $\gamma_{uv}$  of the difference

between prediction and observation, i.e  $\epsilon(t) = y(\hat{t}) - y(t)$ , for the single lot. We define

$$\gamma_{uv} = \frac{1}{T} \frac{1}{\sigma_u \sigma_v} \sum_{t=1}^T (\epsilon_u(t) - \bar{\epsilon}_u)(\epsilon_v(t) - \bar{\epsilon}_v) \quad (4)$$

, where  $T$  is the number of time points in timeseries,  $\epsilon_u, \epsilon_v$  are the respective timeseries of deviations between prediction and observation.  $\sigma_u, \sigma_v$  are the standard deviations of  $\epsilon_u, \epsilon_v$ .  $\bar{\epsilon}_u, \bar{\epsilon}_v$  represent mean of corresponding series across the time span.

The cross-correlation coefficient of  $\epsilon$  quantifies how similar two signals are. In our case, it measures the correlation of the prediction error between two separated lots. If  $\gamma_{uv}$  are given for all pairs of lots, the aggregation prediction error, say  $\epsilon_N$ , has the standard deviation  $\sigma_N$ . It can be expressed by

$$\sigma_N^2 = \frac{1}{N^2} \sum_u \sum_v \sigma_u \sigma_v \gamma_{uv} \quad (5)$$

where  $N$  is the number of aggregated lots. If all single lot prediction errors (e.g  $\epsilon_u, \epsilon_v$ ) are iid, assuming they have gaussian  $(\mu, \sigma)$ . Then  $\sigma_N^2 = \frac{\sigma^2}{N} < \sigma^2$ . Thus, the prediction error goes down by aggregation. Here we use a general but very strong assumption that each lot prediction error is independent identical distributed. But in reality, those error may or may not be correlated. We need spend more effort to investigate it.

#### IV. CLUSTERING

An aggregation of  $N$  parking lots is given by  $x_N = \frac{\sum_{n=1}^N s_n x_n}{\sum_{n=1}^N s_n}$ , where  $s_n, x_n$  are size of a lot and corresponding occupancy. Assume a forecaster that produces the corresponding forecast  $\hat{x}_N$ . The forecaster is a function of the previous data available at a particular time and other features. Denote by  $P_N \in R^{N \times T}$  the matrix of the profile patterns  $p_n$  for all  $N$  parking lots. The mean squared error (MSE) for the forecaster can be decomposed by conditioning on the (random) profile matrix using the tower property:

$$\mathbf{E}[\text{MSE}(x_N, \hat{x}_N)] = \mathbf{E}[\mathbf{E}[\text{MSE}(x_N, \hat{x}_N) | P_N]] \quad (6)$$

In order to find out if prediction errors are independent, we try to explore the daily parking profiles for different regions in this section. We divide up the San Francisco city into 7 regions, including ATT park, South Market Street, Financial District, Mission District, Fisherman Wharf, Lombard Street and Fillmore Street. A quick clustering by k-mean is shown as following Figure 5 We also aggregate nearby parking lots up to 120 spots to generate a daily shape. We move the geolocation centroid by grid search, and cover the whole ATT park region first to see if there is particular pattern in daily shapes.

- (a) **Dual Peak:** dual peak pattern is very obvious in AT&T park region. It hits first peak around 12pm and second peak around 8pm. Similar dual peak effect also can be found at cluster 1, 4 in Financial District.

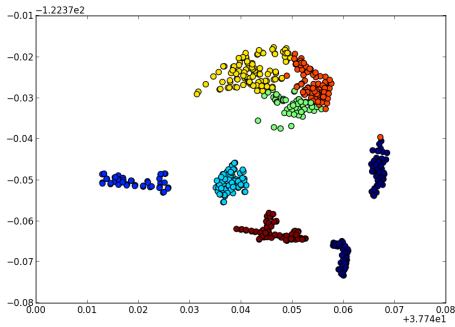


Fig. 5: 7 regions in San Francisco Parking

- (b) **Drop before Ramp up:** some shapes have pattern that occupancy decreased before increased to high level. Cluster 2, 4, 6 in Fig 6 has such effect because a lot of parking spots start to charge after 9am.
- (c) **Noon Peak:** Cluster 6 in Financial District have highest occupancy around 12pm

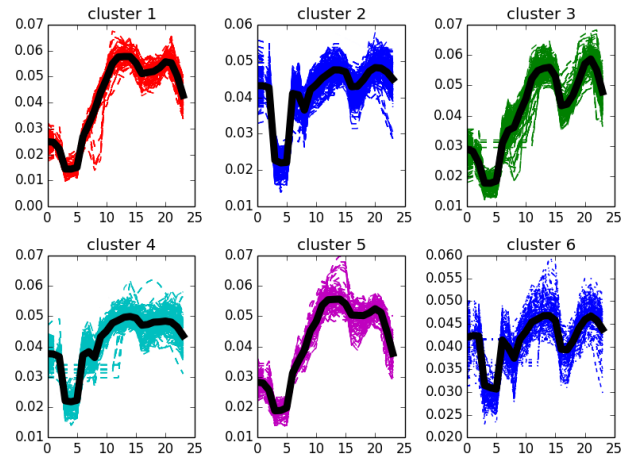
Different shape patterns could represent different group of travelers with their respective travel behavior. The dual peak shape shows some people leave after 4pm who are most likely to be regular commuters. At same time, other new drivers arrive in from 7pm till 10pm representing second peak of parking occupancy. The ramping up happens before 7am may result from drivers who are coming to work for business, which explains the real situation that most people start to heading work at that time. Noon peak appears in Financial District region which means travelers/tourists take major proportion of people who go to that place. In other words, for some days, a lot of tourists go to Financial District for visit.

Figure 6b is the histogram of cluster counts in AT&T park region. It shows all 6 clusters spread though half year more or less the same. While the Figure 7b shows that profile 6 has highest counts, suggesting that the corresponding shape appears more frequently during 6 months time in Financial District region.

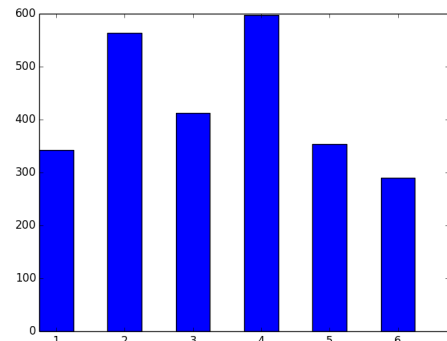
### V. DISCUSSION

We try to explore the relationship between aggregating parking lots and predicting parking occupancy. Obviously, prediction error is reduced by aggregating multiple parking lots. But the benefit is decreasing when the aggregation goes over certain threshold. We compare the trade-off between the aggregation and specific accuracy of place. And we take 100 - 120 spots as our prediction level. Usually, it equivalently means adding 6-8 parking lot together. We check the geolocation, and believe it doesn't deviate user too far, since most aggregated lots are within 0.5 mile radius.

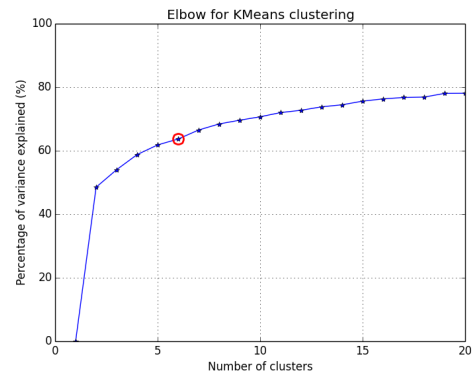
In feature selections, we also discover that not necessarily all previous 24 timesteps occupancy need to be taken in as a feature input for some model, such as linear regression. And the testing error doesn't increase too much with about 0.6% difference compared with SVR.



(a) ATT park region, 6 clusters



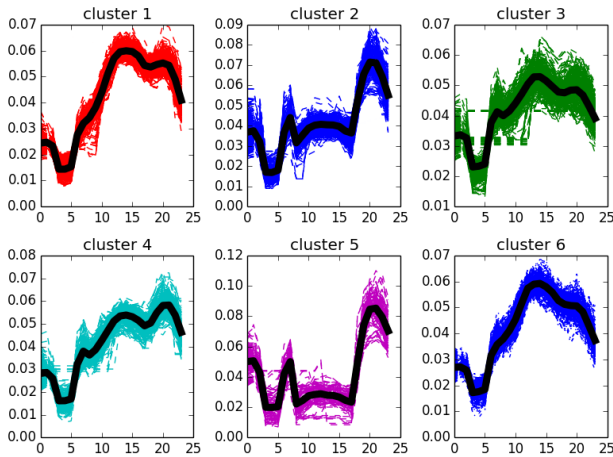
(b) histogram for 6 clusters



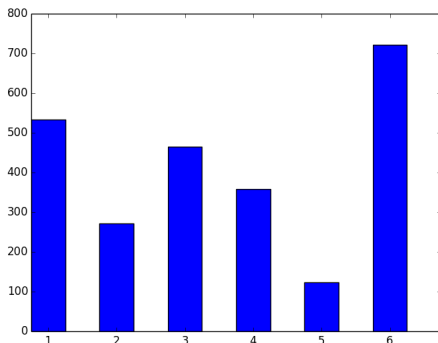
(c) variance explained percentage v.s. cluster number

Fig. 6: AT&T Park region

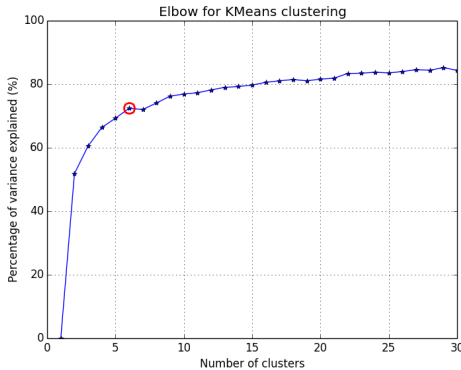
Comparing all prediction models we find neural network gives best result. But it also takes longest time to run. More interesting aggregation phenomena is observed from the daily shape pattern analysis, including *Dual Peak*, *Noon Peak*, and *Drop before Ramp up*. Profiling such groups will give huge benefits for predictions. It will help us to decompose the prediction error, and targeting different type of travelers.



(a) Financial District region, 6 clusters



(b) histogram for 6 clusters



(c) variance explained percentage v.s. cluster number

Fig. 7: Financial District region

## VI. CONCLUSION

In this project, we investigate the effect of aggregation on prediction for parking occupancy. We show that forecasting accuracy, as measured in relative error (MAPE) improve with larger mean occupancy. Several Models are implemented here for our prediction, and we find that Neural Networks performs best with longest training time.

We also give a sufficient condition leading to the observed

aggregation error decreasing, which is basically based on law of large number. Furthermore, to decompose the error we try to find if there is a fixed set of profiles of daily parking occupancy. From the region of AT&T park and Financial District, we find that the profile pattern indeed can be clustered into a few typical groups, such as dual peak group and noon peak group.

## VII. FUTURE WORK

The aggregation phenomena is also likely to be observed in other types of prediction processes, such as for example day ahead occupancy prediction, 15 minute ahead occupancy prediction. More models can be utilized to justify a detailed understanding of how aggregate occupancy patterns are formed and verified on higher resolution data. Also more features can be incorporated into our current prediction models, such as humidity or weather, population density in corresponding region, etc. In terms of clustering the occupancy profile, we might need to explore more regions to cluster the daily occupancy shape. And more profiles may be discovered from clustered daily shapes.

## REFERENCES

- [1] D. Shoup, "The high cost of free parking," *SFGate Article*, Jun 2005.
- [2] M. Richtel, "Now, to find a parking spot, drivers look on their phones," *NYTimes Article*, May 2011.
- [3] SFpark. <http://sfpark.org/>.
- [4] J. Lu, F. Valois, M. Dohler, and M.-Y. Wu, "Optimized data aggregation in wsns using adaptive arma," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*. IEEE, 2010, pp. 115–120.
- [5] M. Welling. Support vector regression.
- [6] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.
- [7] P. Auer, H. Burgsteiner, and W. Maass, "A learning rule for very simple universal approximators consisting of a single layer of perceptrons," *Neural Networks*, vol. 21, no. 5, pp. 786–795, 2008.