Extending estimation of parking occupancy to untracked city areas using city background information

15th May 2017

1 Abstract

Several smart cities around the world have begun monitoring parking areas in order to estimate free spots and help drivers that are looking for parking. The current results are indeed promising, however this approach is limited by the high cost of sensors that need to be installed throughout the city in order to achieve an accurate estimation rate. This work investigates the extension of estimating parking information from areas equipped with sensors to areas that are missing them. To this end, similarity values between city neighborhoods will be computed based on background data, e.g. from geographic information systems. Using the derived similarity values, the adaptation of occupancy rates from monitored- to unmonitored parking areas will be analysed.

2 Technical details

2.1 Data

There are some cities around the world that offer open parking occupancy information, either for a time interval in the past and/or currently. In Europe, the author found Dresden[1], Zurich[2] and Cologne[3] up to now that offer this kind of information for car parks. In the US such data is found for the cities of Philadelphia[4], Santa Monica[5] and San Francisco[6].

The SFpark project in San Francisco offers a large dataset for on-street parking. Occupancy data has been collected there for over two years between April 2011 - July 2013 in more than 400 blocks, resulting in over 1 million data records. The project was conducted with the main purpose of leveling off the parking occupancy in the city. In seven pilot areas of the city parking prices were adapted to the level of occupancy. Therefore drivers received incentives to park in areas less occupied by paying a smaller parking fee. Another two control areas were used to verify the effectiveness of the pricing measures. Parking data for all nine areas exist (see figure 1). In the following, SFpark will be taken as data reference.

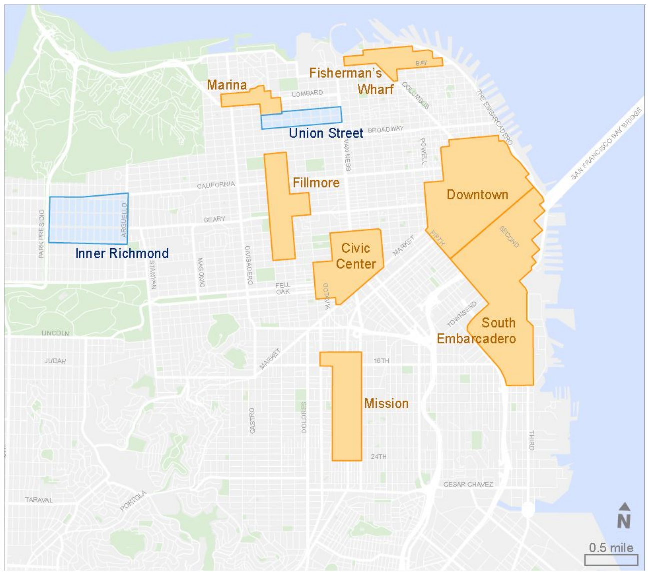


Figure 1: SFpark pilot and control areas

2.2 City background information

Metadata that indirectly indicates parking demand is highly interesting to analyse. Do areas with a high number of restaurants display a similar parking occupancy graph? Can parking behavior for quarters with a high sales revenue be correctly guessed?

Some of this metadata can be found in geographic information systems. In OpenStreetMap there can be found a large number of Points of Interest (POIs) that mark buildings of various types (offices, restaurants, cinemas, etc.) and some of their properties (working hours, opening times, capacity, etc.). Moreover, the physical coordinates corresponding to POIs are accessible as polygons, while streets are represented as lines. Hence further data can be inferred when merged with street metadata, e.g. size of parking lane in front of an office. This information will be used to create a "profile" of an area. For example, the POIs in northern part of San Francisco are displayed in figure 2.

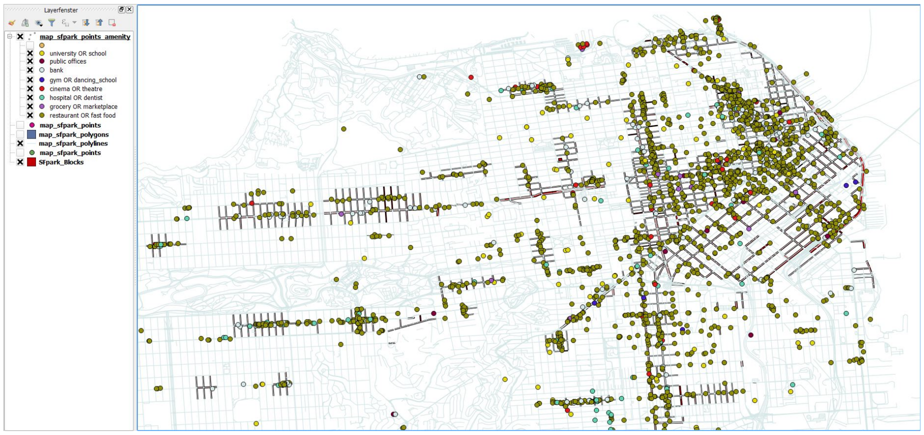


Figure 2: Screenshot from QGIS showing the dimensions of metadata available in OSM

Additionally, SFpark provides sales revenue information for its original nine areas, concerning food product and general retail. Even some of the information that the ML model is trained with but has little added value there can be used to consolidate an area profile.

2.3 Method

1. **Cluster all blocks into compact areas**

   Parking information is initially provided per block face. A block is the space between two consecutive streets in a typical American city (see figure 3). Since blocks are too small and unrepresentative a unit to process data for, the first step will be to geographically cluster blocks. A method as K-Means, K-Medoid or similar will therefore be used to group individual block faces into a convenient number of clusters. Only the geographic position and (Euclidean/Manhattan) distance are needed for clustering. Monitored and unmonitored areas will be clustered separately.

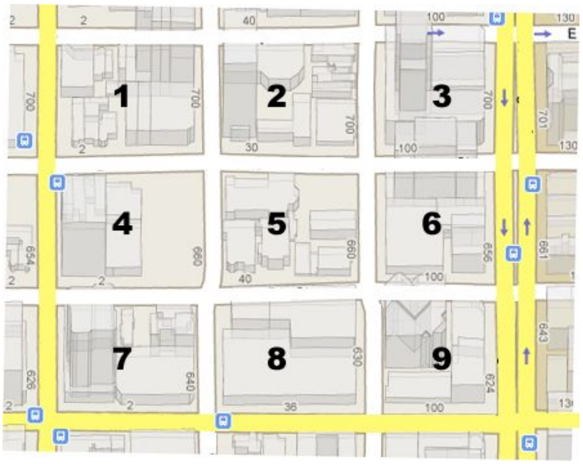


Figure 3: Blocks in a city, with each block displaying four faces for each surrounding street[7]

2. **Calculate pairwise similarity between unmonitored- and monitored clusters**

   Per cluster, vectors with the following dimensions will be built using GIS attributes:

   - count of office buildings × their capacity/size
   - count of cinemas/theaters/concert halls × their capacity/size
   - count of restaurants/fast food × their capacity/size
   - (other types of buildings that create parking demand)

   Further attributes that account for parking demand may be added to this list. The degree in which attributes increase or decrease the similarity of vectors will be evaluated.

   A similarity measure is introduced at this point. Candidates are Cosine Similarity, Jaccard Index, Kullbach-Leibler Divergence, possibly after reducing data dimensionality by running Principal Component Analysis or Latent Semantic Analysis.

   Taking Cosine Similarity as an example, between vectors A and

B with n dimensions, it is defined as:

$$\text{similarity}(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Its value lies generally between -1 and 1, but in our case it is limited to [0..1], since all dimensions have positive magnitudes. 0 corresponds to uncorrelated vectors, while 1 indicates that the vectors are the same.

3. **Build estimation models for monitored clusters**

For training ML models, the following SFpark attributes will be used:

- parking price - varies with zones and block faces
- traffic density - determined by sensors installed on the street surface, it indicates how much from a street section is occupied by cars
- fuel price - daily rate, valid for the whole city
- precipitation - the amount of rain or snow that fell per day, valid for the entire city
- street closing information - due to street parades, construction work, etc.
- parking demand up - signals the block faces that have a higher parking demand at a certain point, due to concerts or other extraordinary events
- construction site - signaled for a block face that it affects
- parking place count - the number of operable parking spots
- occupancy - the rate of occupied spots divided by the parking place count

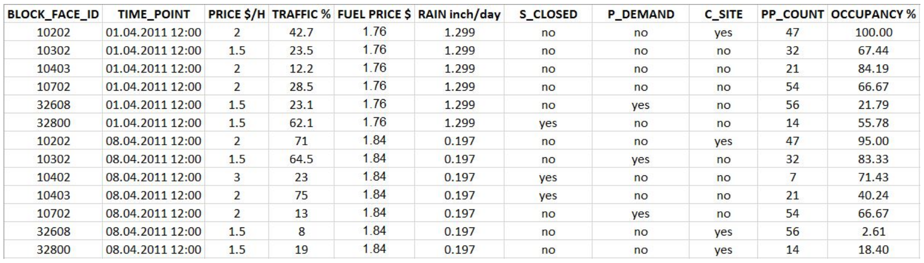
The target variable for the model is *occupancy*.

The estimation models will be computed using state-of-art methods, like neural networks, autoregression models and support vector regression[8][9].

In figure 4 there is a snippet of training data.

4. **Apply estimation models to unmonitored clusters, factoring in the respective similarity value**

For every unmonitored cluster C_u:



BLOCK_FACE_ID	TIME_POINT	PRICE $/H	TRAFFIC %	FUEL PRICE $	RAIN inch/day	S_CLOSED	P_DEMAND	C_SITE	PP_COUNT	OCCUPANCY %
10202	01.04.2011 12:00	2	42.7	1.76	1.299	no	no	yes	47	100.00
10302	01.04.2011 12:00	1.5	23.5	1.76	1.299	no	no	no	32	67.44
10403	01.04.2011 12:00	2	12.2	1.76	1.299	no	no	no	21	84.19
10702	01.04.2011 12:00	2	28.5	1.76	1.299	no	no	no	54	66.67
32608	01.04.2011 12:00	1.5	23.1	1.76	1.299	no	yes	no	56	21.79
32800	01.04.2011 12:00	1.5	62.1	1.76	1.299	yes	no	no	14	55.78
10202	08.04.2011 12:00	2	71	1.84	0.197	no	no	yes	47	95.00
10302	08.04.2011 12:00	1.5	64.5	1.84	0.197	no	yes	no	32	83.33
10402	08.04.2011 12:00	3	23	1.84	0.197	yes	no	no	7	71.43
10403	08.04.2011 12:00	2	75	1.84	0.197	yes	no	no	21	40.24
10702	08.04.2011 12:00	2	13	1.84	0.197	no	yes	no	54	66.67
32608	08.04.2011 12:00	1.5	8	1.84	0.197	no	no	yes	56	2.61
32800	08.04.2011 12:00	1.5	19	1.84	0.197	no	no	yes	14	18.40

Figure 4: Training data snippet indicating SFpark attributes

(a) the monitored cluster C_m will be selected that has the maximum similarity between the two clusters

$$S = max\{similarity(C_u, C_m)\}, \forall C_m \in \mathcal{C}$$

(b) C_m's model will be applied on the SFpark attributes of C_u, resulting in the prediction

$$P = M_{C_m}(attr)$$

(c) the final result will be a value interval

$$\text{occupancy}(C_u, \text{attr}) = [max\{0, P-(1-S)\cdot 100\}, min\{100, P+(1-S)\cdot 100\}]$$

(d) in case the resulting interval is too large, the previous steps can be repeated for the next most similar cluster C'_m. By calculating the respective interval and intersecting all of them, one may arrive at a more precise result. Alternative for interval intersection are Kernel Density Estimation and Kalman Filter.

5. **Use RDF to structure the data sources and model the estimation on a graph**

   Both the data readily available for estimation purposes and the open data resulting from mining can be organised in a Semantic Ontology, in the form of an RDF graph. Such a representation will not only help to logically structure the data and store it accordingly in the database, but it will enable the easy plug-in of information from other sources and from other cities, and adapt its format to the structure defined. In figure 5 an ontology that contains the simplified parking- and open data sources is depicted. Even though this ontology has been suited to San Francisco Data, in the course of investigating various cities, it will change accordingly so that different types of data can be adapted and plugged in.

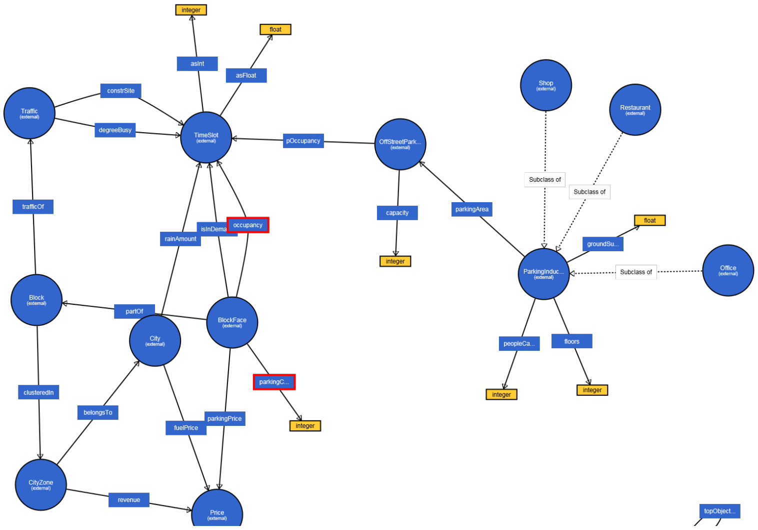
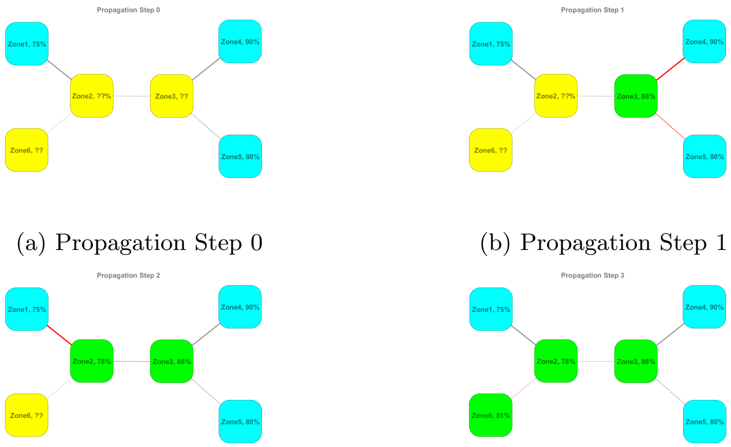


Figure 5: An Ontology of the parking- and open data built in Protégé and visualised using WebVOWL

Once such a representation is ready, it can be used (possibly in an aggregated form, see below) as part of a Machine Learning method. The idea behind it is the organisation of supervised and unsupervised elements as nodes in a graph that are linked by meaningful edges. In the context of estimating occupancy values, the transition from a supervised node (node for which the occupancy values are known) to an unsupervised node (node that is missing information on occupancy values) is done through a weighted edge (representing similarity or distance), by which an estimation for the unsupervised node can be inferred.

For example, in the following figure 6 a graph of city zones is depicted. For some zones there occupancy values at a certain point are known (supervised nodes, in light blue), for other nodes occupancy values are unknown (unsupervised nodes, in yellow). The edge weights are represented by their thickness (thin = similarity is low, thick = similarity is high). In a



(a) Propagation Step 0

(b) Propagation Step 1

(c) Propagation Step 2

(d) Propagation Step 3

Figure 6: Process of inferring occupancy values for untracked zones using occupancy values from tracked zones and their pair-wise similarity values as edges

propagation step the inference of a new node value is carried out, i.e. a value for the unsupervised node is calculated based on the edges from supervised nodes, which will appear in green.

This method is known as Semi-Supervised Machine Learning. Its application alongside other Prediction Methods like Linear Regression, Support Vector Machines and Neuronal Networks was researched in a smaller degree, therefore its result for this use case is interesting to find out.

2.4 Evaluation

In order to verify the hypothesis that parking occupancy in unmonitored areas is up to a degree similar to the occupancy in some monitored areas, concrete data is needed for the respective sensor-free locations. We will therefore split the sensor data available and take into consideration only areas for which occupancy data exists.

To test the accuracy of the inferred occupancy values, we estimate to need 10% of the available data. Training will therefore be performed using the remaining 90%.

3 Research Goals

The thesis will concentrate on elaborating answers on the following topics:

1. **Relevant features for Machine Learning models**
   Determine the pieces of data which correlate with the end-values and use them as features for model training. Which of these features are enough for the model to achieve peak estimation results?

2. **Selection of Machine Learning models and their result accuracy**
   Which ML models are best suited to yield best results?

3. **Relevant open data information that will be used for similarity**
   What kind of data complements the readily available parking information and is suited to form background parking profiles?

4. **Selecting a suitable similarity measure and representation of occupancy rates for untracked parking zones**
   What methods for calculating similarity lead to the best results?

5. **Accuracy of final occupancy rates**
   How much of the accuracy of the estimation rates for tracked city zones translates to the untracked city zones?

6. **Take into account the variability of data from multiple city use cases in the data organisation and all the above points**
   Apart from SFpark, the process of deriving occupancy rates for untracked zones will be carried out for other cities, e.g. Dresden[1], Zurich[2] and Cologne[3], Philadelphia[4], Santa Monica[5]. Thus, the resulting approach will be standardized to fit future use cases.

Up to now, the smart parking literature concentrates on capturing data and predicting parking information using static or mobile sensors, by covering the areas for which future rates are calculated[10]. An extrapolation approach as described here, to the best of my knowledge, has not been explored yet.

4 Work Plan

Time period	**Activity**
1st - 2nd Week	Literature Research; Identify relevant sources
3rd - 4th Week	Work out concept; Settle on Features/Type of Data used
5th - 14th Week	Technical realisation with focus on extensibility
15th - 20th Week	Evaluation: Represent findings through visualisation
21st - 22nd Week	Collect Feedback and formulate improvements
23rd - 25th Week	Finish Documentation

References

[1] *Freie Parkplätze Dresden.* http://codefor.de/projekte/2014-04-19-dd-freieparkplaetze.html. Accessed: 2017-03-16.

[2] *Parkleitsystem Zürich.* https://data.stadt-zuerich.ch/dataset/parkleitsystem. Accessed: 2017-03-16.

[3] *Parkhausbelegung Köln.* https://www.offenedaten-koeln.de/dataset/parkhausbelegung. Accessed: 2017-03-16.

[4] *City of Philadelphia - Airport Parking Availability.* https://www.opendataphilly.org/dataset/airport-parking-availability. Accessed: 2017-03-16.

[5] *Santa Monica Open Data - Parking Lot Counts.* https://data.smgov.net/Transportation/Parking-Lot-Counts/ng8m-khuz. Accessed: 2017-03-16.

[6] *SFpark project.* http://sfpark.org. Accessed: 2017-03-05.

[7] *Japanese addresses: No street names. Block numbers.* http://sivers.org/jadr. Accessed: 2017-02-19.

[8] Xiao Chen. *Parking occupancy prediction and pattern analysis.* Tech. rep. Technical report, Stanford University, 2014. Machine Learning Final Projects.

[9] Tooraj Rajabioun and Petros A Ioannou. “On-street and off-street parking availability prediction using multivariate spatiotemporal models”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.5 (2015), pp. 2913–2924.

[10] Trista Shuenying Lin. "Smart parking: Network, infrastructure and urban service". PhD thesis. Lyon, INSA, 2015.

[11] *Google Blog Entry on Semi-Supervised Machine Leaning.* `https://research.googleblog.com/2016/10/graph-powered-machine-learning-at-google.html`. Accessed: 2017-05-14.

[12] Tim Tiedemann et al. "Concept of a Data Thread Based Parking Space Occupancy Prediction in a Berlin Pilot Region". In: *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence.* 2015.

[13] Reinhard Hössinger et al. "Development of a Real-Time Model of the Occupancy of Short-Term Parking Zones". In: *International Journal of Intelligent Transportation Systems Research* 12.2 (2014), pp. 37–47. ISSN: 1868-8659. DOI: `10.1007/s13177-013-0069-5`. URL: `http://dx.doi.org/10.1007/s13177-013-0069-5`.

[14] Andreas Klappenecker, Hyunyoung Lee and Jennifer L Welch. "Finding available parking spaces made easy". In: *Ad Hoc Networks* 12 (2014), pp. 243–249.

[15] Eric Hsiao-Kuang Wu et al. "Agile urban parking recommendation service for intelligent vehicular guiding system". In: *IEEE Intelligent Transportation Systems Magazine* 6.1 (2014), pp. 35–49.