Predicting Free Parking Spots using Streaming-Data in the city of Aachen

Andrei Ionita

November 9, 2016

1 Introduction

Looking for a parking spot has increasingly become a difficult affair for drivers. Modern navigation systems are missing a component that informs them on available parking. Not only would drivers benefit from this piece of information when delivered in real time, but receiving it in advance would ensure a better planned trip, without unnecessary time being lost or by causing additional traffic.

This thesis will develop a machine learning framework based on live and historical data provided by sonah UG, while considering various contextual attributes in order to predict free parking spots in the city of Aachen, Germany. The forecast will focus especially on determining whether a parking location has at least a certain number of parking spots available at a future time point, which translates into a high likelihood for a driver to find a free parking spot on arrival.

sonah UG (haftungsbeschränkt) develops software for automated parking space detection in urban areas. This Software can be embedded into optical sensor technology and existing CCTV infrastructure. In addition, sonah UG is working with cutting-edge statistical tools to deliver parking space prediction to everyday users.

1.1 Conventions

The following terms will be used throughout this document:

- **parking spot** - a single parking place
- **parking location** - a space consisting of multiple parking spots

2 Data Acquisition

Data plays a pivotal role in an Information System. In our case, it can broadly be split into two parts: direct parking- and contextual data. The former consist

generally of parking snapshots, i.e. parking situations at a certain moment in time, e.g. 71 parking spots are now free, parking spot #42 got occupied at 17:00. Contextual data refer to the geographical infrastructure where parking locations are represented, i.e. type of parking (free, for customers, for employees), "forbidden parking" spots, relevant businesses in the area that lead to parking occupancy, etc.

2.1 Parking-Data Snapshots

1. **Sensor Data** — the optical sensors deliver data records consisting of status change per parking spot; the format is: location (latitude, longitude), timestamp and parking spot status

2. **Parking Meter Data** — consists of the number of tickets that have been purchased in a time period, e.g. day; this number alone does not offer direct information about the total number of parked vehicles during the certain time period, i.e. cars with permanent permission are not included, neither ticket dodgers

3. **Car Park Data** - the current number of free parking spaces in the city car parks as they are made available online

2.2 Parking Geographical Information

1. **Open Street Map** — OSM is an open Geographical Information System (GIS) platform where content is contributed by users. OSM is able to store detailed parking information, among others, consisting of location, type of parking (public, customers, company, etc.), parking fee, parking capacity, and others attributes.

2. **User-Generated Data** — a smartphone application that asks users to introduce pieces of information regarding parking areas at their convenience, e.g. disabled parking spots, number of current free parking spots, type of business in the building nearby etc. Some of the information asked for is intentionally redundant, in order to cross-validate data from other sources. The app users will hence have a stake at improving the parking-related infrastructure.

2.3 Other contextual information

1. **Events** — whether concerts, sporting events, Christmas markets or others, events can greatly impact parking occupancy for certain periods of time. Therefore it makes sense to gain access to the time and place of events.

2. **Weather Conditions** — dry, rainy or snowy weather plays a large role when deciding to use a car in the city and hence impacts parking as well.

3. **Traffic Flow / Construction Sites** — investigating the correlation between high/low traffic and nearby parking locations may help in improving parking prediction. It is therefore important to gain access to traffic flow data and temporary construction site locations.

2.4 Access to Data and Persistence

As soon as we discover a useful data source, gaining access to it is essential. Internal data will be saved in our **database**s. In case of Parking Snapshots, a NoSQL database will be used, since snapshots are committed very frequently and non-relationality offers the advantage of easier processing, i.e. by using map/reduce, etc. Static data like geographical infrastructure shall conveniently be stored inside a relational database, since its size and rate of change pose no issues for processing.

A challenge here is modeling the parking areas. Since there are no clearly defined limits for a parking areas, i.e. neither are parking meters used by the drivers necessarily the closest to their parking position, nor is this distance trivial to define, we will organise parking areas according to the business or institutions in that area. Hence all parking spots belonging to a restaurant are a unit, the same for school parking spots, office parking spots, and so on. These parking layers may overlap (see Figure 1).

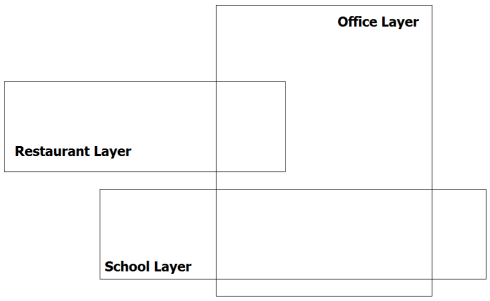

Figure 1: Parking Locations organised in layers according to business and institutions on site

External data is at best accessed via an **Application Programming Interface (API)**, preferably REST, since this paradigm simplifies HTTP request/response processing; nevertheless, we accept querying other kinds of APIs too.

When external information is not explicitly made available, we investigate the approach of **crawling** or scraping the providers' web-sites. Scraping is the process in which regular expressions are matched against retrieved HTML content in order to "clip out" interesting information. Before any external party content is retrieved however, we will make sure to comply with the law in this respect and will check whether the certain web-sites are crawlable at all. Upon

Data type	Object reported	Timestamp	Retrieval cycle	Access
Sensors	parking spot	live	immediately	database
Parking Meters	parking ticket	historical	periodically	API
Car Parks	vehicle	live	periodically	crawling
OSM	GIS element	—	periodically	API
User-generated	anything	—	immediately	database
Events	event element	near future	periodically	crawling
Weather	weather element	near future	periodically	crawling/API
Traffic flows	traffic report	live	periodically	crawling/API

Table 1: Data overview and its features

accessing external data, it shall be persisted in our databases, so as to avoid repeated retrieval and not to rely on external servers.

2.5 Data Coverage Limitations

In spite of all the previously described different data sources at our disposal, there will be significant numbers of parking locations that will not be covered. Therefore, we investigate extrapolating the already available data to the unsupervised parking locations, i.e. extend and adapt values from one location to another by taking into account contextual information (see 2.2 and 2.3). More on this shall be discussed in the Model Selection section of the thesis.

To summarize all data sources, Table 1 presents an overview of the comprised data and its characteristics.

3 Modeling parking locations

3.1 Feature selection

In order to better describe the model, we first need to determine its relevant features, i.e. parameters that correlate with the parking availability outcome. Common features include time of day, day of week and events[1][3][5]. Weather conditions may also play a role, which we shall investigate. Further, we shall look into the presence relevant buildings or geo-features in the neighborhood, i.e. offices, schools, residential areas, etc. that provide parking areas and account for certain parking behaviors.

3.2 Measuring accuracy

We need some metrics in order to evaluate the performances of the following models. Given a time series $y(t), t = 1, .., n$ representing observed values, and its predicted series $\hat{y}(t)$, the Mean Absolute Percentage Error is defined as [3][12]

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

3.3 Model selection

The pool of applicable models may broadly be divided into discrete and continuous. The former consists mainly of classifiers, i.e. predicted values are a set of classes, e.g. less than 5 spots, 5 - 20 spots, more than 20 spots free. Discrete models fit themselves therefore to less precise data and require only approximate information for training. Continuous models, on the other hand, aim at calculating exact numbers in their predictions. The more precise data they are trained with, the higher the accuracy they deliver on test instances.

As the literature on Parking Occupancy mainly focuses on the continuous variety, we shall merely mention the possibly applicable multiclass classifiers: Decision Trees, k-Nearest Neighbors, Naive Bayes, Neural Network (discrete version), Support Vector Machines. The thesis will treat them in more detail.

Further we shall concentrate on the methods that have been applied in the relevant literature for problems similarly defined as ours. Each section consists of a short definition of the method and points out its hallmarks and the reasons for which it applies in our setting. The thesis may go into some further degree of relevant mathematical detail.

3.3.1 Linear Regression (OLS)

Linear Regression models suppose that the predicted time series is a linear combination of defined attributes. Considering $y(t)$ as a time series, $f_i(t)$ the value of the feature i at time t, its estimation is given by:

$$y(t) = w_0 + \sum_{i=1}^{|features|} w_i f_i(t) + \epsilon(t)$$

where w_i are weights determined during training and $\epsilon(t)$ is the least square error that results from optimally fitting the line to the points in the graph (see Figure 2).

Chen[3] tries out a linear regression model for 120 parking spots and 1 hour ahead prediction. His R^2 value is 92% and MAPE is close to 8%.

3.3.2 Support Vector Regression (SVR)

Support Vector Regression is a prediction method applicable also for non-linear models. Compared to linear regression, it distinguishes itself by fitting future points using a non-linear combination of support vectors, i.e. selected points from the current set. During the training phase, the support vectors are selected as the points that fall outside the ϵ error value range (see Figure 3). The error is measured in absolute value and only accounts for points outside the 2ϵ range.

$$x = b + \sum_{i\ support\ vector} \alpha_i (a(i) \cdot a)$$

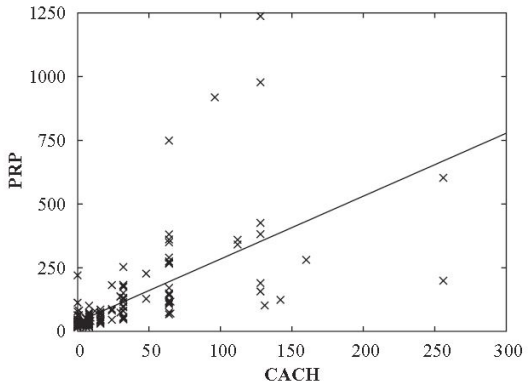

Figure 2: Example of a line that fits a set of points according to linear regression; plot taken from [18]

where b and α_i are determined in the training phase, $a(i)$ are the support vectors and a is the current instance. The dot product term is called the kernel function and is responsible for non-linear behavior when it contains higher order factors. There are various kernel functions used and choosing a good one for one's case is not trivial. As a whole, the algorithm tends to both minimize the error and simultaneously to maximize the flatness of the regression function, thus avoiding overfitting, i.e. biasing the model towards the training data.

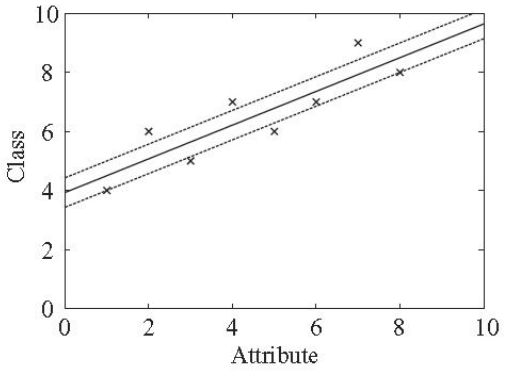

Figure 3: Example of a line determined by SVR; some points lie outside the 2ϵ-wide band; plot taken from [18]

Chen[3] constructs vectors of 24 elements that represent parking values corresponding to the last 24 hours and uses Support Vector Regression to predict future values:

$$y(t+1) = w^T \gamma(x_k(t)) + b$$

where $x_k(t)$ holds the parking data for $y(t-k+1), y(t-k+2), ..., y(t)$. The application of this model yields a MAPE value of about 7% for about 100 parking spots.

3.3.3 ARIMA

In time series analysis autoregressive integrated moving average (ARIMA), models are used to forecast future points in the series. The autoregressive (AR) part accounts for its own lagged (i.e. previous) values, while its moving average (MA) part computes future values using past prediction errors. An ARIMA model predicts future values using its both components. Its prerequisite is that the data has to be stationary, i.e. its mean and variance are constant over time. In order to achieve that, the time series at hand may be differentiated (i.e. replace current point with the difference to its previous point) more times.

$ARIMA(p, d, q)$ stands for a model that has been differentiated d times, looks behind at its last p values and at its previous q prediction errors. An $ARIMA(p, d, q)(P, D, Q)_m$ is a seasonal model that recognizes some temporary behavior based on reoccurring events of period m.

Chen[3] uses an $ARIMA(2, 0, 1) \times (1, 1, 0)_{24}$ model with the seasonal factor corresponding to the hours of the day. He measures a MAPE error ratio for 100+ parking spots for 1 hour ahead of about 6%.

Rajabioun and Ioannou[12] propose a multivariate autoregressive model that takes into account not only temporal but also spatial correlations of parking availability. They observe that the nearer the parking spots are, the more similar the occupancy rates are. Their model yields a 14% MAPE for a 20-minute prediction horizon.

3.3.4 Neural Networks

Neural Networks (NN) relax the conditions of linearity and stationarity that the previous models impose and hence may achieve new levels of accuracy. They consists of a directed graph that contains nodes (i.e. neurons) with a activation function, which map inputs to output values. In particular, Multilayer Perceptrons (MLP) are models where the activation function is non-linear. Apart from the input and output layers, MLPs have one or more hidden layers. Learning occurs by changing the weights in the activation functions by observing the error between the output and expected result. This supervised learning method is called backpropagation.

Vlahogianni et al.[17] use a MLP with 8 hidden layers for a 4% MAPE 1 hour prediction error. Chen[3] achieves a 3% error when using 2-hidden layer NN for an 1 hour-ahead prediction.

3.3.5 Continuous Markov Chains

Markov Chains are used to model processes in which future states depend causally only on the current state and not on previous states. Between any two states there may be a transition probability which expresses how likely the process is to jump from one state to the other. Markov Chains can be either discrete- or continuous-time. For the latter, the time spent in each state has an exponential distribution, i.e. states transitions at fixed time intervals.

This model's prerequisites indeed seem to apply to the parking space occupation problem: future parking configurations depend solely on the current configuration. Caliskan et al.[2] defines a state of the process by the absolute occupancy value. The authors consider the average parking arrival rate and the average parking rate (i.e. inverse of parking duration) to construct a Continuous-Time Markov Chain (CTMC) and to predict future occupancy values. In Figure 4, the CTMC is represented by λ (parking arrival rate) and μ (parking rate). The authors do not provide an exact error measure for their application of the model, however state that "our prediction algorithm is most effective for prediction times up to 15 minutes [..] with increasing prediction time, the uncertainty of the model increases".

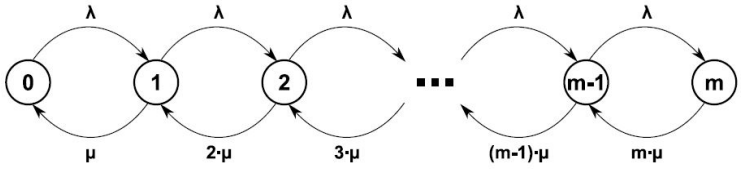

Figure 4: Continuous-Time Markov Chain corresponding to a Parking Location; taken from [2]

To summarize the presented methods, we outline their main features in Table 2. An exact error-comparison based on the same testing ground is, at this point, not available. Individual testing results have been provided in corresponding methods' paragraphs.

4 Implementation

The implementation will be done in collaboration with sonah UG, in the following way. The data (for training and testing) will mostly be available through sonah UG, as shown in the Table 3. The company members will furthermore be available for consultations, however their support will be limited to suggestions, whereas the last word will be with the supervising chair. The software component will be realised in Java or Python. It remains to be decided whether it will take the form of a library or will integrate with the User-Interface in an MVC

Method	Recommending features	Limitations
Linear Regression	general and simple prediction method	only matches linear models
SVR	additionally fits non-linear models	difficulty choosing kernel function
ARIMA	applies directly to time-series	requires linearity and stationarity
Neural Networks	adaptable to latest observed results via backpropagation	proneness to overfitting; its "black box" nature
Markov Chains	models processes of independent events	does not accommodate other features than time-related ones

Table 2: Prediction Method Summary

application (e.g. in Django). Databases used will be a NoSQL (e.g. MongoDB) and a relational database (e.g. MySQL).

Data type	Current Availability	Source
Sensors	available	sonah UG
Parking Meters	partially available	Aachen City via sonah UG
Car Parks	partially available	Aachener Parkhaus GmbH
OSM	available	open and free
User-generated	not yet available	sonah UG
Events	not yet available	unknown for now
Weather	partially available	Deutscher Wetterdienst via sonah UG
Traffic flows	not yet available	unknown for now

Table 3: Data source and availability for implementation

4.1 Evaluation

The testing bed is the city of Aachen, Germany. sonah UG provides data for the Frankenberger quarter, hence the application will be tested on the parking activity in this part of the city. A fraction of the accumulated live parking data will be used towards training the model. We estimate to use 10-30% of the data for testing, depending on how consolidated the model already is.

Alternatively, test data can be simulated, in case not enough parking data was collected by that time.

5 Research Goals

In order to arrive at an efficient parking prediction application, there are several unknowns that need to be investigated. The thesis will elaborate answers to the following questions.

How are the data stored and retrieved?
As outlined in 2.4, data is either stored internally or accessed from external sources. The former is split among dynamic and static data. Dynamic data are mainly represented by sensor data records and are expected to be uploaded in frequent intervals. NoSQL databases are suited for this particular case, as inbuilt retrieval and aggregation operations are available, e.g. map/reduce. How efficient will this turn out in practice, will be reflected in the application performance.

How are parking- and contextual data associated?
Both parking- and contextual information (GIS, traffic, events, etc.) have a geographical position in common, around which data can be linked. How efficient will the association be saved and retrieved, the thesis will provide answers.

What features are indeed relevant in order to build a prediction model with?
A feature (e.g. business surroundings, weather, traffic) can be considered relevant to a prediction model if it has a mathematical influence on the parking occupancy values, i.e. if a correlation with the end-values can be determined. Whether all the considered data in this proposal is indeed relevant, or other features will be found to be relevant (sections 2.1, 2.2 and 2.3), it remains to be seen.

In what degree can "parking profiles" be extrapolated to fit unsupervised parking locations?
The degree in which extrapolation of parking information and contextual elements is possible will depend on the results of the prediction models. The data first will be tested on areas for which there are data sources, so that the results can be cross-validated. Whether the validation will be positive, we expect to find out.

How accurate can a system predict availability of free parking spots?
There have been related investigations that yield results as good as %3 MAPE for 1-hour in advance prediction(see 3.3.4). The results of our work will be measured on the same scale and will be compared to existing research. It is still to be seen whether our results will improve on those with a similar setup.

What additional conditions should a parking recommending system take into account?
Apart from rate of free parking spots in parking locations, another relevant condition may be minimizing the distance towards the final destination of drivers, i.e. home, work, restaurant. How a trade-off between multiple optimization

functions can be achieved and possibly other conditions, will be further investigated.

6 Timetable

Time period	Activity
1 - 2 Week	Literature Research
3. - 4. Week	Consolidating Concept
5. - 14. Week	Implementation
15. - 20. Week	Evaluation
21. - 25. Week	Documentation

References

[1] Felix Caicedo, Carola Blazquez, and Pablo Miranda. "Prediction of Parking Space Availability in Real Time". In: *Expert Syst. Appl.* 39.8 (June 2012), pp. 7281–7290. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2012.01.091. URL: http://dx.doi.org/10.1016/j.eswa.2012.01.091.

[2] Murat Caliskan et al. "Predicting parking lot occupancy in vehicular ad hoc networks". In: *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*. IEEE. 2007, pp. 277–281.

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

[4] Jatuporn Chinrungrueng, Udomporn Sunantachaikul, and Satien Triamlumlerd. "Smart parking: An application of optical wireless sensor network". In: *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*. IEEE. 2007, pp. 66–66.

[5] 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.

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

[7] Rongxing Lu et al. "SPARK: a new VANET-based smart parking scheme for large parking lots". In: *INFOCOM 2009, IEEE*. IEEE. 2009, pp. 1413–1421.

[8] Suhas Mathur et al. "Parknet: drive-by sensing of road-side parking statistics". In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM. 2010, pp. 123–136.

[9] Anandatirtha Nandugudi et al. "PocketParker: pocketsourcing parking lot availability". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2014, pp. 963–973.

[10] Ramu Panayappan et al. "VANET-based approach for parking space availability". In: *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*. ACM. 2007, pp. 75–76.

[11] Tooraj Rajabioun, Brandon Foster, and Petros Ioannou. "Intelligent parking assist". In: *Control & Automation (MED), 2013 21st Mediterranean Conference on*. IEEE. 2013, pp. 1156–1161.

[12] 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.

[13] Piotr Szczurek et al. "Learning the relevance of parking information in VANETs". In: *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking*. ACM. 2010, pp. 81–82.

[14] Piotr Szczurek et al. "Learning the relevance of parking information in VANETs". In: *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking*. ACM. 2010, pp. 81–82.

[15] Dušan Teodorović and Panta Lučić. "Intelligent parking systems". In: *European Journal of Operational Research* 175.3 (2006), pp. 1666–1681.

[16] 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.

[17] Eleni I Vlahogianni et al. "Exploiting new sensor technologies for real-time parking prediction in urban areas". In: *Transportation Research Board 93rd Annual Meeting Compendium of Papers*. 2014, pp. 14–1673.

[18] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123748569, 9780123748560.

[19] 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.