POLITECNICO DI TORINO

Master Degree Course in Computer Engineering

Master Degree Thesis

Characterization and Prediction of Car Sharing usage exploiting Points of Interest information

Supervisors
prof. Paolo Garza
prof. Luca Cagliero
prof. Silvia Anna Chiusano

Candidates
Alexander Sebastian Abstreiter
matricola 250157

Academic year 2018/2019

Abstract

Recently, free-floating car-sharing systems were introduced as a novel way of mobility allowing users to reserve a car shortly before the rental with their smartphone. This approach is more dynamic than previous station-based car sharing systems, since users can start their trip wherever a car is available and terminate it anywhere in the operator's area. Hence, this new form of car sharing is not only appealing to users with the need for a car but also to those who want to speed up their travel.

One drawback of this system is that the user does not know in advance if there will be a car available nearby. This work solves this problem by predicting future car availabilities around Points of Interest (POIs) in the city. We train Random Forest models on features from the car sharing and POI datasets in order to generate predictions for the car availability in the near future.

By applying this method to car sharing systems in Portland, Seattle and Turin, we are able to outperform the baseline of predicting the last recorded value. Therefore, our predictions can be used by users to inform themselves about future car availability or by car sharing operators to relocate their fleet.

In addition, we discover behavioral patterns of users by applying sequence pattern mining to the car sharing data. Results from Portland and Turin show that we can extract different sequences with respect to both, specific POIs and POI categories. Moreover, this work shows that the extracted sequences coincide with sequences from a check-in database, where the users explicitly specify that they are visiting a POI at a given timestamp. Thus, the discovered sequences of car sharing can be used to study the general movement of citizens.

Furthermore, a temporal and a spatial contextualization was conducted. Dividing the time of the day into four timeslots, differences in the frequency and confidence of the discovered sequences are found in each of the timeslots. Similarly, we split the area of the car sharing operator into multiple smaller areas and study the sequences extracted for each area. The results suggest that car sharing usage patterns highly depend on the area, as well as the time of the day.

The discovered sequences can be used to study origins and destinations of car sharing trips, as well as the general movement of citizens. For instance, this can help the car sharing operator to decide which areas provide a good market to expand to and to discover general mobility patterns.

Contents

Chapter 1

Introduction

1.1 Station-based Car Sharing

A Car Sharing System (CSS) is a car rental service conceived for drivers who use cars only occasionally or do not own a car. The traditional car sharing model is based on stations where users can start and end their trip. The older model offers only round-trips, meaning that the client has to return the car at the same station where it was picked up originally. Another version, the so-called one-way car sharing model is more convenient for the user because it allows returning the car at any station owned by the provider [37]. Since an advance reservation is necessary for many of these services, station-based car sharing is less spontaneous but rather a planned way of mobility.

However, Heafeli et. al showed that car sharing has a positive impact on the environment by reducing the CO_2-emissions [15] and another study by Shaheen et. al found that it is reducing the number of kilometers traveled in a car [31].

Despite its long and steady growth, station-based car sharing has not been adopted by the mass market but only by a very specific customer group, e.g. by young, highly educated and car-free households [5, 15].

1.2 Free-Floating Car Sharing

In the last few years, a new car sharing model has been introduced: the so-called free-floating car sharing. Free-floating car sharing systems have no depot stations at all instead, the vehicles can be taken and left anywhere in the operative area. In addition, reservation can be done short time prior using a smartphone, hence advance reservation is not required making the overall system more spontaneous and flexible than the station-based ones. Furthermore, the business models do not rely on fixed prices and users can access GPS-based real-time availability information of cars. The first free-floating vehicle-sharing system was settled up in Ulm, Germany, in 2009. The success of this new sharing model is such that in 2011, in Germany, this market has already been approximately 25 times higher than the average market of traditional car sharing providers [11].

1.3 Forecasting data in the context of car sharing

Forecasting demand, supply and future trips of cars is an important aspect for car sharing providers in order to optimize their product and business processes. Such forecasts can be used to evaluate potential operating areas, to adjust the number of vehicles and to relocate the fleet on short notice to areas of high demand [38, 40, 25, 7]. These approaches are described in more detail in the following chapter.

At first, in part I of this work, a prediction of car availability in the future is conducted. Chapter 2 motivates this task giving an overview of the related work and explains the background for the prediction models. After that, we describe the related experiment in chapter 3 and discuss the results in chapter 4.

In part II, we explore behavioral patterns of car sharing users as sequences of points of interest. Motivation and background on sequence pattern mining is given in chapter 5, followed by the description of the experiment in chapter 6. The discovered sequences are discussed in chapter 7.

Finally, we draw a conclusion and propose future work on this topic in chapter 8.

Part I

Prediction of future car availability around Points of Interest

Chapter 2

Motivation and Background

Considering the use case of a customer desiring to know whether there will be a car near him or her available in two hours, we formulate the task as a classification problem deciding if at the specific timestamp in the future the number of cars around a Point of Interest (POI) is greater than zero or equal to zero. For this prediction we use a Random Forest classifier, which is explained in section 2.3.

2.1 Related work

Numerous scientific work was conducted on station-based round-trip CSSs focusing on finding variables which have an impact on the demand of such a CSS. In [8], Celsor and Millard-Ball conducted a Geographic Information Systems based analysis characterizing market segments in urban areas in the United States of America. Instead of individual users' demographics, they found that the neighborhood and transportation characteristics are important indicators for the success of a CSS. For instance, members tend to be highly educated but not the whole population in their neighborhood shares this attribute. Thus they suggest that using variables such as commute mode split, household composition, and vehicle ownership in a neighborhood are better indicators for the adoption of CSS.

Another study by Stillwater et. al [34] confirms the previous one and suggests that also the built environment, such as the street width, impact the usage of CSSs. In Seoul however, Kang et al. showed in [18] that the adoption of a CSS is closely linked to high vehicle ownership and less rail accessibility.

However, this knowledge might not be transferable to free-floating car sharing systems, which was shown to have different customer groups and user behavior due to its more flexible approach [20]. Becker et al. confirm this findings in [4] by suggesting that free-floating CSS are often used in order to save time compared to other transportation options, while station-based vehicles are rented in the case of an actual need for a car.

Schmoeller and Bogenberger analyzed external factors on the spatial and temporal demand of Car Sharing systems in [30]. Using car sharing data from Munich in early 2012 they found a high impact of the time of day, the weekday and the spatial factors on the demand. The weather was found to have only low impact on the car sharing demand.

In [39], Weikl and Bogenberger clustered the car sharing demand of one year into six clusters using K-means clustering. They found that for different timeslots of three hours and one of six hours over the night the clusters differ highly, which underlines the importance of the time of the day for car sharing usage.

Wagner et al. investigated the impact of the time and the zone on the idle time of the vehicles of a Car Sharing system in Vancouver [38]. They identified several hotspots in the city and proposed a user-based relocation model, which incentives the customer to do a trip from an area with less demand to one with higher demand.

In another work, Wagner et al. analyzed Car Sharing data with the corresponding neighborhood data and Point Of Interest (POI) data [36]. Using a zero-inflated regression model which incorporates POI density and population density, education, unemployment and foreigner rate and income per person, they were able to predict the demand for different areas well. The most positive influence on the demand was found to being airports, ATMs, bus and train stations, a high population density, a low-income neighborhood, and a high share of foreigners in the neighborhood.

Extending the previous approach Willing et al. added the time of the day and the weekday to the model in [40]. They found that different POI categories have a different influence on trip destinations depending on the time of day and the day of the week, e.g. restaurants have a negative influence in the 12:00-16:00 slot but a strongly positive one in the subsequent two timeslots 16:00-20:00 and 20:00-24:00.

In [24], the authors performed a spectral analysis to cluster the zip code areas of the city of Berlin into four clusters according to their frequency in the number of bookings. For each of these clusters, they applied an ARIMA model and Holt-Winters Filtering, out of which the latter one produced better results when forecasting the future number of bookings.

Random Forests, first proposed by Leo Breiman in [6], have been successfully applied to multiple domains, including online learning and tracking, Cancer classification, astronomical object classification, Traffic signs classification and classification of crops [14] as well as multiple organ segmentation and detection of Parkinson-related lesions [27]. Since Random Forests are an ensemble of Decision Trees, we start by explaining the latter one.

2.2 Introduction to Decision Trees

A decision tree for classification consists of several splits, which determine the predicted class for an input sample, which is a leaf node in the tree [9]. The construction of the decision trees is done with a greedy algorithm because the theoretical minimum of function exists but it is NP-complete to determine it, since the number of partitions has a factorial growth [17]. Specifically, a greedy top-down approach is used which chooses a variable at each step that best splits the set of items. For measuring the *best* split, different metrics can be used, which generally measure the homogeneity of the target variable within the subsets. Two of the most commonly used ones are:

1. Gini impurity: Breiman et al. used the Gini impurity in their decision tree algorithm in [6]. Let j be the number of classes and p_i the fraction of items of class i in a subset p, for $i \in \{1,2,...,j\}$. Then the Gini impurity is defined as follows:

$$I_G(p) = 1 - \sum_{i=1}^{j} {p_i}^2. \tag{2.1}$$

2. Information gain: Introduced for Decision Trees in [28], it measures the reduction in entropy when applying the split. The entropy is defined as

$$H(t) = - \sum_{i=1}^{j} p_i \log_2 p_i. \tag{2.2}$$

Then we define the information gain to split n samples in parent node p into k partitions, where n_i is the number of samples in partition i as

$$IG = H(p) - \sum_{i=1}^{k} \frac{n_i}{n} H(i). \tag{2.3}$$

Both criteria were found to perform very similarly disagreeing in only 2% of the cases [29]. The hyperparameters of a Decision Tree include the following ones:

- *criterion*: the criterion which decides the feature and the value at the split;
- *max_depth*: the maximum depth of each tree;
- *min_samples_split*: the minimum number of samples in a node to be considered for further splitting.

2.3 Introduction to Random Forests

A random forest is an ensemble model that fits a number of decision tree classifiers on various sub-samples of the dataset which are created by the use of bootstrapping [10]. In the inference stage, a random forest for classification uses a majority vote over all trees to obtain the prediction. Empirical results show that this improves the predictive accuracy as well as controlling over-fitting [3], and therefore, the random forest model is chosen for our experiment.

The hyperparameters of a random forest include the following ones:

- *n_estimators*: the number of trees;
- *criterion*: the criterion which decides the feature and the value at the split;
- *max_depth*: the maximum depth of each tree;
- *min_samples_split*: the minimum number of samples in a node to be considered for further splitting;
- *max_features*: the number of features which are considered for each split.

2.4 Prediction of car availability

Our goal is to predict whether at the specific timestamp in the future the number of cars around a POI is greater than zero or equal to zero.

A car is around a POI p if it is inside the square with length $\pi * 300m/2 \approx 265.87m$ with the location of the POI being the center point. This square approximately has the same area as a circle with radius 300m but it is faster to compute if a point lies inside of it.

For all geograhical distances we adopt the Haversine formula [32] for the great-circle distance, which is shown to have an error of only 0.5% for the latitude and 0.2% for the longitude [26]. Let (ϕ_1, λ_1) and (ϕ_2, λ_2) be the latitude and longitude in radians of two points P_1 and P_2. Then, the central angle between those two points is computed as:

$$\Delta\sigma = 2 \arcsin \sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right)}\,, \tag{2.4}$$

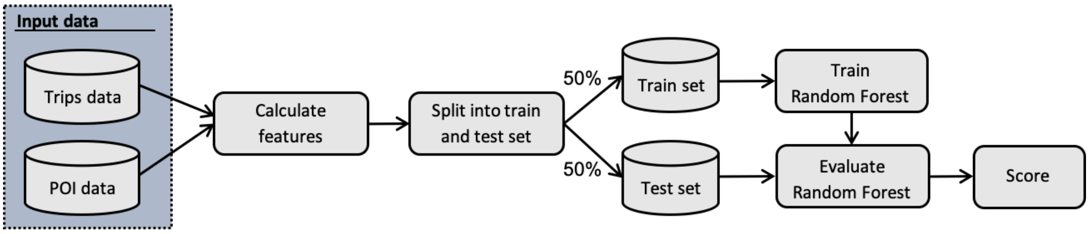


Figure 2.1. Overview of the process of the experiments carried out. First, we extract and calculate all features from the datasets and split the resulting dataset into a train set and test set using the temporally first 50% for training and the following 50% for testing. After that, the random forest model is trained on the train dataset. Then, we predict the labels for the test set and compare them with the true labels in order to evaluate the performance of the random forest and obtain the score.

where $\Delta\phi = |\phi_1 - \phi_2|$ and $\Delta\lambda = |\lambda_1 - \lambda_2|$. In our experiments, we implement a vectorized version of formula 2.4 using *numpy* for speeding up the computation.

Our experimental task is to predict if there is at least one car around a POI at $z + 1$ timesteps in the future, where z is the horizon.

Our process to address this problem is depicted in Figure 2.1. At first, we extract and calculate all relevant features from the datasets. In order to obtain a hold-out test set we split the resulting dataset into train set and test set using the temporally first 50% for training and the following 50% for testing. After that, the random forest model is trained on the train dataset. Then, we predict the labels for the test set and compare them with the ground-truth labels in order to evaluate the performance of the random forest and obtain the score of its performance. This process is done for each POI.

In order to train the random forest model for POI_X, we explore different feature sets which include a subset of the following features:

- The number of cars around POI_X at the last l timestamps in the past;
- The average number of cars around all POIs of a specific category c at the last l timestamps in the past;
- The average number of cars around all POIs, which are between d_{min} and d_{max} meters away from POI_X, at the last l timestamps in the past;
- The weekday, the hour and whether it is 30 minutes into an hour for the timestamp of the prediction;
- The timeslot in the day, when dividing the day equally into t timeslots;
- A binary feature indicating whether the current day is a working day or holiday;
- A set of statistical features for number of cars x_t around POI_X for the period from timestamp t_o to t_p, which is denoted with $x = (x_{t_o}, x_{t_o+1}, \ldots, x_{t_p})$:
  - Arithmetic mean: $\text{mean}(x) = \frac{1}{p-o+1} \sum_{t=t_o}^{t_p} x_t$;
  - Standard deviation: $\text{std}(x) = \sqrt{\text{mean}(|x - \text{mean}(x)|)}$;
  - Minimum, maximum value in x and the range, defined as maximum − minimum;
  - First order change: $\text{foc}(x) = \text{mean}(c_x)$ with $c_x = (|x_{t_o+1} - x_{t_o}|, |x_{t_o+2} - x_{t_o+1}|, \ldots, |x_{t_p} - x_{t_p-1}|)$;

- Second order change: $\text{soc}(x) = \text{foc}(c_x)$ with c_x defined as above;
- Count of timesteps with zero cars around the POI_X, count of timesteps with at least one car around the POI_X.

The resulting predictions of the classifier can be used for giving the users an estimation if they are going to find a car nearby in the future. Another application of this forecasted data can be short-term relocation of the fleet by the car sharing operator in order to optimize the distribution of the vehicles.

Chapter 3

Experiment

3.1 Dataset description

3.1.1 Car Sharing data

The experiments are carried out on databases from the free-floating car sharing providers *car2go*[1] and *enjoy*[2]. The data is from three different cities: Portland, Oregon (US); Seattle, Washington (US); and Turin, Italy.

An overview of the datasets can be seen in Table 3.1. The dataset of Seattle is the smallest one with about 14.5 thousand trips of 744 distinct cars on six days in August 2016. In Turin, we have a total number of 276,818 trips from 790 cars in the time period of April 2016 to October 2016. The oldest but also largest dataset is of Portland containing over 495 thousand trips of 316 cars in 19 months from June 2012 to December 2013.

Table 3.1. The number of trips, distinct cars and the time period of the data in the database of the three cities.

city	#trips	#distinct cars	time period	#days
Seattle	14598	744	2016-08-07 – 2016-08-12	6
Turin	276818	790	2016-04-18 – 2016-08-04	109
Portland	485857	316	2012-06-01 – 2013-12-31	579

3.1.2 Points of Interest data

The data about the Points of Interest (POIs) was extracted from OpenStreetMap[3] leveraging the Overpass API[4] and running the queries on Overpass Turbo[5].

[1] https://www.car2go.com

[2] https://enjoy.eni.com

[3] https://www.openstreetmap.org

[4] http://www.overpass-api.de

[5] https://overpass-turbo.eu

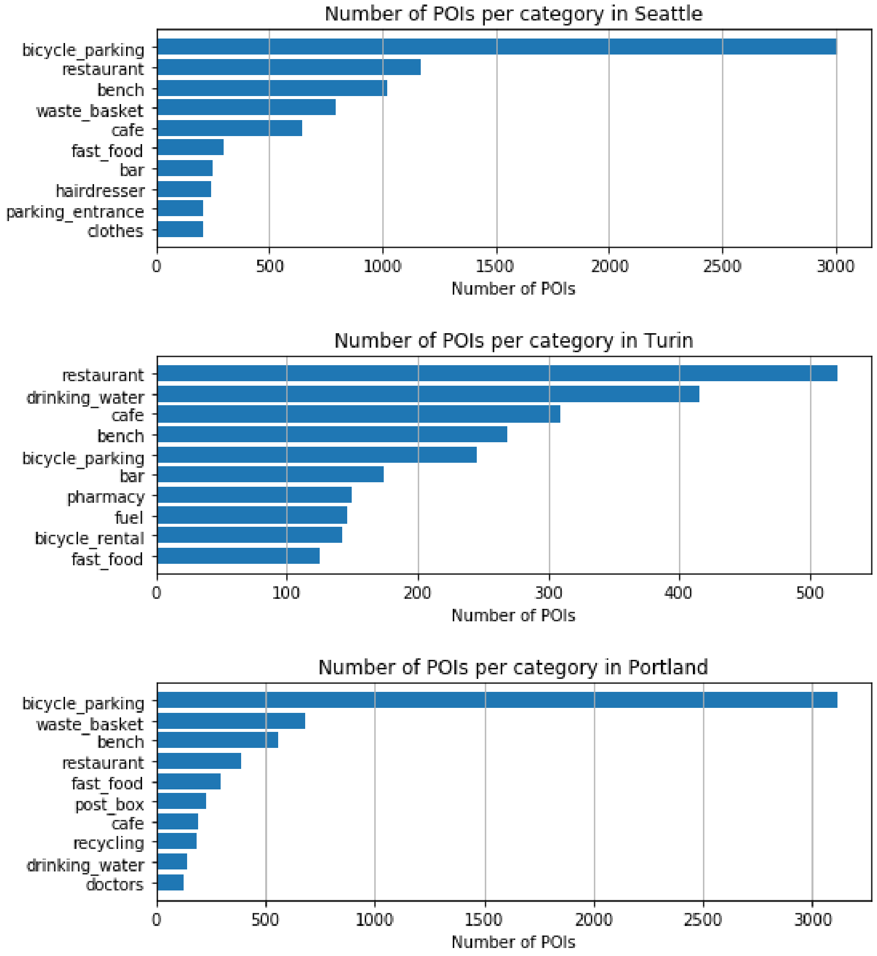


Figure 3.1. Number of extracted POIs for the ten most common categories for each city.

Firgure 3.1 shows the ten most common POI categories for each city along with the number of POIs belonging to them. In Seattle the most common categories are bicycle parking station, restaurant and bench, while for Turin they are restaurant, drinking water station and cafe. The categories of portland are similar to the ones of Seattle but the second most common one is waste basket and restaurant is on position four.

Since many of those categories, such as bicycle parking or bench, do not make sense to incorporate for predicting car usage, we restrict the analysis on the following ten shop and ten amenity categories: beauty, bicycle, car repair, clothes, dry cleaning, florist, pet, sports, supermarket, toys, pub, restaurant, university, clinic, bar, arts centre, bank, fast food, dentist, school.

3.2 Experimental setup

3.2.1 Environment of the experiments

The experiments were conducted in Python 3.5.2 and 3.7.0 using Jupyter Notebook which is a web application that allows creating an interactive environment that contains live code, visualizations, and text. In addition, the following packages were used:

- pandas[6]: pandas provides high-performance data structures and operations for manipulating numerical tables and time series.
- numpy[7]: NumPy provides scientific computing capabilities such as a powerful N-dimensional array object, linear algebra, and random number capabilities.
- sklearn[8]: scikit-learn provides tools for data mining and data analysis.
- matplotlib[9]: matplotlib is a graphing library.
- workalender[10]: workalender is used to determine if a day on a specific date was a working day or holiday.

3.2.2 Feature sets

While having tried 25 different feature sets, we restrict the analysis in this work to the following six feature sets which yielded the most promising results:

- FS1: ten lags into the past of the number of cars around POI_X and of the average of all shops which are 200m-300m away from the POI_X, the weekday, the hour in the day and whether it is 30 minutes into an hour;
- FS2: FS1 with an additional binary feature if the current day is a working day, and the four hour timeslot (00:00-04:00, 04:00-08:00, 08:00-12:00, 12:00-16:00, 16:00-20:00, 20:00-24:00) in the day;
- FS3: FS1 with an additional binary feature if the current day is a working day, and the six hour timeslot (00:00-06:00, 06:00-12:00, 12:00-18:00, 18:00-24:00) in the day;
- FS4: For the period of the last ten recorded timesteps and the period from t_{-50} to t_{-40}, the following statistical measures: arithmetic mean, standard deviation, minimum, maximum, max-min, first order change (mean of differences between consecutive timesteps), second order change (mean of differences of differences between consecutive timesteps), count of timesteps with zero cars around the POI, count of timesteps with at least one car around the POI;
- FS5: The same statistical measures of FS4 in the periods of the last 5 recorded timesteps and the last 10 without the last 5;
- FS6: FS5 and the number of cars around POI_X in the last 10 recorded timesteps.

3.2.3 Parameter settings

For our classification problem, we denote with C_0 the class containing all data points with timestamps, in which there are no cars around POI_X. If there is at least one car around POI_X, the sample belongs to class C_1.

[6] http://pandas.pydata.org

[7] https://www.numpy.org

[8] http://scikit-learn.github.io/stable

[9] https://matplotlib.org

[10] https://github.com/peopledoc/workalendar

Table 3.2. The number of POIs per city and category, and the average probability of having at least one car around a POI (C1) with the corresponding standard deviation.

city	POI category	#POI	avg. $P(C_1)$ [%]
Seattle	shops	2555	60.65 ± 18.97
	amenities	9536	60.15 ± 21.69
Turin	shops	1178	57.83 ± 13.62
	amenities	3645	56.66 ± 15.68
Portland	shops	665	44.82 ± 13.68
	amenities	6993	47.24 ± 14.30

As a preparation step for our experiment, we create a new table counting the number of cars inside the square around each POI at each timestamp. Some statistics about the resulting dataset can be seen in Table 3.2. The average probability of a data point belonging to class C_1 is very similar for shops and amenities in each city, but different between the cities, e.g. shops in Portland have a probability of about 60.65% for having a car around in a specific time, while the probability for shops in Turin is 57.83% and in Portland only 44.82%. These numbers and the high standard deviations show that for many POIs we are having a class imbalance in the target variable.

For the parameters of the random forest we choose the Gini index, defined in 2.1, as criterion and the square root function to limit the number of features considered for each split. In addition, we set the minimum number of samples for a split to the minimum, i.e. two.

Additionally, every classifier has a set of hyperparameters, which can be tuned by training the classifier with different values for these hyperparameters and selecting the classifier with the best score. In order to estimate the performance of a classifier in a more reliable way, k-fold cross-validation (CV) is used. In k-fold CV, the training set is divided into k subsets. Then we train k times our classifier on different unions of k-1 subsets and calculate its score on the subset which was not used for training. Then the final score is calculated by averaging the score of each iteration. In detail, let $C_1, C_2, ...C_k$ be the indices of the samples in each of the K parts of the dataset and let n_k be the number of observations in part k. Then the score from the cross validation is computed as follows:

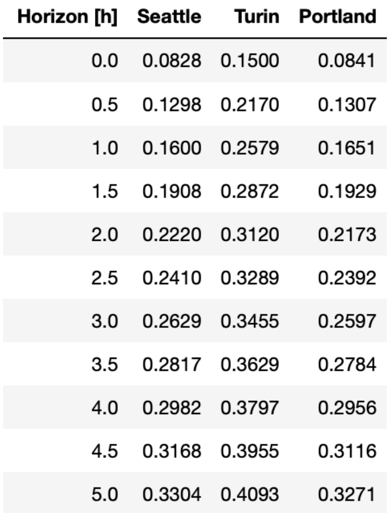
$$\text{Score}_{CV(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \text{Score}_k. \tag{3.1}$$

In our hyperparameter tuning the Score is the F1-Score defined in section 3.3.

In order to determine good parameter settings for each POI specifically, we perform a grid-search with a 5-fold cross validation over the following previously empirically determined sets of parameters:

- *max_depth* $\in \{5, 9, 17, 27\}$;
- *n_estimators* $\in \{300, 800, 1500\}$.

In order to choose an adequate value for the horizon z we analyze the stationarity in the train set, i.e. how probable it is that the value at timestep t_i is not equal to the one at t_{i+z+1}. For a horizon between zero and five hours, this probability does not exceed 33.04% for Seattle, 40.93% for Turin and 32.72% for Portland, which is shown in Figure 3.2. These values are rather low, but our goal is also to keep the application case in mind that an user wants to know if there will be a car available in a certain amount of time. Thus, we choose $z \in \{4, 5\}$ which corresponds to 2/2.5 hours and gives us a stationarity of 22.20% and 24.10% in Seattle, 31.20% and 32.89% in Turin, and 21.73% and 23.92% in Portland.



Horizon [h]	Seattle	Turin	Portland
0.0	0.0828	0.1500	0.0841
0.5	0.1298	0.2170	0.1307
1.0	0.1600	0.2579	0.1651
1.5	0.1908	0.2872	0.1929
2.0	0.2220	0.3120	0.2173
2.5	0.2410	0.3289	0.2392
3.0	0.2629	0.3455	0.2597
3.5	0.2817	0.3629	0.2784
4.0	0.2982	0.3797	0.2956
4.5	0.3168	0.3955	0.3116
5.0	0.3304	0.4093	0.3271

Figure 3.2. The probability that the target value changes from zero to one or vice versa for each city and horizon values of 0-5 hours.

We carry out multiple experiments to investigate the effectiveness of our approach. First, we try out different feature sets for training the classifiers. Then, an analysis of the impact of the time in the day on the performance of the random forest models is conducted, and the impact of choosing a different number of lags into the past for the features is investigated.

In addition, we analyze the pair-wise feature correlation using the sample Pearson correlation coefficient and investigate into selection features according to their correlation with the target variable. Similarly to [12], we define the sample Pearson correlation coefficient for two feature vectors $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ with means $\bar{x}$ and $\bar{y}$ respectively, as

$$r_{xy} = \frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=0}^{n}(y_i - \bar{y})^2}}. \tag{3.2}$$

3.3 Evaluation

3.3.1 Baseline

In order to evaluate the performance of our Random Forest classifier, we choose the last value predictor as baseline, which uses the last recorded value as the prediction. The last recorded value for a prediction at timestamp t_i is the value at timestamp $t_{i-(z+1)}$ for $z \in \{4, 5\}$. Since the stationarity of the series are very high, we expect a rather good performance of the baseline.

3.3.2 Measurement

For the evaluation of the classifiers, a hold-out test set is used, which contains the temporally last 50% of the data. To account for the class imbalance of our target variable, we use the F1-score as our main evaluation metric.

We define:

- TP = #samples for which the prediction is positive and the true label is positive,
- FP = #samples for which the prediction is positive but the true label is negative,

- TN = #samples for which the prediction is negative and the true label is negative,
- FN = #samples for which the prediction is negative but the true label is positive.

Then we define the following metrics:

$$\text{precision} = \frac{TP}{TP + FP}, \tag{3.3}$$

$$\text{recall} = \frac{TP}{TP + FN}. \tag{3.4}$$

With that, the F1-score is given by the following equation:

$$F_1 = 2\,\frac{\text{precision } \times \text{ recall}}{\text{precision } + \text{ recall}}. \tag{3.5}$$

Chapter 4

Results

We now present the results of the experiments predicting future car availability. In order to reduce the computation time the first experiments are executed on ten randomly chosen shops, one per shop-category, while we compare it with the performance for amenities in section 4.5.

4.1 Feature set comparison

In this section, we evaluate the performance of the random forest model trained on different sets of features, which are defined in section 3.2.2.

The classification results using the different feature sets are depicted in Figure 4.1. In Seattle, the feature set FS2 performed the best achieving an about 7 percentage points higher F1-score than the baseline, followed by FS3, which performed about 1.8 percentage points worse compared to FS2. The random forests trained on the three different feature sets containing statistical measures scored even worse than the baseline. On the database of Turin, all six random forests trained on the feature sets clearly outperformed the baseline. However, using the statistical features was again worse than the other feature sets. FS1, FS2 and FS3 performed comparatively well with slight advantages for FS3 at a horizon of two hours and for FS2 at a horizon of 2.5 hours.

Although the feature sets with statistical measures did not perform well in Seattle and Turin, FS6 is closely behind FS2 and FS3 in the Portland dataset. The random forest trained on FS2 yielded a performance which is approximately two percentage points better than the baseline for a two hour horizon and about 2.5 percentage points for the 2.5 hours horizon. The only feature set which could not outperform the baseline was FS4, which shows that the statistical information about the period of the day from the last day is not valuable in this settings.

Since FS2 and FS3 performed similarly on the datasets of Turin and Portland, but FS2 outperformed FS3 in Seattle, we analyze the feature importances of the features in FS2. Figure 4.2 shows the feature importances of the predictions with a horizon of 2 hours for each city. The most important feature in Seattle and Turin is the hour of the day, while for Portland it is the number of cars at the last recorded timestep. The hour of the day still shows some importance in Portland with a value of 0.02. Besides the number of cars at the most recent timesteps, which are heavily important in all three cities, we can see that the weekday, the timeslot of the day and the information if it is a working day show high importance for the predictions in Portland and Turin as well. In Seattle, the mean of the number of cars around the POIs which are 200-300m away has a strong additional impact on the predictions.

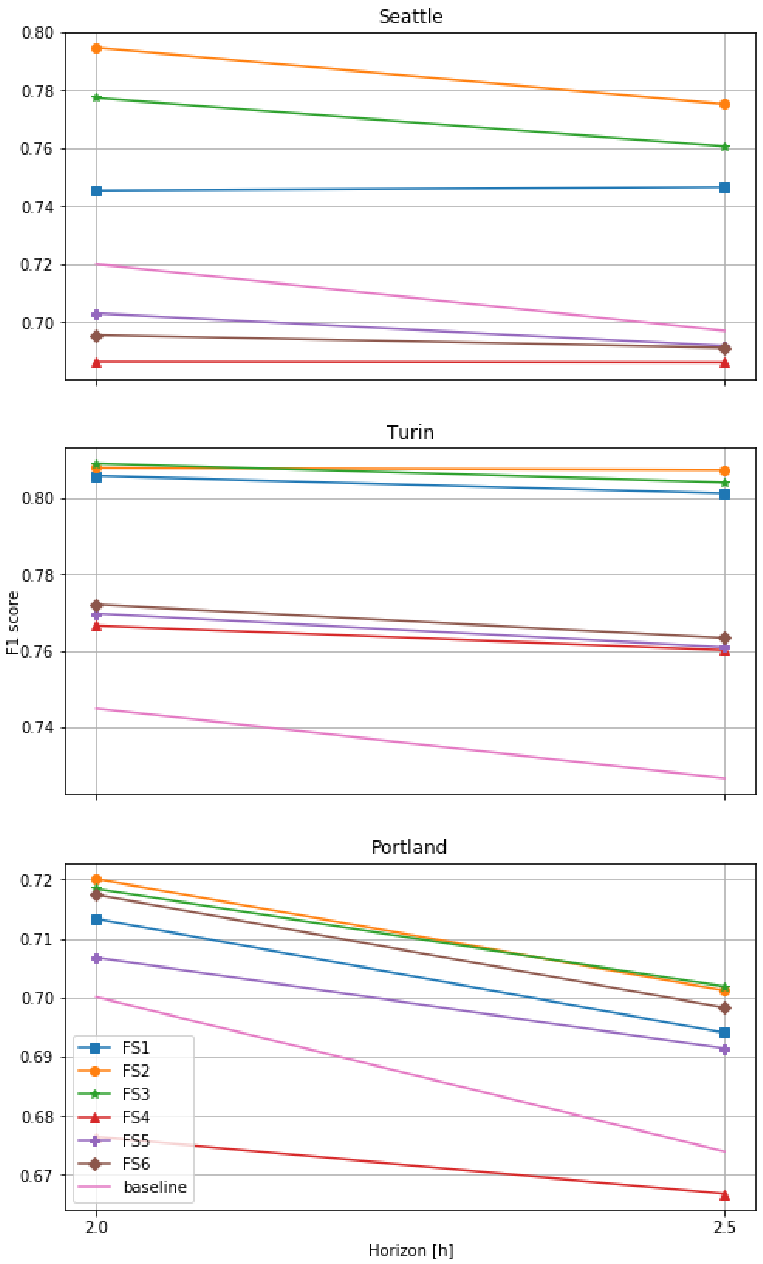


Figure 4.1. Mean F1-score of the random forest using different feature sets versus the baseline.

4.2 Impact of the time of the day

In order to analyze the impact of the time of the day, we equally divide the day into six timeslots and calculate the F1-score of the predictions for each of them. The results are depicted in Table 4.1.

For Seattle, the maximum score of 82.62% is achieved in the timeslot 16:00-20:00. In the

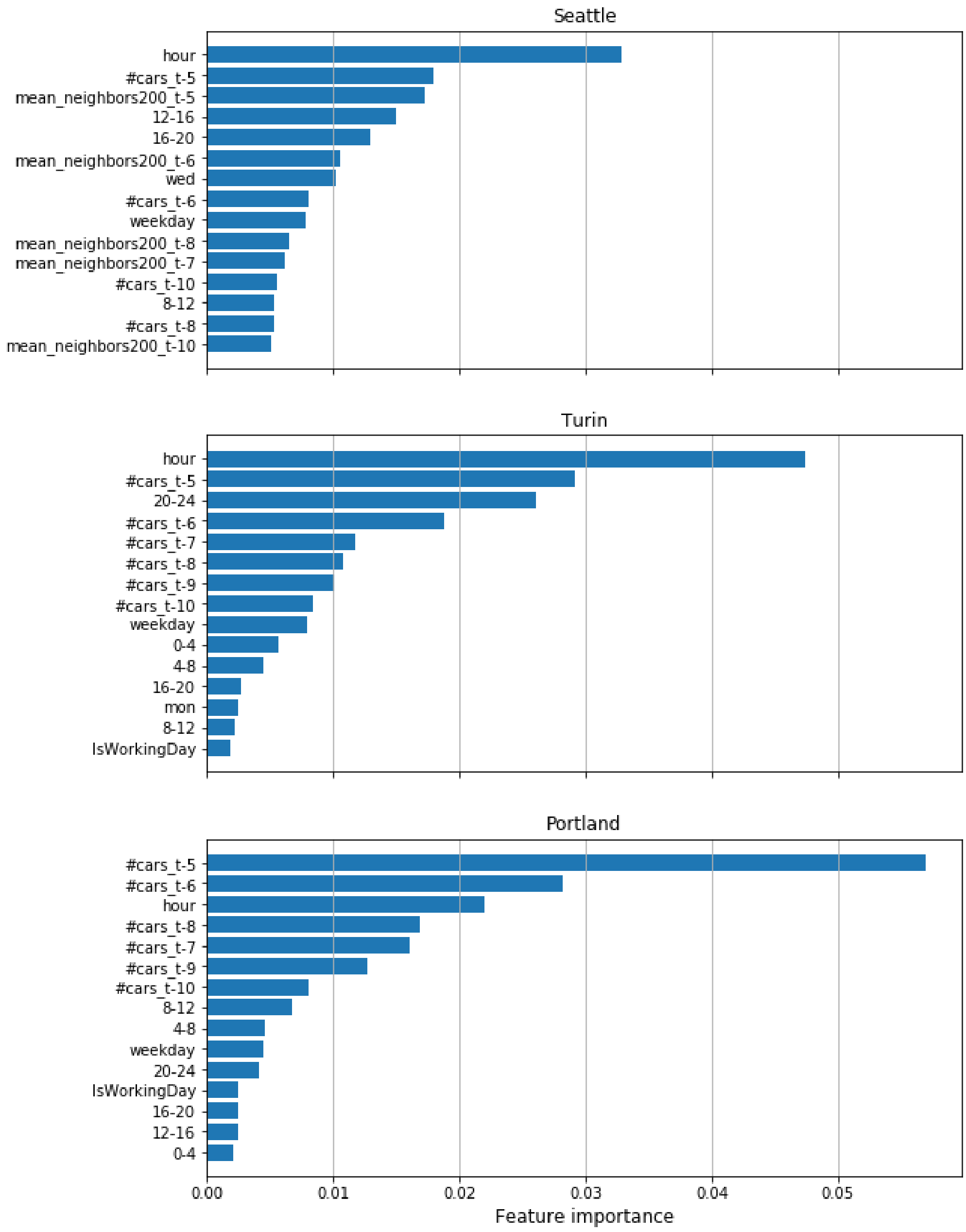


Figure 4.2. Mean feature importances for FS2 for predictions with a horizon of 2 hours.

timeslot before, which is 12:00-16:00, it has the lowest performance with only 57.43%. In Turin however, the classifiers in this timeslot achieved an average F1-score of 86.52% which is the highest for this city. It is closely followed by the 85.24% which were reached in the timeslot from 04:00 to 08:00.

In Portland, the best score of 83.77% was achieved in the early morning from 04:00 to 08:00. The predictions are significantly better than in the other timeslots, as the second highest score is

Table 4.1. The number of trips, distinct cars and the time period of the data in the database of the three cities.

Timeslot	Mean F1 Seattle	Mean F1 Turin	Mean F1 Portland
00:00-04:00	0.6836	0.5967	0.6728
04:00-08:00	0.6588	0.8524	**0.8377**
08:00-12:00	0.6987	0.8256	0.6816
12:00-16:00	0.5743	**0.8652**	0.6175
16:00-20:00	**0.8262**	0.8242	0.5937
20:00-24:00	0.7454	0.5736	0.6219

only 68.16% in 08:00-12:00.

Although there are huge differences between the three cities, the random forest models tend to perform worse during the night time, such as the timeslot 00:00-04:00. The varying scores in the different timeslots for each city show that the time of the day strongly impacts the performance of the predictions.

4.3 Impact of the number of lags

In order to investigate the impact of the number of lags into the past we supply as input to the random forest, we ran the experiment with feature set FS2 for each city and horizon values of 2h and 2.5h. Since the granularity of the timesteps is 30 minutes, 48 lags into the past includes the value recorded at the exact same time but for yesterday. Therefore, the lag values tested are $l \in \{10, 15, 20, 30, 40, 50\}$.

Figure 4.3 shows the results of this experiment. We can see a different behavior for each city. For Seattle, the random forest achieves the highest score when using only ten lags, but interestingly it also performs well with 50 lags. For Turin, the results do not change significantly, but the maximum score is obtained when using ten lags. In Portland the score decreases from using ten lags to using 50 lags by over more than one percentage point for both horizon values. Therefore, for the following experiments we are using only ten lags into the past as input features to the random forest of the time series itself.

4.4 Analysis of the correlation

We also analyzed the correlation between the features themselves and with the target variable. An exemplary correlation heatmap of the features of the FS2 feature set for a POI in Turin called *Beauty Key* can be seen in 4.4. It shows that the highest correlations are between the number of cars at the last timesteps as well as the number of cars in the neighborhood.

Regarding the target variable, we can see some positive correlation between the last recorded number of cars around the POI and the one of POIs in the neighborhood. Interestingly, the target variable does not have a large correlation with the hour which we previously found to be an impactful feature for the predictions of the random forest. We can also identify negative correlation of the target with the timeslot 00:00-04:00 and 20:00-24:00, while for 08:00-12:00 and 12:00-16:00 we see positive correlation. This aligns with our expectation, since clients of a beauty shop go there during the day and not during the night time.

As a next step, we evaluate the performance of the classifier, while only selecting the features which have the highest absolute correlation with the target variable. In order to find out if there are additional interesting features, we extend the feature set FS2 to fifty lags and add the following

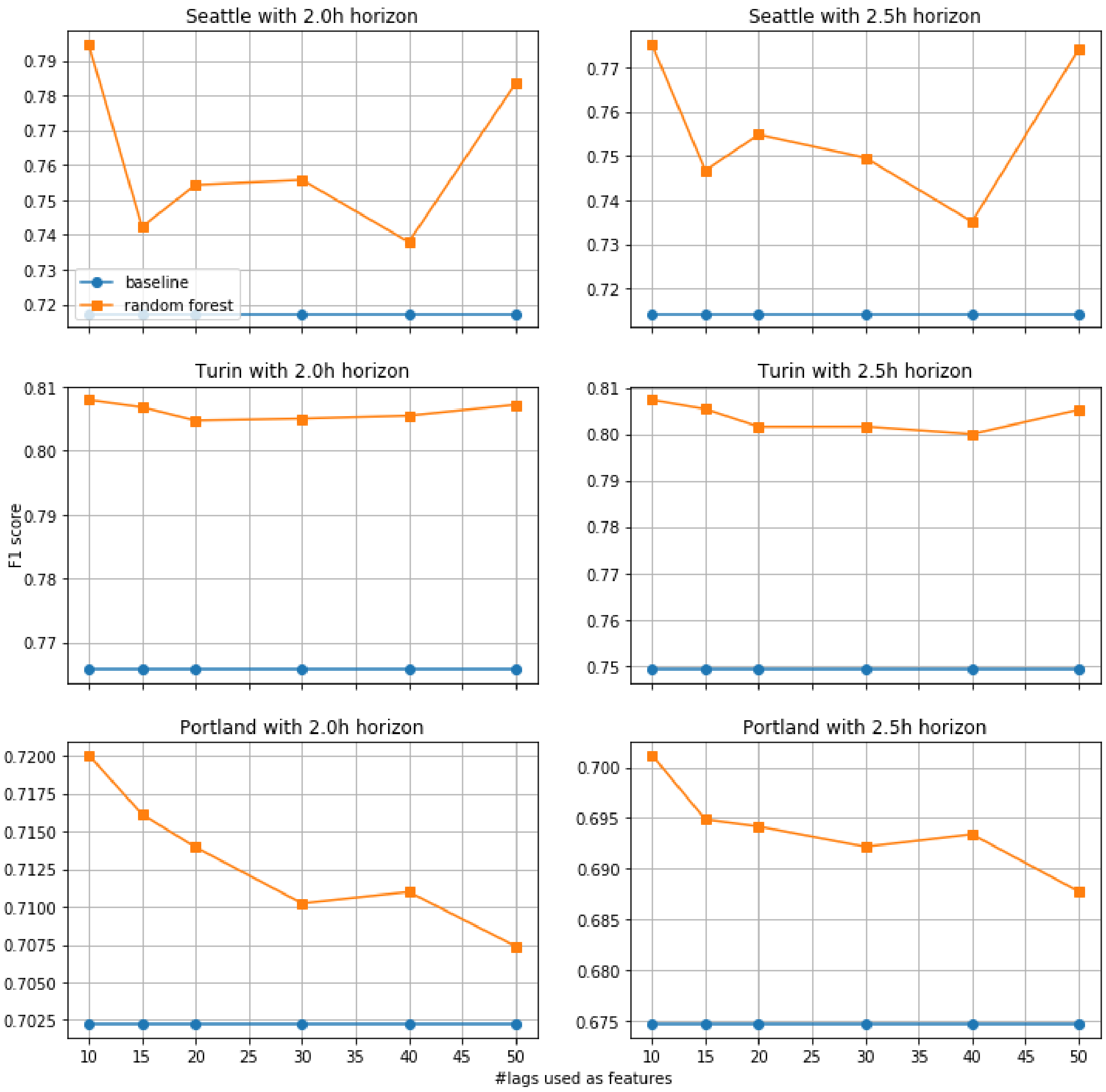


Figure 4.3. F1-score of the experiment using feature set FS2 with varying lags into the past.

features: average of the number of cars around all POIs for each category, average of the number of cars around all POIs which are 200-400m / 500-2000m / 2-4.5km / 10-11km away and the average of the number of cars around all POIs of the same category of the target POI which are 0-500m away from it. This gives us a total of 800 features.

Sorting them by correlation with the target variable, we then train the random forest models on the first x features for $x \in 1, 2, ..., 195$ and compute the F1-score of the predictions for a horizon of two hours. The results are depicted in Figure 4.5. In the case of Seattle, we see an unstable behavior of the score when increasing the number of features. Its F1-score is already peaking at 73.84% using only 15 features. However, this score is significantly lower than the one found for the feature set FS2. Using more than 55 features the model achieves a score which is even worse than the baseline.

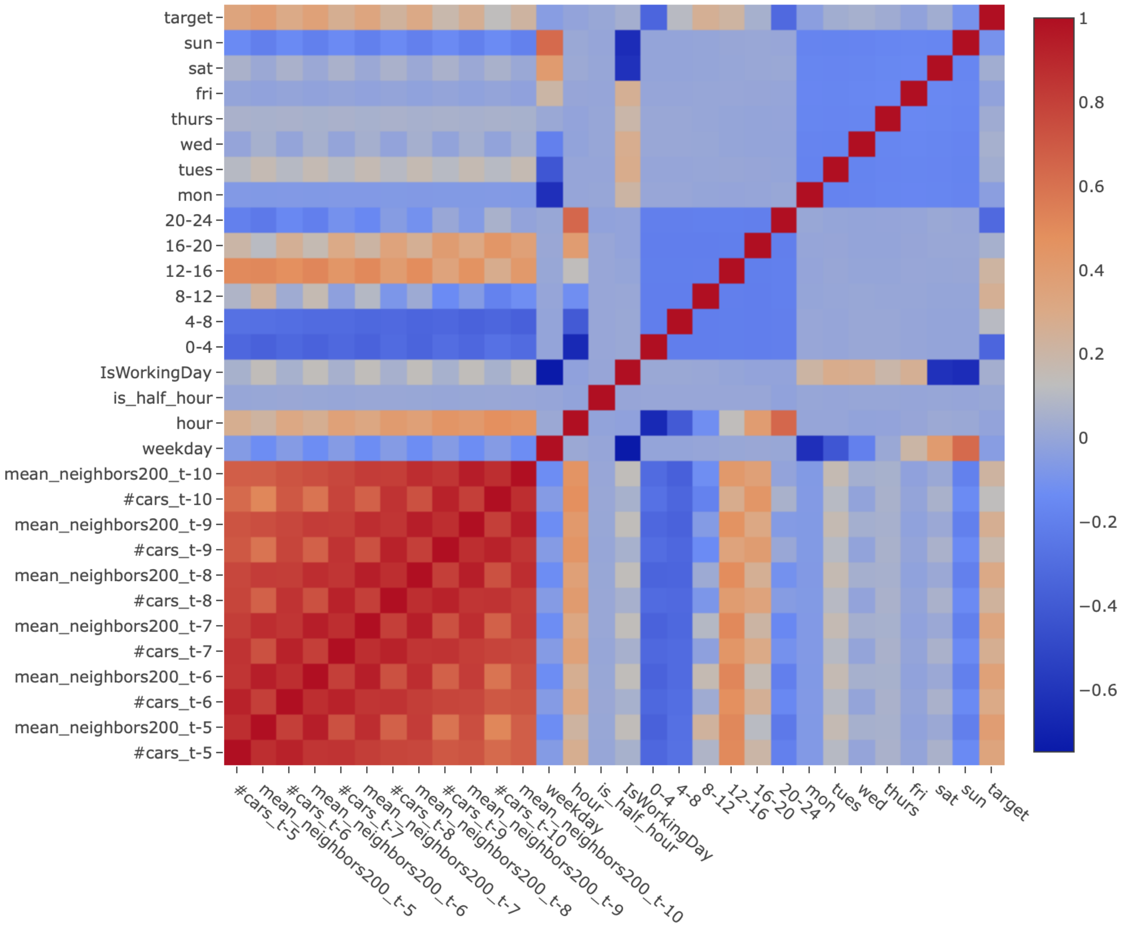


Figure 4.4. Exemplary correlation heatmap of the features of the FS2 feature set for a POI in Turin called *Beauty Key*.

In contrast to that, we can see an almost steady increase of the score for Turin when incrementing the number of features up to 105 features, for which the performance peaks with an F1-score of 80.89%. Additional features do not impact the score significantly. In Portland however, the random forest performs worse or similar than the baseline when using less than 40 features. When incrementing the number of features to more than 40, the random forest outperforms the baseline, peaking at 176 features with a score of 71.24%. Since even the peaking scores obtained from this experiment are lower than the ones of the feature set FS2, which consists of only 29 features, we use FS2 for the next experiment.

4.5 Comparison with performance on amenities

In this section, we explore whether the random forest predictions are also working for POIs which are amenities. One random POI was chosen per category and the random forest model was trained on its data using the FS2 feature set. The mean F1-scores achieved by the classifiers are depicted in Figure 4.6.

In Seattle, the random forest model for the amenities performed significantly worse than the

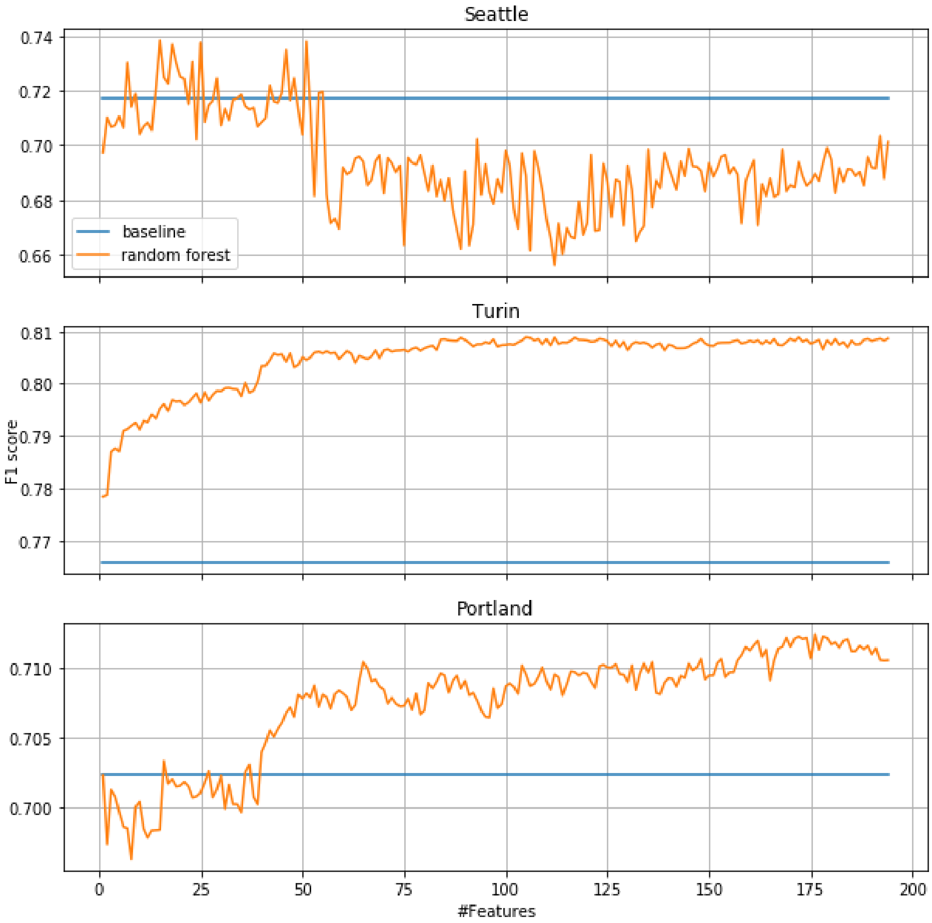


Figure 4.5. Mean F1-scores of random forests trained with the x features which are correlated with the target variable most, for $x \in \{1, 2, ..., 195\}$ and a horizon of two hours.

baseline of the amenities, which had similar scores to the baseline of the shops. In contrast to that, the prediction of the random forests for the POIs of Turin achieved better results than the baselines for both, shops and amenities. Interestingly, also the difference between the score of the random forests and the baselines are with about four percentage points for a two hour horizon and more than five percentage points for a 2.5 hour horizon similar, although the general performance of the random forests and the baseline of the shops performed better than their counterparts of the amenities.

For the dataset of Portland, the random forest classifier clearly outperforms the baselines for both types of POIs. However, differently than in Turin, the classifiers achieve higher scores on the amenities than on the shops. Furthermore, the difference between the random forest of the amenities to its baseline is higher than the one for the shops.

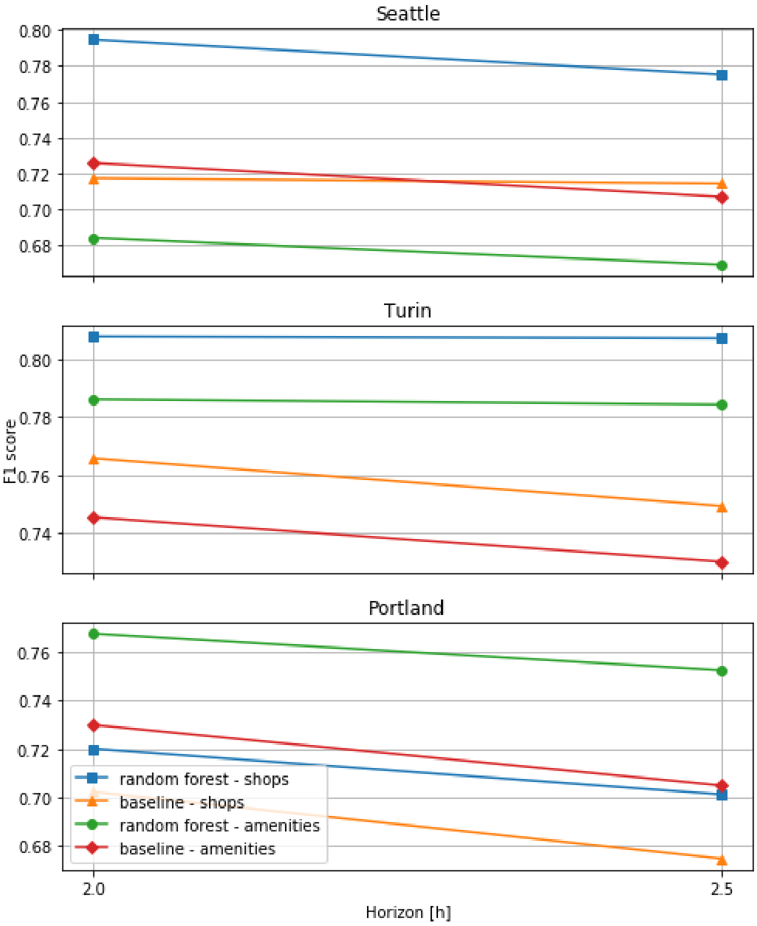


Figure 4.6. Mean F1-score of random forests trained on ten shops and ten amenities.

Part II

Discovery of frequent sequences of Car Sharing vehicles

Chapter 5

Motivation and Background

Identifying mobility patterns of citizens is an important task for cities. For instance, the city has to decide what kind of districts to build in a certain region or how to improve their current transportation systems. Since car sharing data is often openly available and easier to obtain, our goal is to show that extracted sequences of car sharing vehicles can represent the movement of citizens.

5.1 Related work

A lot of research was conducted on behavioral patterns for users of car sharing systems, but are using surveys rather than the data of the car sharing operator itself.

For instance, Becker et. al have identified behavioral patterns for car sharing usage surveying 412 users [4]. They found that most members use a car sharing vehicle in order to visit people, go shopping, go to work or for errands. In [40], POIs belonging to the health sector, restaurants, book stores, banks were found to be very important predictors for car sharing trips. However, to the author's best knowledge, car sharing usage has not yet been modeled as sequences of POIs.

5.2 Introduction to Frequent Sequence Mining

Frequent Sequence Mining, also called Frequent Sequential Pattern Mining, is used to extract a set of patterns which are shared across time among a large number of objects in a given dataset. For instance, Zaki et al. discovered event sequences causing failures in plans in [46]. Another application domain where Frequent Sequential Pattern Mining was used is telecommunication network data, for which the authors identified network alarms in [16]. Kang et al. used Frequent Sequence Mining in biology to find contiguous sequences in DNA and amino acid sequences which typically contain a large number of items [19].

Since in the past Frequent Sequence Mining has been succesfully applied to different types of domains, it could be interesting to apply it in the context of car sharing usage, which to the author's knowledge is the first time of doing so. Using this technique our goal is discovering interesting sequences of frequent trips between different POIs, as well as higher-level sequences between different POI categories. Due to the extremely large search space, the task of discovering all frequent sequences in large datasets is challenging. For instance, with n attributes and a sequence length of at most k there are $O(n^k)$ potentially frequent sequences [45].

5.3 Algorithms for Frequent Sequence Mining

In this section, we give a brief overview of the existing algorithms for Frequent Sequence Mining.

Agrawal and Srikant proposed in [2] AprioriSome and AprioriAll, which are two algorithms to discover frequent sequential patterns. They both scale linearly with the number of transactions. In [33], they introduced a new algorithm called GSP, which discovers more generalized patterns while being faster than the previous approaches. They incorporated time constraints that specify a minimum and maximum time period between two events in the mined sequences. In addition, they added possibilities to extract sequences with items from different transactions and to use taxonomy in the form of an is-a hierarchy to mine sequences with items from all levels. In another work by Hannila et al., WINEP and MINEP were introduced, which extract frequent sequences by only considering a sequence when all its sub-sequences are frequent, and by using an incremental check whether a sequence occurs in a time window [22].

With the aim of incorporating constraints defined as regular expressions (RE) for discovering interesting frequent sequences, Garofalakis et al. proposed the SPIRIT family of algorithms in [13]. The algorithms try to push the RE constraints deep inside the pattern mining computation by exploiting the equivalence of REs to deterministic finite automata [21]. Since RE constraints can be not anti-monotone, the authors have proposed different approaches of dealing with the tradeoff between enforcing the RE constraints at each state and being able to do effective support-based pruning.

In [44], Zaki introduced a novel approach called SPADE which unlike previous approaches usually makes only three database scans by decomposing the original problem into smaller sub-problems using equivalence classes on frequent sequences. SPADE outperforms GSP by a factor of two while having linear scalability with respect to the number of input sequences.

In order to be able to improve the performance of SPADE, Zaki proposed cSPADE which incorporates constraints into SPADE. It supports several different constraints including a minimum and maximum gap, the maximum sequence length and the maximum size of an event in the sequence. Due to its efficiency and ease of use this approach is used for our experiments and therefore described in more detail in the following sections.

5.4 Frequent Sequence Mining Problem Definition

In the following, we formalize the problem of Frequent Sequence Mining similarly to the definition in [45]. Let $I = \{i_1, i_2, ..., i_n\}$ be a set of n distinct items. Furthermore, we define an itemset $\{i_1, i_2, ..., i_k\} \subseteq I$ with $k \geq 1$. Then, a sequence $\alpha = (\alpha_1 \to \alpha_2 \to ... \to \alpha_q)$ is an ordered list of itemsets α_i, where $\to$ denotes a happens-after relationship. The length of sequence α is q and its width is defined as the maximum size of any itemset in it: $width(\alpha) = max(|\alpha_i|),\ 1 \leq i \leq q$.

A sequence α is called a k-sequence if $k = \sum_{i=0}^{q} |\alpha_i|$. For example, the sequences $A \to B, C$ and $A \to B \to C$ are both 3-sequences. In addition, we introduce the notation $\alpha_i < \alpha_j$ for two events α_i and α_j if α_i occurs before α_j in the sequence α.

The sequence α is a subsequence of another sequence β, denoted with $\alpha \preceq \beta$ if there exists a function f that maps events in α to events in β and preserves the order. To formalize this, $\alpha \preceq \beta$, if f fulfills the following conditions:

$$\alpha_i \subseteq f(\alpha_i),\ \ i \in \{1, ..., q\}, \tag{5.1}$$

$$f(\alpha_i) < f(\alpha_j) \text{ for } \alpha_i < \alpha_j \text{ with } i \neq j,\ \ i, j \in \{1, ..., q\}. \tag{5.2}$$

If $\alpha \preceq \beta$, we also say β contains α. For instance, $A \to B$ is a subsequence of $A \to A, B, C$, but $B \to A$ is not a subsequence of $A \to A, B, C$.

A sequence α is called maximal if there does not exist a sequence β such that $\alpha \preceq \beta$.

The database D for sequence mining consists of a set of m input-sequences C_i, $i \in \{1, ..., m\}$. Then, we can define for all subsequences $\alpha \subseteq C_i$, the support as

$$sup(\alpha) = \frac{|\{C_i : C_i \in D \wedge \alpha \subseteq C_i, i = 1, ..., m\}|}{|D|} \ . \tag{5.3}$$

We call a sequence α frequent if $sup(\alpha) \geq \theta$ where θ is a user-defined minimum support threshold. The problem of sequence pattern mining can be stated as the discovery of all frequent subsequences from the database D.

5.5 SPADE

As cSPADE is highly based on SPADE we start by describing the latter one. SPADE [44] exploits the fact that if a sequence β is frequent, all subsequences $\alpha \preceq \beta$ are also frequent. This can be concluded because $\preceq$ defines a partial order on the set of sequences, as shown in the following lemma.

Lemma 5.5.1. *The relation $\preceq$ defines a partial order on the set of sequences.*

Proof. To prove the lemma, we show that the relation $\preceq$ is reflexive, anti-symmetric and transitive.

1. Reflexive relation:

   For every sequence α, there exists a mapping f which maps each itemset α_i to itself and thus $\alpha \preceq \alpha$.

2. Anti-symmetric relation:

   For every pair of sequences α and β with $\alpha \preceq \beta$ and $\beta \preceq \alpha$, we have two mapping functions $f_{\alpha\beta}$ and $f_{\beta\alpha}$. We know that for every i, $\alpha_i \subseteq f_{\alpha\beta}(\alpha_i)$, as well as $\alpha_i \subseteq f_{\beta\alpha}(\alpha_i)$ holds. Since the relation $\subseteq$ is anti-symmetric, we can infer $f_{\beta\alpha}(\alpha_i) = f_{\alpha\beta}(\alpha_i)$. Hence, the sequences α and β are equal, and therefore the relation is anti-symmetric.

3. Transitive relation:

   Let α, β, and γ be sequences with $\alpha \preceq \beta$ and $\beta \preceq \gamma$. Thus, there exist two mapping functions $f_{\alpha\beta}$ and $f_{\beta\gamma}$ with $\alpha_i \subseteq f_{\alpha\beta}(\alpha_i)$ and $\beta_i = f_{\alpha\beta}(\alpha_i)$, and $\beta_i \subseteq f_{\beta\gamma}(\beta_i)$ and $\gamma_i = f_{\beta\gamma}(\beta_i)$ respectively, for every i.

   We define a new mapping function $f_{\alpha\gamma}$ as the composition of $f_{\alpha\beta}$ and $f_{\beta\gamma}$, thus $f_{\alpha\gamma}(\alpha_i) = f_{\beta\gamma}(f_{\alpha\beta}(\alpha_i))$ for every i. Due to the transitivity of the $\subseteq$ relation, we know that for every i $\alpha_i \subseteq f_{\alpha\beta}(\alpha_i) \subseteq f_{\beta\gamma}(f_{\alpha\beta}(\alpha_i))$ holds.

   It is left to show that $f_{\alpha\gamma}(\alpha_i) < f_{\alpha\gamma}(\alpha_j)$ for $\alpha_i < \alpha_j$. We know that for $\alpha_i < \alpha_j$ holds $f_{\alpha\beta}(\alpha_i) < f_{\alpha\beta}(\alpha_j)$ as well as $f_{\beta\gamma}(f_{\alpha\gamma}(\alpha_i)) < f_{\beta\gamma}(f_{\alpha\gamma}(\alpha_j))$, thus $f_{\alpha\gamma}(\alpha_i) < f_{\alpha\gamma}(\alpha_j)$.

Since the relation $\preceq$ is reflexive, anti-symmetric and transitive, it defines a partial order. □

The algorithm performs a depth-first or breadth-first search on the lattice spanned by the subsequence relation from the most general sequences to the maximal ones. Instead of a horizontal database layout, SPADE uses a vertical layout in which each item in the sequence lattice is associated with the input-sequence and itemset identifier containing the item. Every k-sequence with $k \geq 3$ is generated by two $(k-1)$-sequences which can be obtained by dropping one of its first or second elements. By definition, the generating sequences share a common suffix of length $k-2$. For instance, the two generating subsequences of AB → C are A → C and B → C with common suffix C. Performing a temporal join on the identifier lists of the two generating sequences the algorithm iteratively determines the support of any k-sequence by counting the number of distinct identifiers.

Due to limited main memory space, it is often not possible to store all intermediate identifier lists, so SPADE divides the search space into multiple chunks using suffix-based equivalence classes. Two k-sequences belong to the same class if they share a common $k-1$ length suffix. By construction, each class has complete information for generating all frequent sequences that share the same suffix because all generating subsequences are also in the same class.

Starting with suffix classes of length 1, SPADE computes the support of those 1-sequences. Then, it generates frequent sequences by joining the identifier lists of all distinct pairs of sequences in each class and checking the support against the minimum support θ. The sequences found to be frequent form the classes for the next level. This process is recursively repeated until the maximal sequences were reached and all frequent sequences are discovered.

5.6 cSPADE

In order to reduce the extremely large search space and to discover only relevant sequences, cSPADE [45] introduces the possibility to specify constraints for the sequences. Besides the minimum support, these constraints can include:

- maximum sequence length: this constraint limits the maximum number of itemsets in a sequence.
- maximum itemset size: this constraint limits the maximum number of elements for each itemset.
- minimum and maximum gap: these constrain the sequences such that the difference of the two timestamps t_{α_i}, $t_{\alpha_{i+1}}$ of two consecutive itemsets α_i and α_{i+1} is between the minimum and maximum gap:
  $gap_{min} \leq t_{\alpha_{i+1}} - t_{\alpha_i} \leq gap_{max}$.
- maximum window: this constraint limits the difference of the timesteps of the last itemset in the sequence to the first one.

A constraint is class-preserving if the support of any k-sequence can be still computed by joining the identifier lists of its two generating subsequences within the same class. If a constraint is class-preserving, the frequent sequences can be discovered using only the local suffix class information. Otherwise, the enumeration of the sequences has to be changed. Zaki has shown in [45] that all the above-listed constraints but the maximum gap are class-preserving. In the case of the maximum gap constraint, cSPADE performs a join with the set of 2-sequences instead of joining only with all class members. A more detailed description of cSPADE can be found in [45].

5.7 Sequence mining from car sharing data

The main task consists of finding frequent and reliable sequences in the usage of car sharing vehicles. These sequences are extracted on two different levels of abstraction:

- POI level: The events in the sequences contain a set of specific POIs in the neighborhood of the car at the specific timestamp.
- Category level: The itemsets in the sequences contain the set of categories, the POIs in the neighborhood of the car at the specific timestamp belong to.

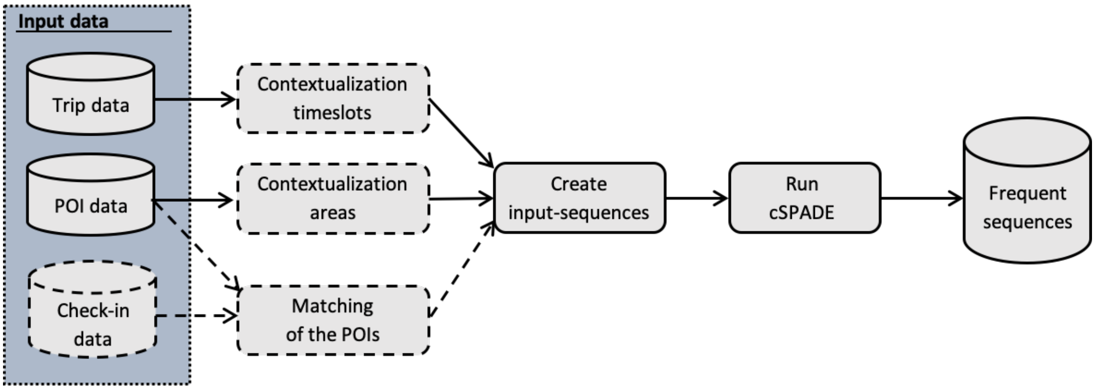


Figure 5.1. Overview of the process of the experiments carried out. First, an optional contextualization step is applied. After that, the input sequences are generated from the car sharing and the POI databases and then cSPADE is used to extract frequent sequences. For one experiment, the POIs of the check-in database are matched with the extracted POI data.

In addition, our goal is to correlate the frequently found sequences with sequences found from a check-in dataset, in which users declare explicitly that they visit a POI at a specific timestamp. Since some of the works described in section 5.1 suggested that temporal factors, as well as spatial factors can highly impact the use of car sharing services, we further perform a temporal and a spatial contextualization in order to compare the discovered sequences between different times in the day and different areas.

The basic process of the experiment is depicted in Figure 5.1. In a first step, the input sequences are generated by extracting the POIs in a radius of 300 meters from the position of a car at the start and end points of each trip with the car sharing vehicle. In order to provide discrete timesteps for the input to cSPADE the timestamps are rounded to the highest granularity of the raw car sharing input, which is 15 minutes. Then each sequence of each car is cut at midnight so that we obtain one sequence for each car per day detailing the POIs in the neighborhood of car's current position.

In the next step, we apply the open-source implementation of the cSPADE algorithm[1] which is written in C++. After applying cSPADE, a post-processing step is performed to evaluate the mined sequences. The sequences which are found to be frequent and reliable can then be used to identify user behavior of car sharing, as well as movement patterns of citizens.

[1] https://github.com/zakimjz/cSPADE

Chapter 6

Experiment

6.1 Dataset description

For this experiment, we use three different datasets. First, the car sharing database and the POI database, which were also used in the experiment in chapter 3. In addition, we correlate our extracted sequences with a third dataset, stemming from the location-based social network Squarespace which was extracted and analyzed in [41, 42].

6.1.1 Car Sharing data

The same data is used as described in section 3.1.1. However, we do not consider the data of Seattle due to its limited time period of only six days.

6.1.2 Points of Interest data

The same data is used as described in section 3.1.2. In order to decrease the computation time for some experiments, we reduce the POIs by only selecting POIs belonging to one of the following categories: beauty, bicycle, car_repair, clothes, dry_cleaning, florist, pet, sports, supermarket, toys, pub, restaurant, university, clinic, bar, arts_centre, bank, fast_food, dentist, school.

For Portland the resulting dataset consists of 1115 POIs in total. Figure 6.1 depicts their distribution over the categories. There are 390 restaurants and 294 fast food restraurants, which together make up over 61% of all POIs in Portland.

In Turin however, we have 1374 POIs in total. The most prominent category is restaurant with 521 venues. In contrast to Portland, there are only 125 fast food POIs, but the number of supermarkets is with 100 much higher than in Portland, which has only 13.

6.1.3 Check-in data

In order to compare the usage of car sharing services with user behavior data, we use a dataset containing 33,278,683 check-ins by 266,909 users on 3,680,126 venues (in 415 cities of 77 countries) [43]. However, we match the venues with the POI data of Portland resulting in a reduced database of 1701 check-ins by 480 users on 532 venues in the time period from 2012-04-03 to 2013-09-16. This gives an average number of check-ins per user of approximately 3.54, which is low considering the large time period.

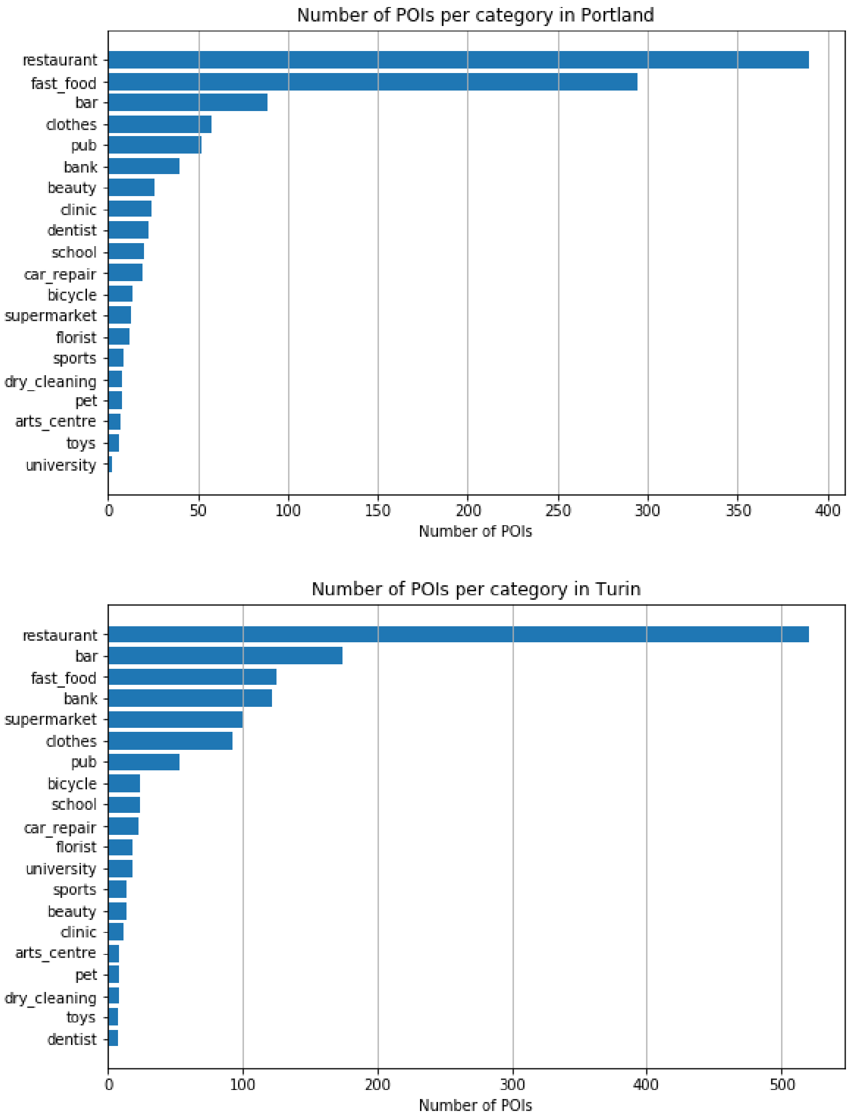


Figure 6.1. The distribution of the POIs in Portland and Turin over the categories.

6.2 Experimental setup

We configure the cSPADE algorithm with the following constraints:

- minimum gap: 1 timestep (corresponding to 15 minutes);
- maximum gap: 4 timesteps (corresponding to 60 minutes);
- maximum sequence length: 4;
- maximum itemset size: 2.

We do not specify a maximum window, because this is already limited by the maximum sequence length and the minimum and maximum gap, leading to a maximum window of $4 * 4 - 1 = 15$

timesteps which corresponds to 3.75 hours. Furthermore, we perform the following two types of contextualization:

- Temporal contextualization: We define four different timeslots based on the time in the current day: [00:00, 06:00), [06:00 - 12:00), [12:00 - 18:00), [18:00 - 24:00). Please note that the lower bound is inclusive while the upper bound is exclusive. Hence, the dataset is divided into four databases, one for each timeslot, which are then fed into the cSPADE algorithm to discover sequences separately for each timeslot.

- Spatial contextualization: The operating area is divided into five different areas: the city center, and one respective region from the center spanning into each direction (north, east, south, west). These areas are depicted in Figure 6.2 for Portland and in Figure 6.3 for Turin. Downtown Portland, the city center of Portland, is located at $D = (45.51935, -122.67962)$ [1]. We define the city center area as a square with middle point D and a length of three kilometers. Then, the other areas are defined by connecting the edges of the city center to the edges of the operating area. Similarly, we define the city center area in Turin, but with a length of 2.4 kilometers to account for the smaller overall operator's area.

  Using these areas we assign each POI to one of them. Table 6.1 shows the resulting distribution of the POIs over the areas. Due to the low number of POIs in the south and west area in Portland, we focus our experiment for the area contextualization on the city center, the north, and the east area. Thus, we create three datasets, one for each area, with the trips conducted within the corresponding area. For Turin, we choose to use the city center, the north, the south, and the west area. These areas together contain 97.16% of the POIs.

  Similarly, we divide the operator's area of Turin into five areas and choose to restrict our analysis on the city center, north, south and west area, because they contain most of the POIs.

Table 6.1. The distribution of the POIs in Portland and Turin over the specified areas.

	Portland		Turin	
Area	#POIs	Percentage	#POIs	Percentage
City center	382	34.26	661	48.11
North area	150	13.45	158	11.50
East area	488	43.77	39	2.84
South area	70	6.28	191	13.90
West area	25	2.24	325	23.65

6.3 Evaluation

In order to evaluate the extracted sequences, we further define the concept of the confidence of a sequence. The definition of a sequence and its support, both defined in section 5.4 should also hold here. Then the confidence of a sequence $\alpha = (\alpha_1 \rightarrow \alpha_2 \rightarrow ... \rightarrow \alpha_q)$ is defined as the probability of α_q happening under the condition that the events $(\alpha_1, \alpha_2, ..., \alpha_{q-1})$ happened before in the specific order $\alpha_1 \rightarrow \alpha_2 \rightarrow ... \rightarrow \alpha_{q-1}$. Hence, the confidence is computed as follows:

$$conf(\alpha) = \frac{sup(\alpha)}{sup(\alpha_1 \rightarrow \alpha_2 \rightarrow ... \rightarrow \alpha_{q-1})} \ . \tag{6.1}$$

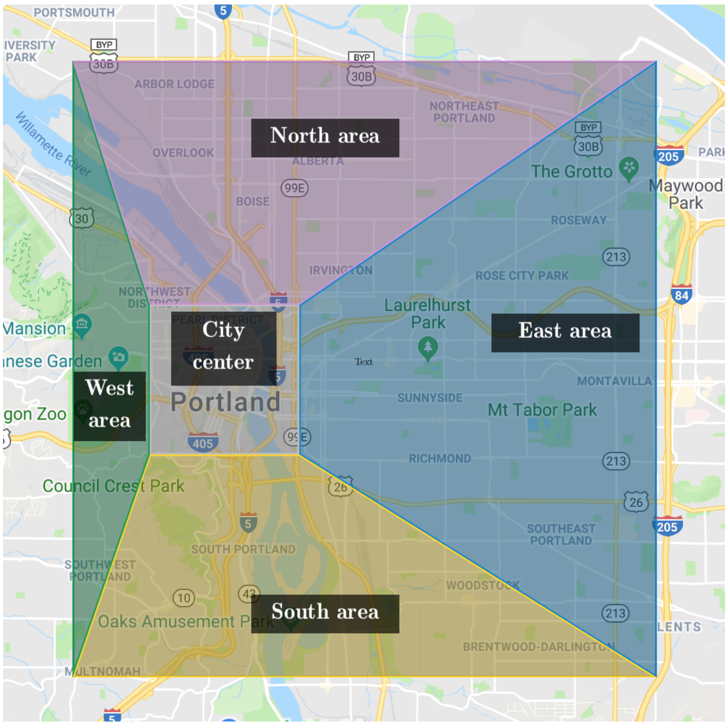


Figure 6.2. The specified areas in Portland which are used for the spatial contextualization. Map generated with Google Maps[2].

For a sequence A → B → C, the confidence expresses how likely it is that after A → B happened, we will observe C. In the car sharing context, a high confidence means that a user who was driving from A to B is likely to move to C next.

For the evaluation of both the car sharing and the check-in dataset we further introduce the following metrics:

- **Mean Reciprocal Rank (MRR):** The MRR was proposed by Vorhees in [35] and it measures how high an item is ranked in a ranking. For a set of queries Q we define:

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}, \tag{6.2}$$

  where rank_i is the position in the ranking of the first relevant item for the ith query.

- **Precision at k (P@k):** P@k evaluates how many items in the top k items are relevant relatively to the total number of extracted items, which is k [23].

- **Recall at k (R@k):** This metric measures how many items in the top k items are relevant with respect to the total number of relevant items.

[2] https://www.google.com/maps

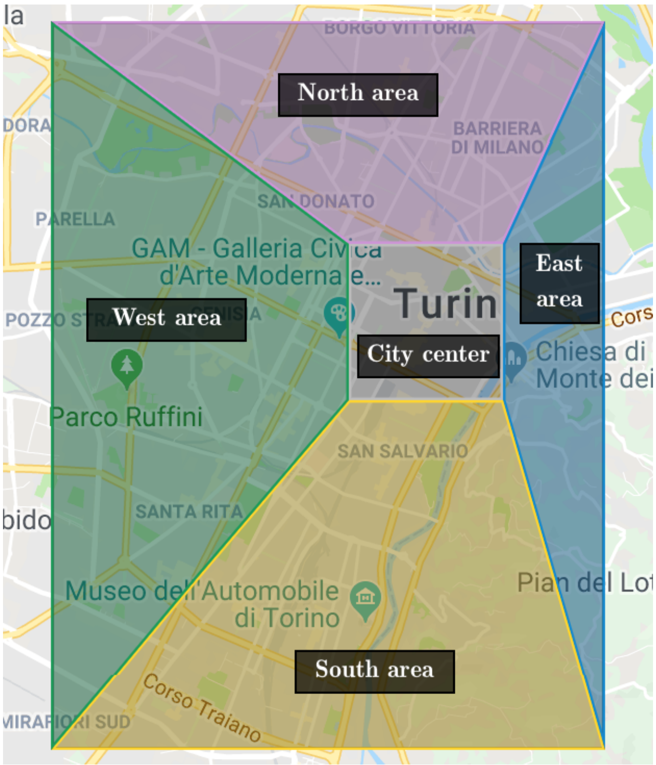


Figure 6.3. The specified areas in Turin which are used for the spatial contextualization. Map generated with Google Maps.

Chapter 7

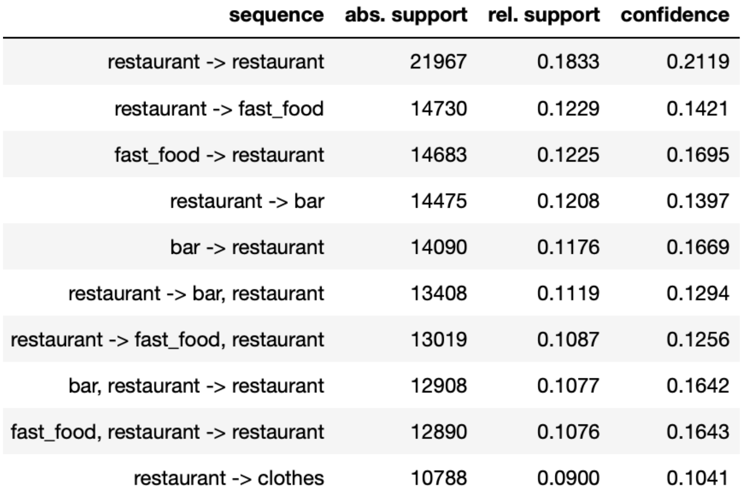
Results

In this chapter, we compare the discovered sequences of the previously introduced experiments and illustrate the results. They provide interesting insights on the user behavior in car sharing systems, but also show that not knowing with certainty which POI exactly is the user's destination gives us some sequences which are not logical and could be misleading.

7.1 Frequent sequences in the car sharing dataset

We start by extracting the sequences from the car sharing database. For this experiment, we are using an adaptive minimum support threshold, starting at a relative minimum support value of 0.1, which is divided by 2 until we discovered at least 1000 sequences.

Under the 1000 sequences with the highest support of Portland, we discovered 995 sequences of length 2 and 5 of length 3. 51% of those sequences have an average itemset size of 1.5, while 36.3% have just one item in each event, and 12.7% have an average of 2 items in each of the itemsets.



sequence	abs. support	rel. support	confidence
restaurant -> restaurant	21967	0.1833	0.2119
restaurant -> fast_food	14730	0.1229	0.1421
fast_food -> restaurant	14683	0.1225	0.1695
restaurant -> bar	14475	0.1208	0.1397
bar -> restaurant	14090	0.1176	0.1669
restaurant -> bar, restaurant	13408	0.1119	0.1294
restaurant -> fast_food, restaurant	13019	0.1087	0.1256
bar, restaurant -> restaurant	12908	0.1077	0.1642
fast_food, restaurant -> restaurant	12890	0.1076	0.1643
restaurant -> clothes	10788	0.0900	0.1041

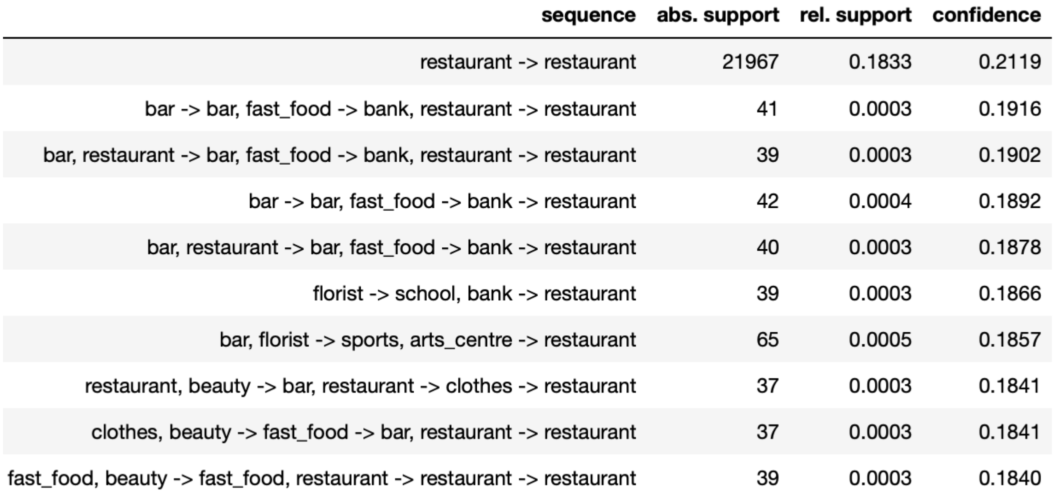
Figure 7.1. The discovered sequences of Portland on category level from the car sharing data, sorted by the support. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

On the category level, the most frequent sequence is *restaurant* → *restaurant*, as shown in

Figure 7.1. This sequence itself is not logical because a user should usually be satisfied after having eaten a meal in the first restaurant. Hence, this sequence can be interpreted in different ways, for example, that the user drove the car from one area with a lot of restaurants to another one with a high density of restaurants, or that it is likely for users to take and return the car-sharing vehicles near restaurants. This shows a limit of sequence pattern mining in this context because we cannot draw a final conclusion of what exactly the reason is for this sequence.

The sequences *restaurant* → *bar* and *bar* → *restaurant* are more interesting because people are having an aperitif before the meal or going out to a bar after dinner. These sequences suggest with high support values of 12.08% and 11.76% with confidence 13.96% and 16.69%, respectively, that users do not hesitate to use a car sharing vehicle to move from the restaurant to the bar or vice-versa.

The most frequent sequences on the POI level, as well as the most confident ones, can be found in A.1 and A.2. Those sequences are mostly examples for the category sequences discussed above.



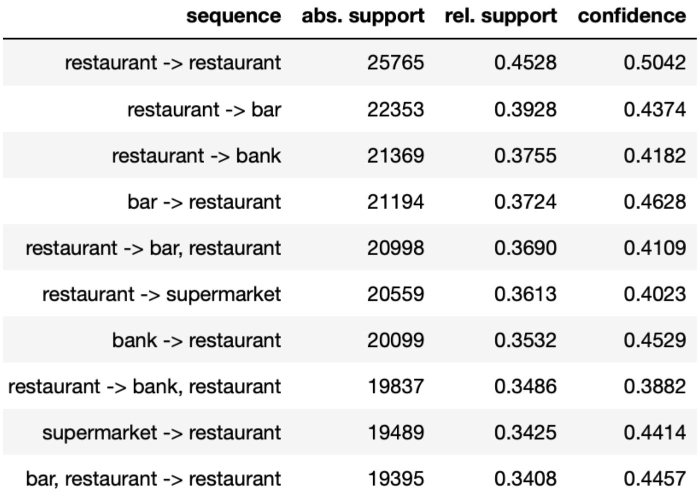
sequence	abs. support	rel. support	confidence
restaurant -> restaurant	21967	0.1833	0.2119
bar -> bar, fast_food -> bank, restaurant -> restaurant	41	0.0003	0.1916
bar, restaurant -> bar, fast_food -> bank, restaurant -> restaurant	39	0.0003	0.1902
bar -> bar, fast_food -> bank -> restaurant	42	0.0004	0.1892
bar, restaurant -> bar, fast_food -> bank -> restaurant	40	0.0003	0.1878
florist -> school, bank -> restaurant	39	0.0003	0.1866
bar, florist -> sports, arts_centre -> restaurant	65	0.0005	0.1857
restaurant, beauty -> bar, restaurant -> clothes -> restaurant	37	0.0003	0.1841
clothes, beauty -> fast_food -> bar, restaurant -> restaurant	37	0.0003	0.1841
fast_food, beauty -> fast_food, restaurant -> restaurant -> restaurant	39	0.0003	0.1840

Figure 7.2. The discovered sequences of Portland on category level from the car sharing data, sorted by the confidence. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

Figure 7.2 shows the discovered sequences with the highest confidence values. We can also identify also longer sequences such as *florist* → *school, bank* → *restaurant*. Although it has a high confidence value of 18.57%, the support value is with just 39 occurrences very low. A similar behavior can be seen for the other extracted sequences.

For the Turin dataset, we discovered 819 sequences of length 2 and 181 of length 3 in the 1000 sequences with the highest support. 38.8% of those sequences have an average itemset size of 1.5, while 34.1% have an average of two items in each event, and 15.1% consist of only one item in each of the itemsets. The 10.2% and 1.8% remaining sequences have an average itemset size of 1.33 and 1.67, respectively.

The extracted sequences of Turin are similar to the ones of Portland. Figure 7.3 depicts the most frequent sequences on category level. As in Portland, the sequence *restaurant* → *restaurant* has with 45.28% the best relative support value. Its confidence is 50.42%, i.e. every second trip which starts near a restaurant also ends in the neighborhood of a restaurant. The second most frequent sequence is *restaurant* → *bar* with relative support of 39.28% and confidence of 43.74%. In contrast to Portland, there are no fast food POIs in the most frequent sequences, which could be due to the fact that fast food is more popular in the USA than in Europe.



sequence	abs. support	rel. support	confidence
restaurant -> restaurant	25765	0.4528	0.5042
restaurant -> bar	22353	0.3928	0.4374
restaurant -> bank	21369	0.3755	0.4182
bar -> restaurant	21194	0.3724	0.4628
restaurant -> bar, restaurant	20998	0.3690	0.4109
restaurant -> supermarket	20559	0.3613	0.4023
bank -> restaurant	20099	0.3532	0.4529
restaurant -> bank, restaurant	19837	0.3486	0.3882
supermarket -> restaurant	19489	0.3425	0.4414
bar, restaurant -> restaurant	19395	0.3408	0.4457

Figure 7.3. The discovered sequences of Turin on category level from the car sharing data, sorted by the support. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

Sorting the sequences by confidence, did not reveal any additional interesting sequences, but the interested reader is referred to Figure A.3, which shows the mined sequences with the highest confidence values.

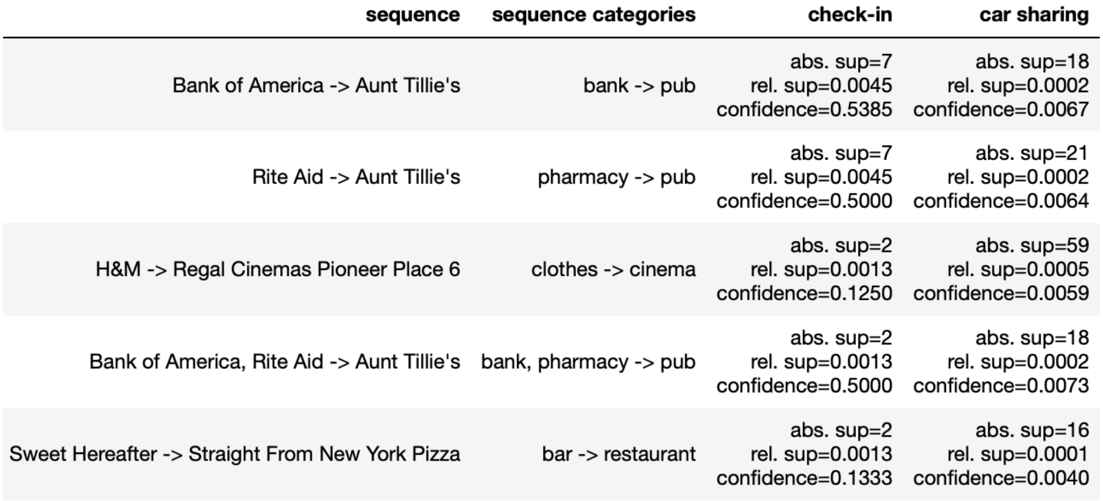
7.2 Comparison with the check-in dataset

In order to compare the extracted sequences from the car sharing database, we also extract sequences from explicit user check-ins to venues. In this experiment, we do not use the reduced version of the POIs in order to have a larger matched dataset with the venues of the check-in database. After the discovery of the sequences in both datasets with the restricted set of POIs containing only matched POIs and a minimum relative support threshold of 10^{-5}, we match the extracted sequences from both datasets. There were a total of 31 sequences matched on the category level and 30 on POI level. The Figures 7.4 and 7.5 show the sequences with the highest support in the check-in database.

On the POI level, we can identify sequences like *Bank of America* → *Aunt Tillie's*, which are discovered in both databases, but have rather low absolute support with just 7 occurrences in the check-in dataset and 18 in the car sharing one. Furthermore, there are also some people which might take an aperitif from the bar *Sweet Hereafter* and go eating at the restaurant *Straight From New York Pizza* afterwards.

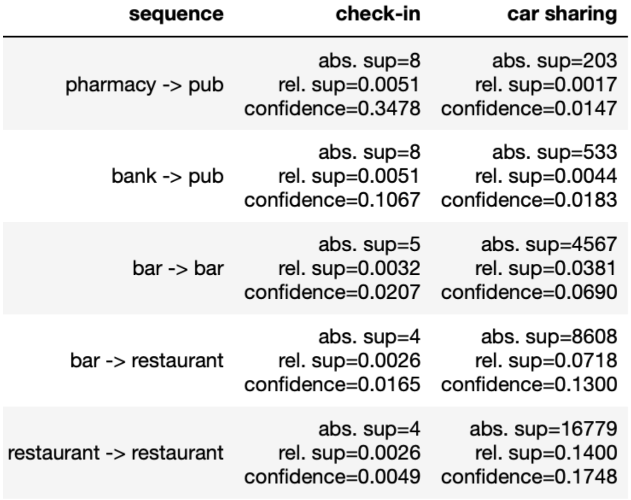
For the first sequence on the category level, *pharmacy* → *pub*, we can see how useful it can be to have data about the user behavior from the check-in dataset, because one might not expect that many users go to visit a pub after having been to a pharmacy. This sequence was also extracted using the car sharing data with absolute support of 203 and a confidence of 1.47%.

The sequence *bank* → *pub*, which was also discovered in both datasets as being frequent can be interpreted as such that users go to a bank, possibly to take some cash, which they can then spend in a pub. There are also some other interesting sequences like *bar* → *restaurant*, but the overall absolute support values of the check-in dataset are with a maximum of eight too low to be representative of the society. There was no larger check-in database available to the authors of this work.



sequence	sequence categories	check-in	car sharing
Bank of America -> Aunt Tillie's	bank -> pub	abs. sup=7 rel. sup=0.0045 confidence=0.5385	abs. sup=18 rel. sup=0.0002 confidence=0.0067
Rite Aid -> Aunt Tillie's	pharmacy -> pub	abs. sup=7 rel. sup=0.0045 confidence=0.5000	abs. sup=21 rel. sup=0.0002 confidence=0.0064
H&M -> Regal Cinemas Pioneer Place 6	clothes -> cinema	abs. sup=2 rel. sup=0.0013 confidence=0.1250	abs. sup=59 rel. sup=0.0005 confidence=0.0059
Bank of America, Rite Aid -> Aunt Tillie's	bank, pharmacy -> pub	abs. sup=2 rel. sup=0.0013 confidence=0.5000	abs. sup=18 rel. sup=0.0002 confidence=0.0073
Sweet Hereafter -> Straight From New York Pizza	bar -> restaurant	abs. sup=2 rel. sup=0.0013 confidence=0.1333	abs. sup=16 rel. sup=0.0001 confidence=0.0040

Figure 7.4. The discovered sequences on POI level from the check-in data and the car sharing data, sorted by the support in the check-in data. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.



sequence	check-in	car sharing
pharmacy -> pub	abs. sup=8 rel. sup=0.0051 confidence=0.3478	abs. sup=203 rel. sup=0.0017 confidence=0.0147
bank -> pub	abs. sup=8 rel. sup=0.0051 confidence=0.1067	abs. sup=533 rel. sup=0.0044 confidence=0.0183
bar -> bar	abs. sup=5 rel. sup=0.0032 confidence=0.0207	abs. sup=4567 rel. sup=0.0381 confidence=0.0690
bar -> restaurant	abs. sup=4 rel. sup=0.0026 confidence=0.0165	abs. sup=8608 rel. sup=0.0718 confidence=0.1300
restaurant -> restaurant	abs. sup=4 rel. sup=0.0026 confidence=0.0049	abs. sup=16779 rel. sup=0.1400 confidence=0.1748

Figure 7.5. The discovered sequences on category level from the check-in data and the car sharing data, sorted by the support in the check-in data. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

To further evaluate the importance of the sequences from the car sharing database with the sequences of the check-in dataset, we computed the MMR, the P@k and the R@k. The results are presented in Figure 7.6. We can see that the MRR is with about 0.09 to 0.20 rather high for k less than seven. After that, it constantly decreases as no more sequences are found to have a high ranking in the check-in dataset.

The precision at k shows a similar picture. For $k \leq 3$, all extracted sequences also appear in the check-in dataset, but as we take into account more extracted sequences we can see that we have several misses in the check-in dataset, which reduces the precision. Hence, this metric shows that the first extracted sequences are very reliable, but as we consider more sequences the P@k

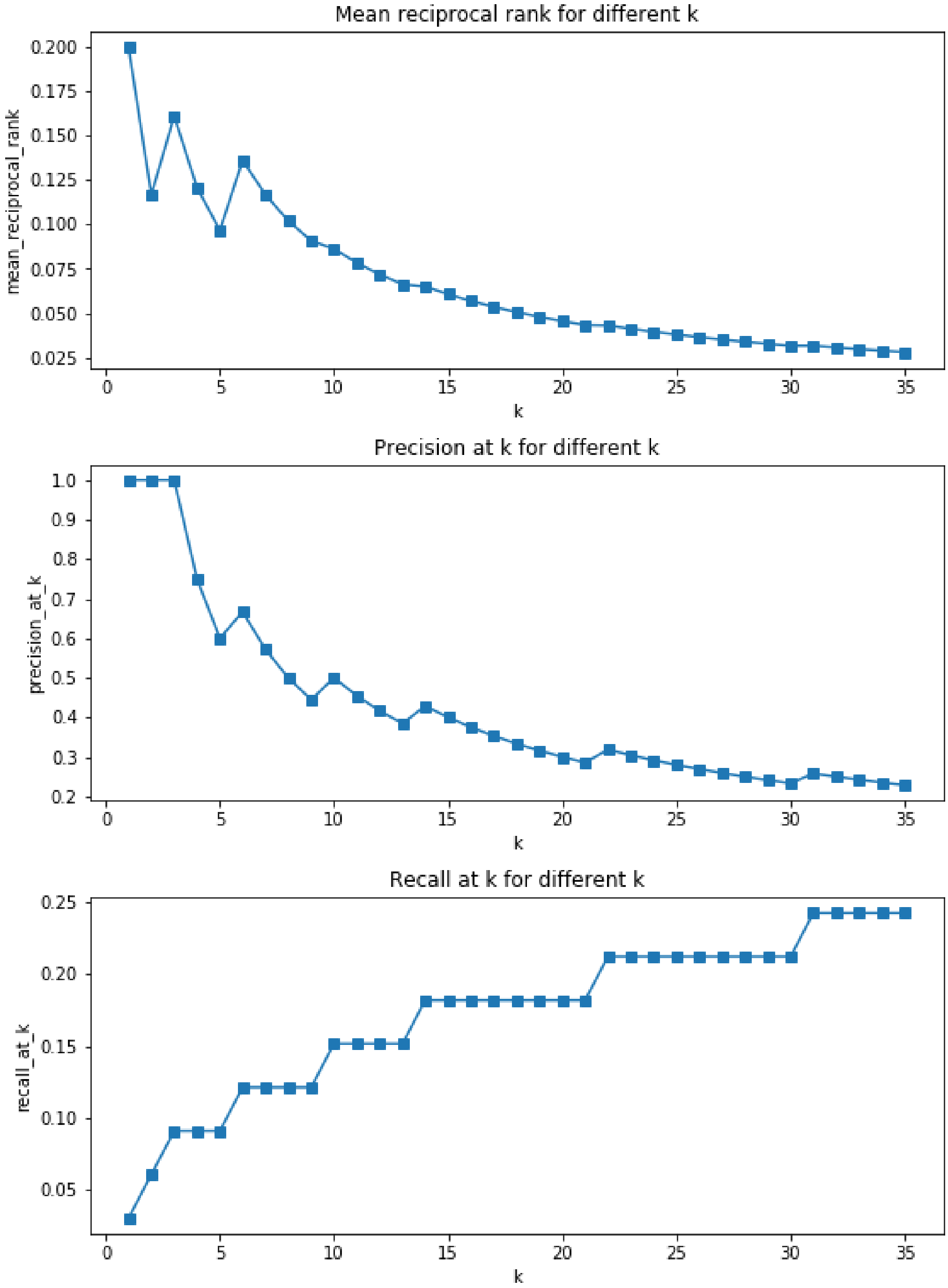


Figure 7.6. The mean reciprocal rank, the precision at k, and the recall at k for the extracted sequences from the car sharing dataset in comparison with the ground truth sequences from the check-in dataset, for $k \in \{1, 2, ..., 35\}$.

decreases. Furthermore, the figure shows that for the recall at k we see a curve which is increasing for each hit. It reaches a maximum of about 0.24 considering the top 31 sequences of the car sharing dataset. Thus, it indicates that by using only a few sequences of the car sharing dataset we can resemble already nearly one-quarter of the sequences of the check-in dataset.

In addition, we conducted the analysis the other way around, validating the sequences of the check-in dataset with the ones of the car sharing database. Interestingly, the MRR is very low for the first sequences but rises with increasing k until it peaks at 0.3 for $k = 5$. For $k > 5$ the MRR almost constantly drops. The P@k is equal to 1 until $k = 15$ meaning that all the top 15 sequences of the check-in dataset were also discovered in the car sharing database. After that, there are some misses, but it is retaining still a high P@k of 0.885 at $k = 35$. For the R@k we can see a near constant growth with increasing k, but the overall values are tiny due to the large number of sequences which were extracted from the car sharing dataset.

7.3 Temporal contextualization

As described in section 6.2, we are extracting sequences for each of the four timeslots in a day separately. In the following, we show only the results on the category level of the POIs. For this experiment, we are using again the adaptive minimum support threshold, starting at a relative minimum support value of 0.1 which we lowered exponentially by multiplying it with 0.5 until we discovered at least 1000 sequences for each timeslot.

Figure 7.8 shows the sequences extracted with their absolute and relative support values, as well as their confidence. Similar to the previous experiments, the support and confidence values are overall rather low, but we can see some interesting differences between the timeslots.

The most frequent sequence is again *restaurant → restaurant*. Taking a look at the support and confidence values in the different timeslots we can see that the confidence in the timeslot from 00:00 to 06:00 is with 4.65% the lowest. This is followed by the confidences of the timeslots 06:00 - 12:00 and 18:00 - 24:00 with values of 9.24% and 9.89%, respectively. With 13.43% the highest confidence was achieved in the timeslot from 12:00 - 18:00. One possible interpretation of this result could be that people might take a car to eat lunch in a restaurant although they have another restaurant in their neighborhood.

Another interesting sequence is given by *restaurant → clothes*. We can see that the relative support value for the timeslot from 00:00 to 06:00 is with 1.09% very low, which aligns with our expectations, since shops are usually closed at these times. Later in the day the support as well as the confidence increase peaking in the afternoon at 5.20% and 6.64%, respectively. This is reasonable because most people shop for clothes in the afternoon. After that, the values decrease again for the last timeslot 18:00 - 24:00.

Interestingly, the sequence *restaurant → fast_food, restaurant* has slightly higher support in the different timeslots than the reversed sequence *fast_food, restaurant → restaurant*. However, the latter has significantly higher confidence values in each of the timeslots. One might conclude again that users often take or return the vehicle near restaurants leading to numerous sequences starting at restaurants which reduces the confidence of the sequence *restaurant → fast_food, restaurant*.

In addition, we have sorted the discovered sequences according to the sum of the confidence in the different timeslots. The results are shown in Figure 7.9. It can be seen that the sequence *restaurant → restaurant* has also the highest confidence values.

Comparing the two sequences *clothes, restaurant → restaurant*, and *clothes, dentist → restaurant* we can see different phenomena for the different times the day. While the confidence of the former is higher in the timeslot from 12:00 to 18:00, it is lower for the other three timeslots. Thus in the afternoon, users starting from an area with at least one clothing store and dentist are less likely to end their trip near a restaurant than users starting from an area with at least one clothing

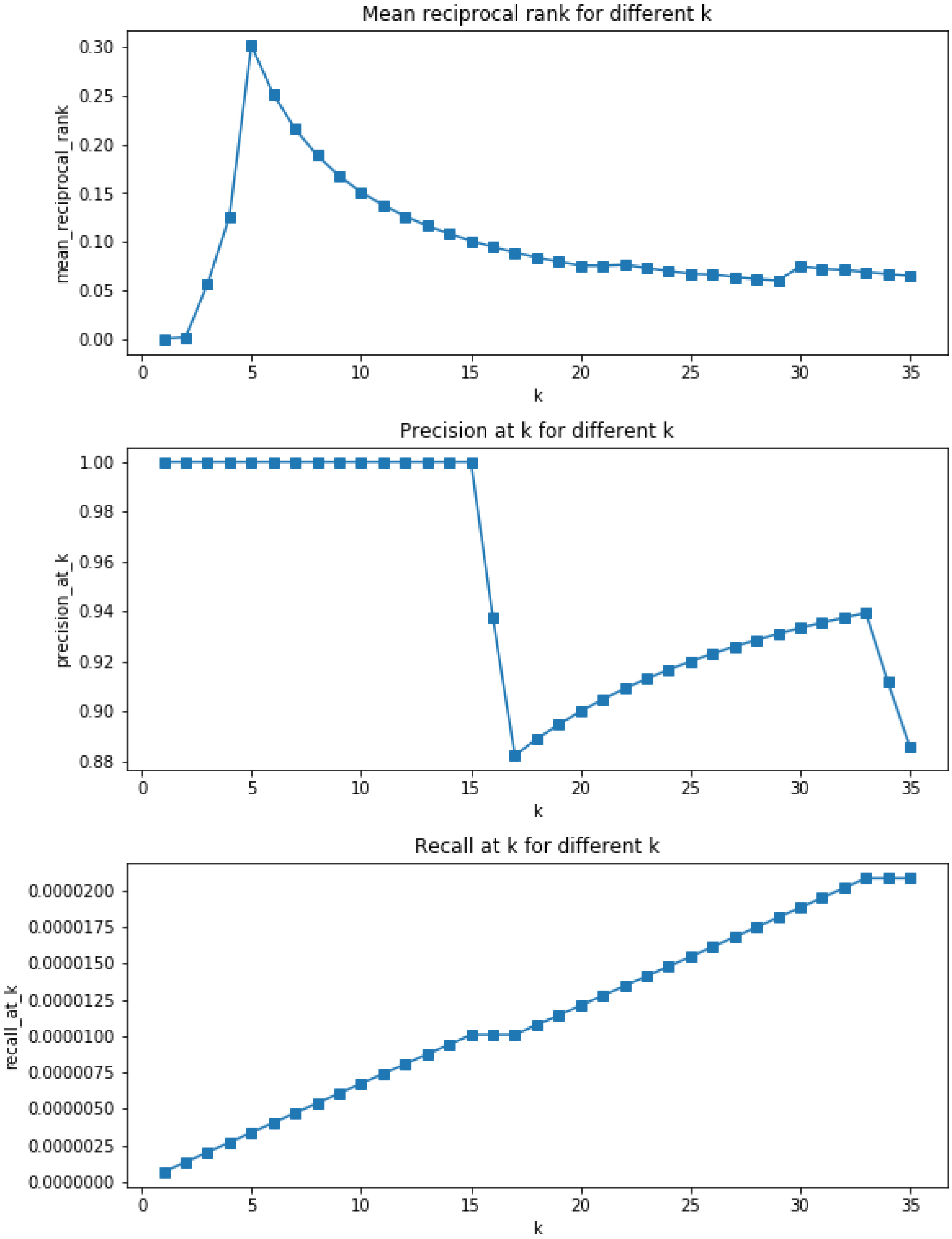
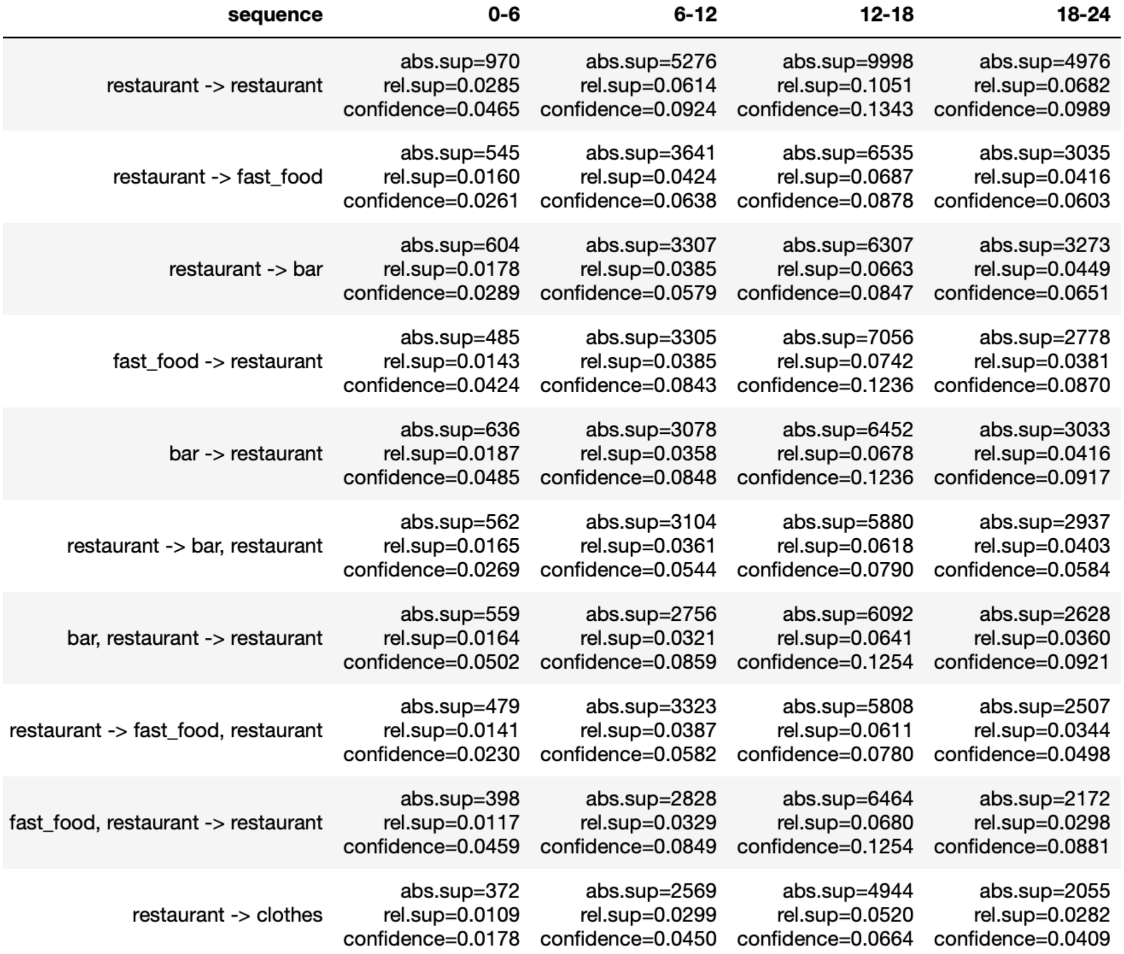


Figure 7.7. The mean reciprocal rank, the precision at k, and the recall at k for the extracted sequences from the check-in dataset in comparison with the sequences from the car sharing dataset, for $k \in \{1, 2, ..., 35\}$.



sequence	0-6	6-12	12-18	18-24

restaurant -> restaurant	abs.sup=970
	rel.sup=0.0285
	confidence=0.0465
		abs.sup=5276
		rel.sup=0.0614
		confidence=0.0924
			abs.sup=9998
			rel.sup=0.1051
			confidence=0.1343
				abs.sup=4976
				rel.sup=0.0682
				confidence=0.0989

restaurant -> fast_food	abs.sup=545
	rel.sup=0.0160
	confidence=0.0261
		abs.sup=3641
		rel.sup=0.0424
		confidence=0.0638
			abs.sup=6535
			rel.sup=0.0687
			confidence=0.0878
				abs.sup=3035
				rel.sup=0.0416
				confidence=0.0603

restaurant -> bar	abs.sup=604
	rel.sup=0.0178
	confidence=0.0289
		abs.sup=3307
		rel.sup=0.0385
		confidence=0.0579
			abs.sup=6307
			rel.sup=0.0663
			confidence=0.0847
				abs.sup=3273
				rel.sup=0.0449
				confidence=0.0651

fast_food -> restaurant	abs.sup=485
	rel.sup=0.0143
	confidence=0.0424
		abs.sup=3305
		rel.sup=0.0385
		confidence=0.0843
			abs.sup=7056
			rel.sup=0.0742
			confidence=0.1236
				abs.sup=2778
				rel.sup=0.0381
				confidence=0.0870

bar -> restaurant	abs.sup=636
	rel.sup=0.0187
	confidence=0.0485
		abs.sup=3078
		rel.sup=0.0358
		confidence=0.0848
			abs.sup=6452
			rel.sup=0.0678
			confidence=0.1236
				abs.sup=3033
				rel.sup=0.0416
				confidence=0.0917

restaurant -> bar, restaurant	abs.sup=562
	rel.sup=0.0165
	confidence=0.0269
		abs.sup=3104
		rel.sup=0.0361
		confidence=0.0544
			abs.sup=5880
			rel.sup=0.0618
			confidence=0.0790
				abs.sup=2937
				rel.sup=0.0403
				confidence=0.0584

bar, restaurant -> restaurant	abs.sup=559
	rel.sup=0.0164
	confidence=0.0502
		abs.sup=2756
		rel.sup=0.0321
		confidence=0.0859
			abs.sup=6092
			rel.sup=0.0641
			confidence=0.1254
				abs.sup=2628
				rel.sup=0.0360
				confidence=0.0921

restaurant -> fast_food, restaurant	abs.sup=479
	rel.sup=0.0141
	confidence=0.0230
		abs.sup=3323
		rel.sup=0.0387
		confidence=0.0582
			abs.sup=5808
			rel.sup=0.0611
			confidence=0.0780
				abs.sup=2507
				rel.sup=0.0344
				confidence=0.0498

fast_food, restaurant -> restaurant	abs.sup=398
	rel.sup=0.0117
	confidence=0.0459
		abs.sup=2828
		rel.sup=0.0329
		confidence=0.0849
			abs.sup=6464
			rel.sup=0.0680
			confidence=0.1254
				abs.sup=2172
				rel.sup=0.0298
				confidence=0.0881

restaurant -> clothes	abs.sup=372
	rel.sup=0.0109
	confidence=0.0178
		abs.sup=2569
		rel.sup=0.0299
		confidence=0.0450
			abs.sup=4944
			rel.sup=0.0520
			confidence=0.0664
				abs.sup=2055
				rel.sup=0.0282
				confidence=0.0409

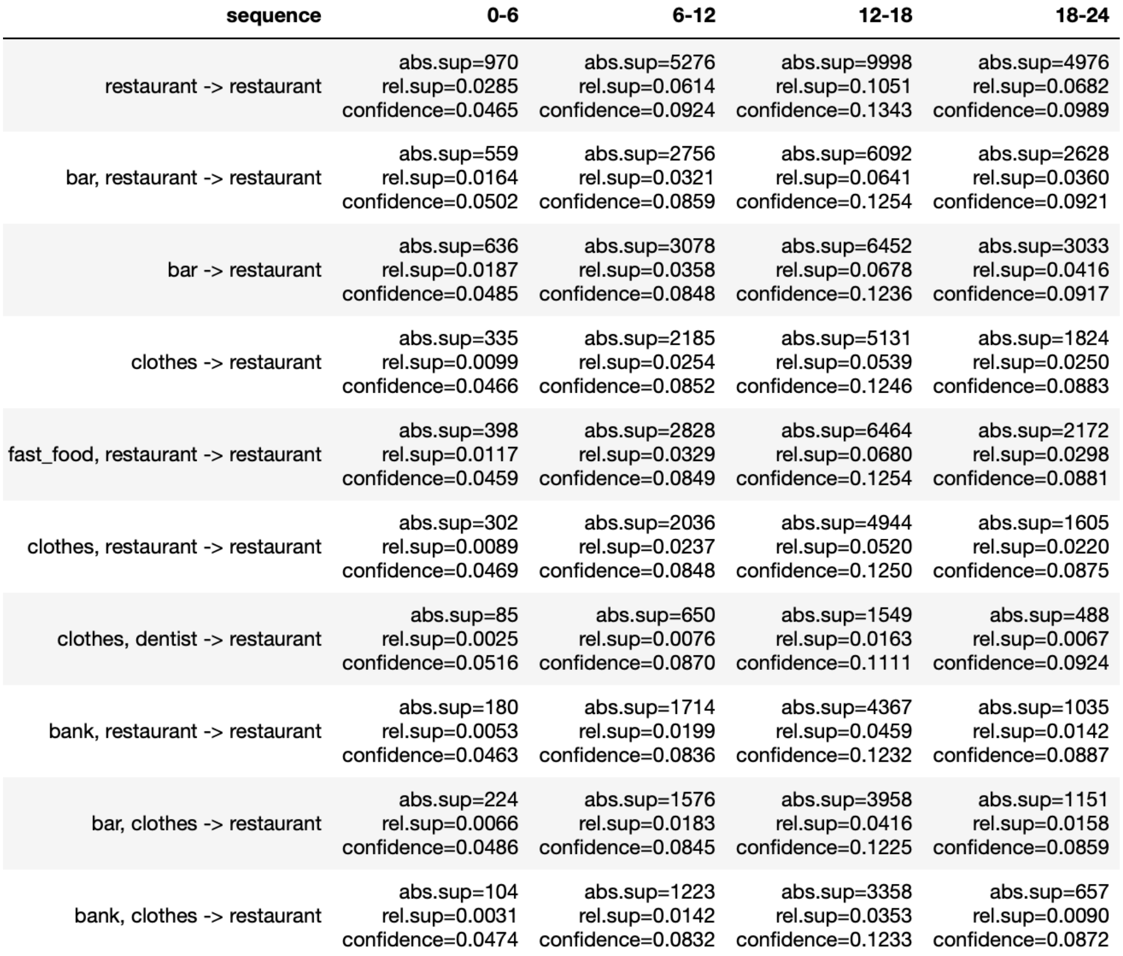
Figure 7.8. The top ten discovered sequences of Portland according to the sum of the support of the sequences in each timeslot by applying the temporal contextualization. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

store and restaurant. In contrast in the morning and night, the users behave the other way around.

We can also observe that the last three sequences in Figure 7.9, namely *bank, restaurant* → *restaurant*, *bar, clothes* → *restaurant* and *bank, clothes* → *restaurant*, all have very similar confidence values in all three timeslots. However, the support values show higher differences. Especially for the timeslots 00:00 - 06:00 and 18:00 - 24:00, the sequence *bar, clothes* → *restaurant* has twice as many occurrences in the former timeslot and nearly double in the latter than the sequence *bank, clothes* → *restaurant*.

The most frequently discovered sequences in Turin for each timeslot can be seen in Figure 7.10. There are large differences in the support and confidence values of each sequence. For instance, the sequence *restaurant* → *bank* has a relative support of only 13.67% in the timeslot from 00:00 to 06:00, 16.18% in 06:00-12:00, 20.95% in 12:00-18:00, while peaking in the evening from 18:00 to 24:00 with a value of 25.21%.

In contrast to Portland, the highest support values are found in the last timeslot from 18:00 to 24:00. Hence, these results suggest that the users in Turin tend to use car sharing most commonly



sequence	0-6	6-12	12-18	18-24

restaurant -> restaurant	abs.sup=970
	rel.sup=0.0285
	confidence=0.0465
		abs.sup=5276
		rel.sup=0.0614
		confidence=0.0924
			abs.sup=9998
			rel.sup=0.1051
			confidence=0.1343
				abs.sup=4976
				rel.sup=0.0682
				confidence=0.0989

bar, restaurant -> restaurant	abs.sup=559
	rel.sup=0.0164
	confidence=0.0502
		abs.sup=2756
		rel.sup=0.0321
		confidence=0.0859
			abs.sup=6092
			rel.sup=0.0641
			confidence=0.1254
				abs.sup=2628
				rel.sup=0.0360
				confidence=0.0921

bar -> restaurant	abs.sup=636
	rel.sup=0.0187
	confidence=0.0485
		abs.sup=3078
		rel.sup=0.0358
		confidence=0.0848
			abs.sup=6452
			rel.sup=0.0678
			confidence=0.1236
				abs.sup=3033
				rel.sup=0.0416
				confidence=0.0917

clothes -> restaurant	abs.sup=335
	rel.sup=0.0099
	confidence=0.0466
		abs.sup=2185
		rel.sup=0.0254
		confidence=0.0852
			abs.sup=5131
			rel.sup=0.0539
			confidence=0.1246
				abs.sup=1824
				rel.sup=0.0250
				confidence=0.0883

fast_food, restaurant -> restaurant	abs.sup=398
	rel.sup=0.0117
	confidence=0.0459
		abs.sup=2828
		rel.sup=0.0329
		confidence=0.0849
			abs.sup=6464
			rel.sup=0.0680
			confidence=0.1254
				abs.sup=2172
				rel.sup=0.0298
				confidence=0.0881

clothes, restaurant -> restaurant	abs.sup=302
	rel.sup=0.0089
	confidence=0.0469
		abs.sup=2036
		rel.sup=0.0237
		confidence=0.0848
			abs.sup=4944
			rel.sup=0.0520
			confidence=0.1250
				abs.sup=1605
				rel.sup=0.0220
				confidence=0.0875

clothes, dentist -> restaurant	abs.sup=85
	rel.sup=0.0025
	confidence=0.0516
		abs.sup=650
		rel.sup=0.0076
		confidence=0.0870
			abs.sup=1549
			rel.sup=0.0163
			confidence=0.1111
				abs.sup=488
				rel.sup=0.0067
				confidence=0.0924

bank, restaurant -> restaurant	abs.sup=180
	rel.sup=0.0053
	confidence=0.0463
		abs.sup=1714
		rel.sup=0.0199
		confidence=0.0836
			abs.sup=4367
			rel.sup=0.0459
			confidence=0.1232
				abs.sup=1035
				rel.sup=0.0142
				confidence=0.0887

bar, clothes -> restaurant	abs.sup=224
	rel.sup=0.0066
	confidence=0.0486
		abs.sup=1576
		rel.sup=0.0183
		confidence=0.0845
			abs.sup=3958
			rel.sup=0.0416
			confidence=0.1225
				abs.sup=1151
				rel.sup=0.0158
				confidence=0.0859

bank, clothes -> restaurant	abs.sup=104
	rel.sup=0.0031
	confidence=0.0474
		abs.sup=1223
		rel.sup=0.0142
		confidence=0.0832
			abs.sup=3358
			rel.sup=0.0353
			confidence=0.1233
				abs.sup=657
				rel.sup=0.0090
				confidence=0.0872

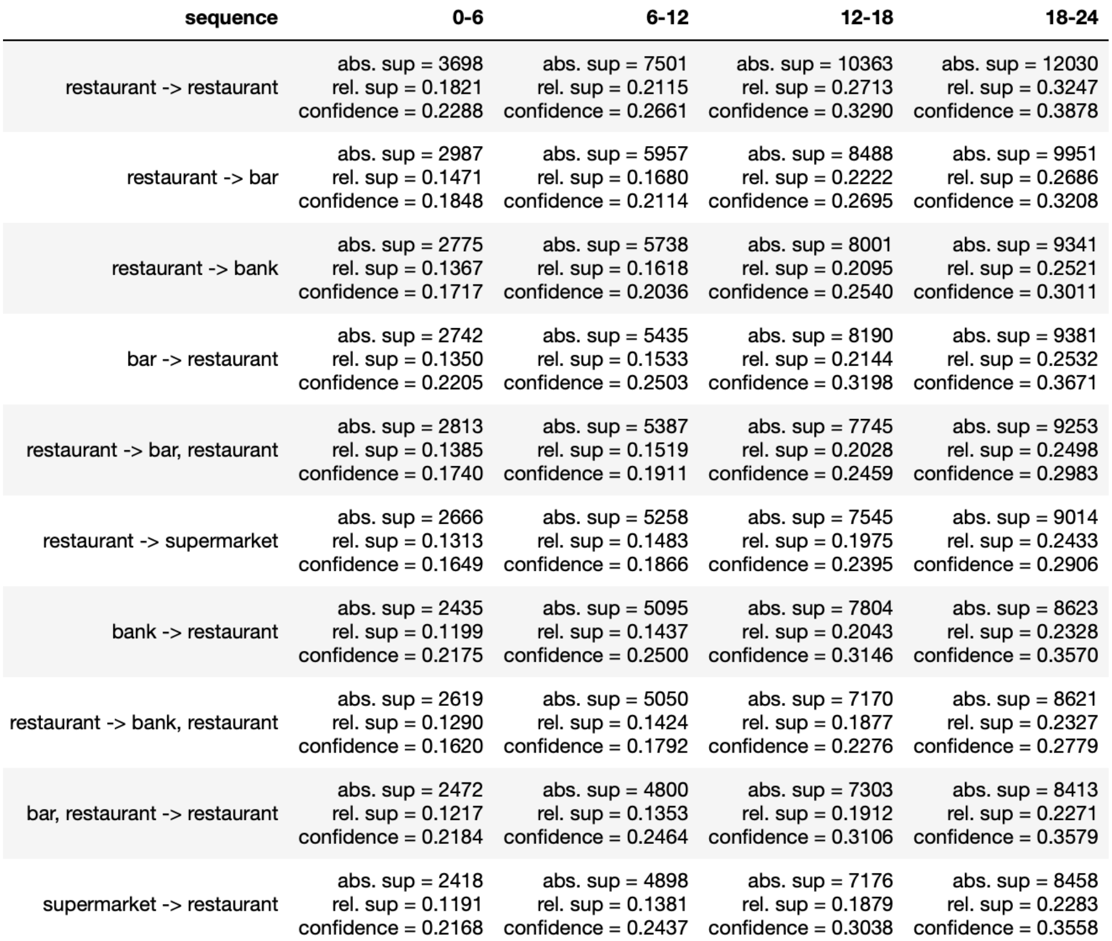
Figure 7.9. The top ten discovered sequences of Portland according to the sum of the confidence of the sequences in each timeslot by applying the temporal contextualization. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

in the evening, while in Portland we can identify higher support values in the afternoon.

Ranking the sequences of Turin according to the sum of the confidences reveals several new interesting sequences. For example, the sequence *pub, restaurant* → *bar* → *restaurant* has a relative confidence of 23.04% in the early morning. This value increases to 24.32 in the timeslot 06:00-12:00 and to 29.06 from 12:00 to 18:00. In the last timeslot, it achieves the maximum of 34.91%. This means that it is more likely for users to drive to a restaurant in the evening after having been near a pub and restaurant, followed by a bar.

7.4 Spatial contextualization

Similar to the temporal contextualization, for the spatial contextualization we run cSPADE for each of the areas of interest (city center, north, and east for Portland and city center, north, south and west for Turin), as described in section 6.2. Using the same adaptive minimum support threshold, we discovered at least 1000 sequences for each area.



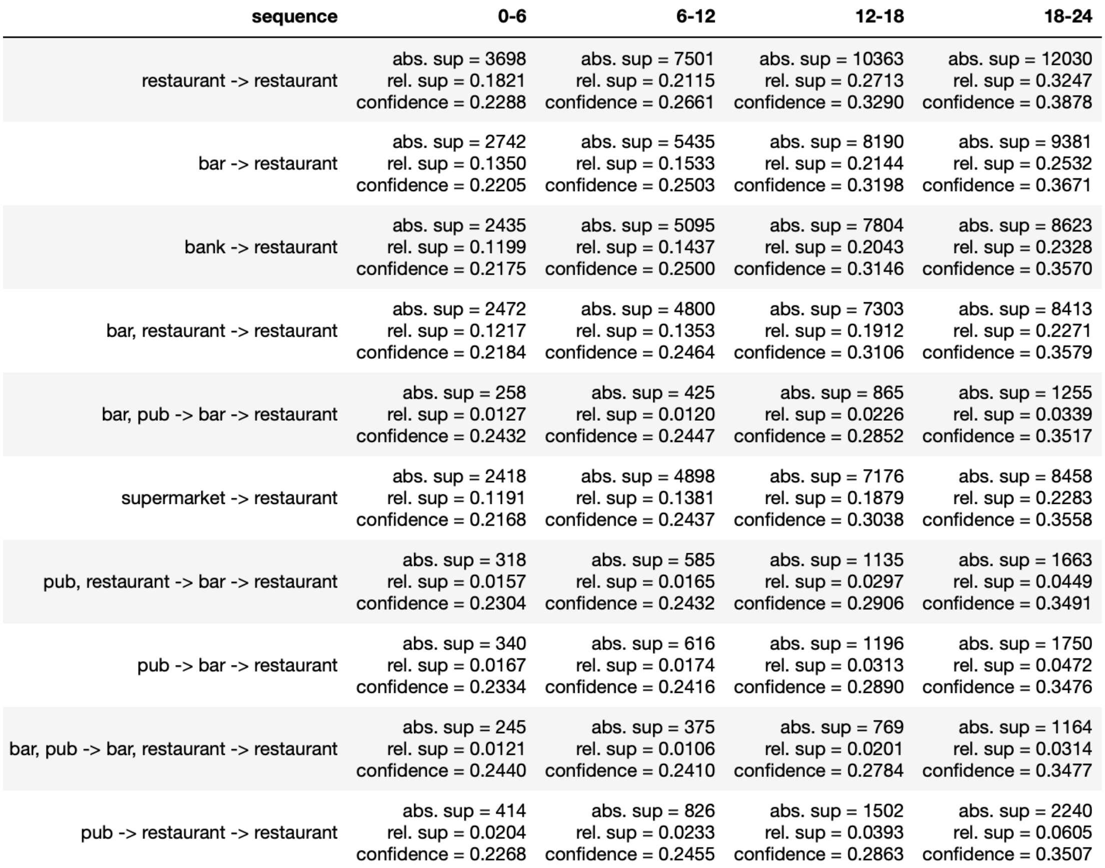
sequence	0-6	6-12	12-18	18-24
restaurant -> restaurant	abs. sup = 3698 rel. sup = 0.1821 confidence = 0.2288	abs. sup = 7501 rel. sup = 0.2115 confidence = 0.2661	abs. sup = 10363 rel. sup = 0.2713 confidence = 0.3290	abs. sup = 12030 rel. sup = 0.3247 confidence = 0.3878
restaurant -> bar	abs. sup = 2987 rel. sup = 0.1471 confidence = 0.1848	abs. sup = 5957 rel. sup = 0.1680 confidence = 0.2114	abs. sup = 8488 rel. sup = 0.2222 confidence = 0.2695	abs. sup = 9951 rel. sup = 0.2686 confidence = 0.3208
restaurant -> bank	abs. sup = 2775 rel. sup = 0.1367 confidence = 0.1717	abs. sup = 5738 rel. sup = 0.1618 confidence = 0.2036	abs. sup = 8001 rel. sup = 0.2095 confidence = 0.2540	abs. sup = 9341 rel. sup = 0.2521 confidence = 0.3011
bar -> restaurant	abs. sup = 2742 rel. sup = 0.1350 confidence = 0.2205	abs. sup = 5435 rel. sup = 0.1533 confidence = 0.2503	abs. sup = 8190 rel. sup = 0.2144 confidence = 0.3198	abs. sup = 9381 rel. sup = 0.2532 confidence = 0.3671
restaurant -> bar, restaurant	abs. sup = 2813 rel. sup = 0.1385 confidence = 0.1740	abs. sup = 5387 rel. sup = 0.1519 confidence = 0.1911	abs. sup = 7745 rel. sup = 0.2028 confidence = 0.2459	abs. sup = 9253 rel. sup = 0.2498 confidence = 0.2983
restaurant -> supermarket	abs. sup = 2666 rel. sup = 0.1313 confidence = 0.1649	abs. sup = 5258 rel. sup = 0.1483 confidence = 0.1866	abs. sup = 7545 rel. sup = 0.1975 confidence = 0.2395	abs. sup = 9014 rel. sup = 0.2433 confidence = 0.2906
bank -> restaurant	abs. sup = 2435 rel. sup = 0.1199 confidence = 0.2175	abs. sup = 5095 rel. sup = 0.1437 confidence = 0.2500	abs. sup = 7804 rel. sup = 0.2043 confidence = 0.3146	abs. sup = 8623 rel. sup = 0.2328 confidence = 0.3570
restaurant -> bank, restaurant	abs. sup = 2619 rel. sup = 0.1290 confidence = 0.1620	abs. sup = 5050 rel. sup = 0.1424 confidence = 0.1792	abs. sup = 7170 rel. sup = 0.1877 confidence = 0.2276	abs. sup = 8621 rel. sup = 0.2327 confidence = 0.2779
bar, restaurant -> restaurant	abs. sup = 2472 rel. sup = 0.1217 confidence = 0.2184	abs. sup = 4800 rel. sup = 0.1353 confidence = 0.2464	abs. sup = 7303 rel. sup = 0.1912 confidence = 0.3106	abs. sup = 8413 rel. sup = 0.2271 confidence = 0.3579
supermarket -> restaurant	abs. sup = 2418 rel. sup = 0.1191 confidence = 0.2168	abs. sup = 4898 rel. sup = 0.1381 confidence = 0.2437	abs. sup = 7176 rel. sup = 0.1879 confidence = 0.3038	abs. sup = 8458 rel. sup = 0.2283 confidence = 0.3558

Figure 7.10. The top ten discovered sequences of Turin according to the sum of the support of the sequences in each timeslot by applying the temporal contextualization. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

The ten sequences with the highest sum of support in each area can be seen in Figure 7.12. It is evident that many of those sequences also appear in the top ones of the temporal contextualization. Again, the sequence *restaurant* → *restaurant* has the highest support, which especially in the center is with 11.03% very high, but in the north and east area significantly lower with 1.71% and 6.14% respectively.

Considering the north area by itself, we can see that, after starting a trip in the neighborhood of a restaurant the user is more likely to end it in the neighborhood of a fast food venue than at a bar or a clothing shop. In numbers, the sequence *restaurant* → *fast_food* has a confidence of 1.94% but the ones of *restaurant* → *bar* and *restaurant* → *clothes* are only 1.36% and 0.91% respectively. In the east area, however, the sequence *restaurant* → *fast_food* has with 3.78% a lower confidence value than the 4.64% of *restaurant* → *bar*. The confidence of *restaurant* → *clothes* is with 2.16% significantly lower. In contrast to the other areas, in the city center, all three sequences have very similar confidence values which differ only by a maximum of 0.31%.

Another interesting phenomenon can be observed in the sequences *bar, restaurant* → *restaurant* and *restaurant* → *bar, restaurant*. While the absolute support value of the former is 237 lower than



sequence	0-6	6-12	12-18	18-24
restaurant -> restaurant	abs. sup = 3698 rel. sup = 0.1821 confidence = 0.2288	abs. sup = 7501 rel. sup = 0.2115 confidence = 0.2661	abs. sup = 10363 rel. sup = 0.2713 confidence = 0.3290	abs. sup = 12030 rel. sup = 0.3247 confidence = 0.3878
bar -> restaurant	abs. sup = 2742 rel. sup = 0.1350 confidence = 0.2205	abs. sup = 5435 rel. sup = 0.1533 confidence = 0.2503	abs. sup = 8190 rel. sup = 0.2144 confidence = 0.3198	abs. sup = 9381 rel. sup = 0.2532 confidence = 0.3671
bank -> restaurant	abs. sup = 2435 rel. sup = 0.1199 confidence = 0.2175	abs. sup = 5095 rel. sup = 0.1437 confidence = 0.2500	abs. sup = 7804 rel. sup = 0.2043 confidence = 0.3146	abs. sup = 8623 rel. sup = 0.2328 confidence = 0.3570
bar, restaurant -> restaurant	abs. sup = 2472 rel. sup = 0.1217 confidence = 0.2184	abs. sup = 4800 rel. sup = 0.1353 confidence = 0.2464	abs. sup = 7303 rel. sup = 0.1912 confidence = 0.3106	abs. sup = 8413 rel. sup = 0.2271 confidence = 0.3579
bar, pub -> bar -> restaurant	abs. sup = 258 rel. sup = 0.0127 confidence = 0.2432	abs. sup = 425 rel. sup = 0.0120 confidence = 0.2447	abs. sup = 865 rel. sup = 0.0226 confidence = 0.2852	abs. sup = 1255 rel. sup = 0.0339 confidence = 0.3517
supermarket -> restaurant	abs. sup = 2418 rel. sup = 0.1191 confidence = 0.2168	abs. sup = 4898 rel. sup = 0.1381 confidence = 0.2437	abs. sup = 7176 rel. sup = 0.1879 confidence = 0.3038	abs. sup = 8458 rel. sup = 0.2283 confidence = 0.3558
pub, restaurant -> bar -> restaurant	abs. sup = 318 rel. sup = 0.0157 confidence = 0.2304	abs. sup = 585 rel. sup = 0.0165 confidence = 0.2432	abs. sup = 1135 rel. sup = 0.0297 confidence = 0.2906	abs. sup = 1663 rel. sup = 0.0449 confidence = 0.3491
pub -> bar -> restaurant	abs. sup = 340 rel. sup = 0.0167 confidence = 0.2334	abs. sup = 616 rel. sup = 0.0174 confidence = 0.2416	abs. sup = 1196 rel. sup = 0.0313 confidence = 0.2890	abs. sup = 1750 rel. sup = 0.0472 confidence = 0.3476
bar, pub -> bar, restaurant -> restaurant	abs. sup = 245 rel. sup = 0.0121 confidence = 0.2440	abs. sup = 375 rel. sup = 0.0106 confidence = 0.2410	abs. sup = 769 rel. sup = 0.0201 confidence = 0.2784	abs. sup = 1164 rel. sup = 0.0314 confidence = 0.3477
pub -> restaurant -> restaurant	abs. sup = 414 rel. sup = 0.0204 confidence = 0.2268	abs. sup = 826 rel. sup = 0.0233 confidence = 0.2455	abs. sup = 1502 rel. sup = 0.0393 confidence = 0.2863	abs. sup = 2240 rel. sup = 0.0605 confidence = 0.3507

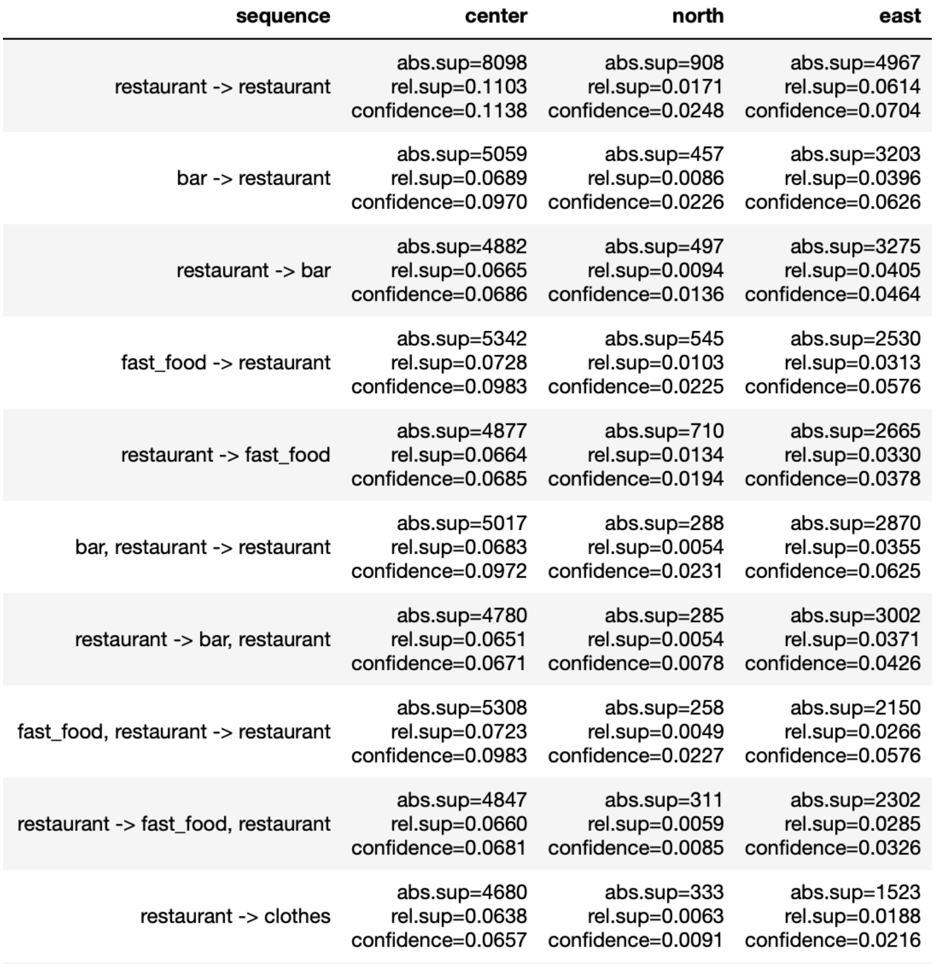
Figure 7.11. The top ten discovered sequences of Turin according to the sum of the confidence of the sequences in each timeslot by applying the temporal contextualization. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

the latter's in the city center, we can see an opposite behavior for the east area, where the former sequence has 132 occurrences more than the latter one. In the north area, however, the absolute support of both sequences is with 288 and 285 similar. This example illustrates, that there is some impact of the area on the usage of car sharing vehicles.

Figure 7.13 shows the top ten sequences sorted by the sum of the confidences of each area. Again, the sequence *restaurant* → *restaurant* has the highest confidence values.

Considering the sequences *bank, clothes* → *restaurant*, *clothes* → *restaurant*, and *bank* → *restaurant* for the city center, we can see that the first one, which is more specific, has with 9.32% slightly lower confidence than the latter ones, which have 9.88% and 9.60%. This is surprising as such that one might think that when having a more specific itemset at the beginning of the sequence, one might have a higher confidence value for that sequence, but in this case users are more likely to end their trip at a restaurant when they start in an area with a clothing shop or a bank compared to when he is starting at a point with both, a bank and a clothing shop in the neighborhood. In contrast, this is not the case in the east area. There we have a confidence of 6.57% for *bank, clothes* → *restaurant*, while having just 5.63% and 5.58% for *clothes* → *restaurant* and *bank* → *restaurant*, respectively.

In Figure 7.14, we can see the discovered sequences of the Turin dataset for each region, sorted



sequence	center	north	east

restaurant -> restaurant	abs.sup=8098
	rel.sup=0.1103
	confidence=0.1138
		abs.sup=908
		rel.sup=0.0171
		confidence=0.0248
			abs.sup=4967
			rel.sup=0.0614
			confidence=0.0704

bar -> restaurant	abs.sup=5059
	rel.sup=0.0689
	confidence=0.0970
		abs.sup=457
		rel.sup=0.0086
		confidence=0.0226
			abs.sup=3203
			rel.sup=0.0396
			confidence=0.0626

restaurant -> bar	abs.sup=4882
	rel.sup=0.0665
	confidence=0.0686
		abs.sup=497
		rel.sup=0.0094
		confidence=0.0136
			abs.sup=3275
			rel.sup=0.0405
			confidence=0.0464

fast_food -> restaurant	abs.sup=5342
	rel.sup=0.0728
	confidence=0.0983
		abs.sup=545
		rel.sup=0.0103
		confidence=0.0225
			abs.sup=2530
			rel.sup=0.0313
			confidence=0.0576

restaurant -> fast_food	abs.sup=4877
	rel.sup=0.0664
	confidence=0.0685
		abs.sup=710
		rel.sup=0.0134
		confidence=0.0194
			abs.sup=2665
			rel.sup=0.0330
			confidence=0.0378

bar, restaurant -> restaurant	abs.sup=5017
	rel.sup=0.0683
	confidence=0.0972
		abs.sup=288
		rel.sup=0.0054
		confidence=0.0231
			abs.sup=2870
			rel.sup=0.0355
			confidence=0.0625

restaurant -> bar, restaurant	abs.sup=4780
	rel.sup=0.0651
	confidence=0.0671
		abs.sup=285
		rel.sup=0.0054
		confidence=0.0078
			abs.sup=3002
			rel.sup=0.0371
			confidence=0.0426

fast_food, restaurant -> restaurant	abs.sup=5308
	rel.sup=0.0723
	confidence=0.0983
		abs.sup=258
		rel.sup=0.0049
		confidence=0.0227
			abs.sup=2150
			rel.sup=0.0266
			confidence=0.0576

restaurant -> fast_food, restaurant	abs.sup=4847
	rel.sup=0.0660
	confidence=0.0681
		abs.sup=311
		rel.sup=0.0059
		confidence=0.0085
			abs.sup=2302
			rel.sup=0.0285
			confidence=0.0326

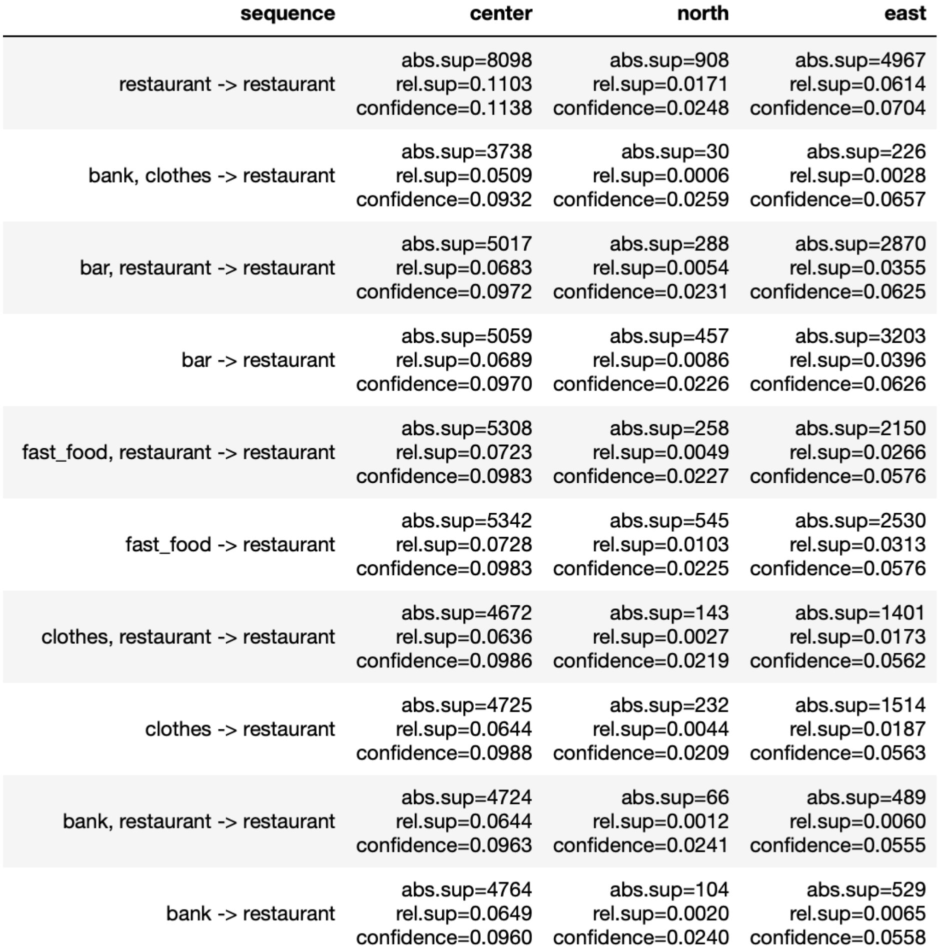
restaurant -> clothes	abs.sup=4680
	rel.sup=0.0638
	confidence=0.0657
		abs.sup=333
		rel.sup=0.0063
		confidence=0.0091
			abs.sup=1523
			rel.sup=0.0188
			confidence=0.0216

Figure 7.12. The top ten discovered sequences of Portland according to the sum of the support of the sequences in each one of the selected areas by applying the spatial contextualization. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

by the sum of the support. A pair of sequences illustrating the differences between the areas is *restaurant* → *bar* and *restaurant* → *bank*. In the city center, the north and the west area, the former sequence has higher support values than the latter one. In the south area however, the relative support for *restaurant* → *bar* is with 5.89% smaller than 7.62% for *restaurant* → *bank*.

The sequences for each area sorted by the sum of the confidence are shown in Figure 7.15. For instance, the sequence *restaurant* → *fast_food* → *restaurant* has with 19.27% the highest confidence in the west area, while the second highest confidence of 15.18% is in the city center. In contrast, the relative support is with 1.79% in the west area less than in the city center where it is 2.36%. The confidence and support values in the other two areas are significantly lower.

Furthermore, we investigated the distribution of the relative support values of the 100 most frequent sequences of Portland, for each timeslot and area, as well as for the general dataset. Figure 7.16 shows that the distribution is left-skewed in all eight cases. If we would take even more sequences into account, the distribution will skew even stronger, since the added sequences will have all lower relative support values as the lowest one which is integrated into the figure. Overall,



sequence	center	north	east

restaurant -> restaurant	abs.sup=8098
	rel.sup=0.1103
	confidence=0.1138
		abs.sup=908
		rel.sup=0.0171
		confidence=0.0248
			abs.sup=4967
			rel.sup=0.0614
			confidence=0.0704

bank, clothes -> restaurant	abs.sup=3738
	rel.sup=0.0509
	confidence=0.0932
		abs.sup=30
		rel.sup=0.0006
		confidence=0.0259
			abs.sup=226
			rel.sup=0.0028
			confidence=0.0657

bar, restaurant -> restaurant	abs.sup=5017
	rel.sup=0.0683
	confidence=0.0972
		abs.sup=288
		rel.sup=0.0054
		confidence=0.0231
			abs.sup=2870
			rel.sup=0.0355
			confidence=0.0625

bar -> restaurant	abs.sup=5059
	rel.sup=0.0689
	confidence=0.0970
		abs.sup=457
		rel.sup=0.0086
		confidence=0.0226
			abs.sup=3203
			rel.sup=0.0396
			confidence=0.0626

fast_food, restaurant -> restaurant	abs.sup=5308
	rel.sup=0.0723
	confidence=0.0983
		abs.sup=258
		rel.sup=0.0049
		confidence=0.0227
			abs.sup=2150
			rel.sup=0.0266
			confidence=0.0576

fast_food -> restaurant	abs.sup=5342
	rel.sup=0.0728
	confidence=0.0983
		abs.sup=545
		rel.sup=0.0103
		confidence=0.0225
			abs.sup=2530
			rel.sup=0.0313
			confidence=0.0576

clothes, restaurant -> restaurant	abs.sup=4672
	rel.sup=0.0636
	confidence=0.0986
		abs.sup=143
		rel.sup=0.0027
		confidence=0.0219
			abs.sup=1401
			rel.sup=0.0173
			confidence=0.0562

clothes -> restaurant	abs.sup=4725
	rel.sup=0.0644
	confidence=0.0988
		abs.sup=232
		rel.sup=0.0044
		confidence=0.0209
			abs.sup=1514
			rel.sup=0.0187
			confidence=0.0563

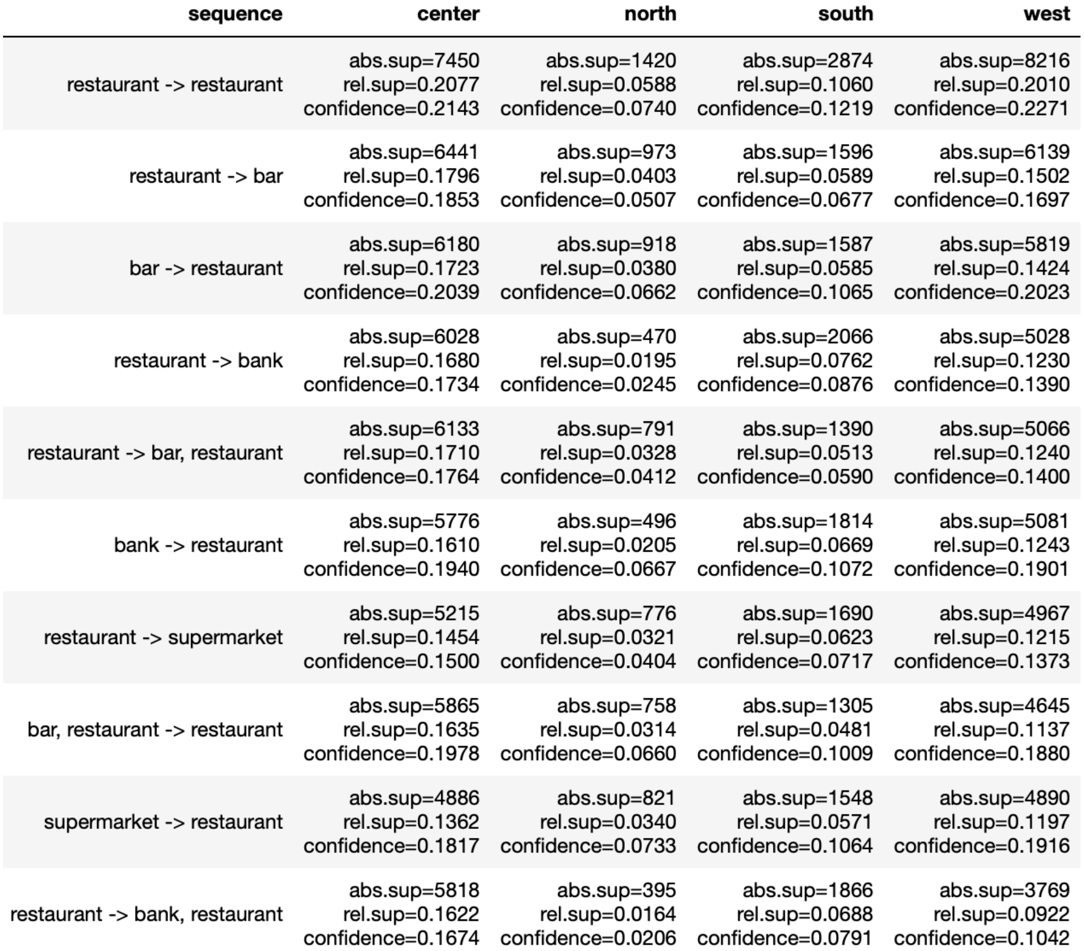
bank, restaurant -> restaurant	abs.sup=4724
	rel.sup=0.0644
	confidence=0.0963
		abs.sup=66
		rel.sup=0.0012
		confidence=0.0241
			abs.sup=489
			rel.sup=0.0060
			confidence=0.0555

bank -> restaurant	abs.sup=4764
	rel.sup=0.0649
	confidence=0.0960
		abs.sup=104
		rel.sup=0.0020
		confidence=0.0240
			abs.sup=529
			rel.sup=0.0065
			confidence=0.0558

Figure 7.13. The top ten discovered sequences of Portland according to the sum of the confidence of the sequences in each one of the selected areas by applying the spatial contextualization. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

the distributions behave very similarly for all of the eight cases with slight differences for example when comparing the distribution of the city center to the one of the north area.

In contrast to the distributions of the relative support, we can see larger differences in the distributions of the confidence values. Figure 7.17 depicts the distribution of the 100 most frequent sequences for each timeslot and area, as well as for the general dataset. In the temporal contextualization, the distribution for the timeslots 06:00 - 12:00 and 18:00 - 24:00 are very similar. However, the timeslot 12:00 - 18:00 is based more around the middle and the end of the range, whereas the other two have also a peak in the third and fourth bin, respectively. The distribution of 00:00 - 06:00 appears a little bit more widespread over its range, but the range is with a maximum value of just above 0.05 smaller than the others. An even greater difference can be seen when looking at the distributions of each area. While the values of the city center are heavily based around the fourth bin, the distribution of the north area resembles a U-shaped distribution and the east area has a distribution with most of its values in the first half of its range but also with a high local peak in the second part of the range.



sequence	center	north	south	west

restaurant -> restaurant	abs.sup=7450
	rel.sup=0.2077
	confidence=0.2143
		abs.sup=1420
		rel.sup=0.0588
		confidence=0.0740
			abs.sup=2874
			rel.sup=0.1060
			confidence=0.1219
				abs.sup=8216
				rel.sup=0.2010
				confidence=0.2271

restaurant -> bar	abs.sup=6441
	rel.sup=0.1796
	confidence=0.1853
		abs.sup=973
		rel.sup=0.0403
		confidence=0.0507
			abs.sup=1596
			rel.sup=0.0589
			confidence=0.0677
				abs.sup=6139
				rel.sup=0.1502
				confidence=0.1697

bar -> restaurant	abs.sup=6180
	rel.sup=0.1723
	confidence=0.2039
		abs.sup=918
		rel.sup=0.0380
		confidence=0.0662
			abs.sup=1587
			rel.sup=0.0585
			confidence=0.1065
				abs.sup=5819
				rel.sup=0.1424
				confidence=0.2023

restaurant -> bank	abs.sup=6028
	rel.sup=0.1680
	confidence=0.1734
		abs.sup=470
		rel.sup=0.0195
		confidence=0.0245
			abs.sup=2066
			rel.sup=0.0762
			confidence=0.0876
				abs.sup=5028
				rel.sup=0.1230
				confidence=0.1390

restaurant -> bar, restaurant	abs.sup=6133
	rel.sup=0.1710
	confidence=0.1764
		abs.sup=791
		rel.sup=0.0328
		confidence=0.0412
			abs.sup=1390
			rel.sup=0.0513
			confidence=0.0590
				abs.sup=5066
				rel.sup=0.1240
				confidence=0.1400

bank -> restaurant	abs.sup=5776
	rel.sup=0.1610
	confidence=0.1940
		abs.sup=496
		rel.sup=0.0205
		confidence=0.0667
			abs.sup=1814
			rel.sup=0.0669
			confidence=0.1072
				abs.sup=5081
				rel.sup=0.1243
				confidence=0.1901

restaurant -> supermarket	abs.sup=5215
	rel.sup=0.1454
	confidence=0.1500
		abs.sup=776
		rel.sup=0.0321
		confidence=0.0404
			abs.sup=1690
			rel.sup=0.0623
			confidence=0.0717
				abs.sup=4967
				rel.sup=0.1215
				confidence=0.1373

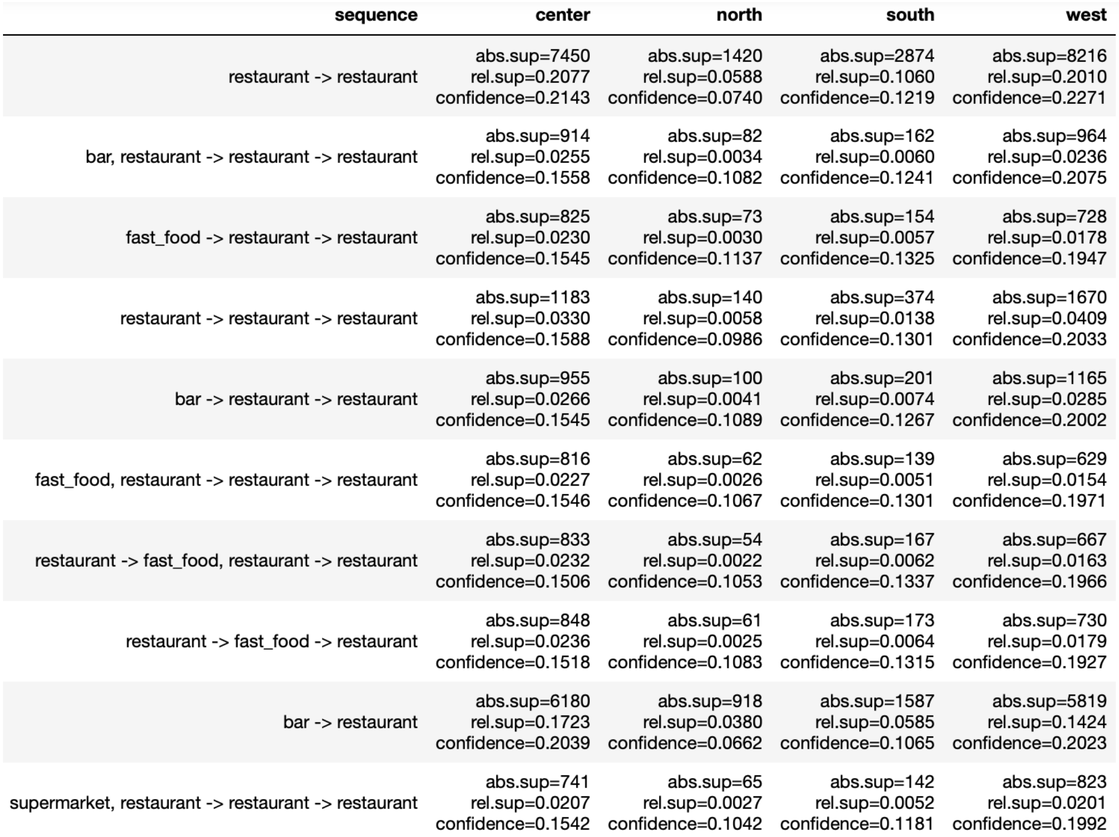
bar, restaurant -> restaurant	abs.sup=5865
	rel.sup=0.1635
	confidence=0.1978
		abs.sup=758
		rel.sup=0.0314
		confidence=0.0660
			abs.sup=1305
			rel.sup=0.0481
			confidence=0.1009
				abs.sup=4645
				rel.sup=0.1137
				confidence=0.1880

supermarket -> restaurant	abs.sup=4886
	rel.sup=0.1362
	confidence=0.1817
		abs.sup=821
		rel.sup=0.0340
		confidence=0.0733
			abs.sup=1548
			rel.sup=0.0571
			confidence=0.1064
				abs.sup=4890
				rel.sup=0.1197
				confidence=0.1916

restaurant -> bank, restaurant	abs.sup=5818
	rel.sup=0.1622
	confidence=0.1674
		abs.sup=395
		rel.sup=0.0164
		confidence=0.0206
			abs.sup=1866
			rel.sup=0.0688
			confidence=0.0791
				abs.sup=3769
				rel.sup=0.0922
				confidence=0.1042

Figure 7.14. The top ten discovered sequences of Turin according to the sum of the support of the sequences in each one of the selected areas by applying the spatial contextualization. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

The distributions of the relative support and confidence values of the discovered sequences in the Turin dataset are similar to the ones of Portland.



sequence	center	north	south	west

restaurant -> restaurant	abs.sup=7450
	rel.sup=0.2077
	confidence=0.2143
		abs.sup=1420
		rel.sup=0.0588
		confidence=0.0740
			abs.sup=2874
			rel.sup=0.1060
			confidence=0.1219
				abs.sup=8216
				rel.sup=0.2010
				confidence=0.2271

bar, restaurant -> restaurant -> restaurant	abs.sup=914
	rel.sup=0.0255
	confidence=0.1558
		abs.sup=82
		rel.sup=0.0034
		confidence=0.1082
			abs.sup=162
			rel.sup=0.0060
			confidence=0.1241
				abs.sup=964
				rel.sup=0.0236
				confidence=0.2075

fast_food -> restaurant -> restaurant	abs.sup=825
	rel.sup=0.0230
	confidence=0.1545
		abs.sup=73
		rel.sup=0.0030
		confidence=0.1137
			abs.sup=154
			rel.sup=0.0057
			confidence=0.1325
				abs.sup=728
				rel.sup=0.0178
				confidence=0.1947

restaurant -> restaurant -> restaurant	abs.sup=1183
	rel.sup=0.0330
	confidence=0.1588
		abs.sup=140
		rel.sup=0.0058
		confidence=0.0986
			abs.sup=374
			rel.sup=0.0138
			confidence=0.1301
				abs.sup=1670
				rel.sup=0.0409
				confidence=0.2033

bar -> restaurant -> restaurant	abs.sup=955
	rel.sup=0.0266
	confidence=0.1545
		abs.sup=100
		rel.sup=0.0041
		confidence=0.1089
			abs.sup=201
			rel.sup=0.0074
			confidence=0.1267
				abs.sup=1165
				rel.sup=0.0285
				confidence=0.2002

fast_food, restaurant -> restaurant -> restaurant	abs.sup=816
	rel.sup=0.0227
	confidence=0.1546
		abs.sup=62
		rel.sup=0.0026
		confidence=0.1067
			abs.sup=139
			rel.sup=0.0051
			confidence=0.1301
				abs.sup=629
				rel.sup=0.0154
				confidence=0.1971

restaurant -> fast_food, restaurant -> restaurant	abs.sup=833
	rel.sup=0.0232
	confidence=0.1506
		abs.sup=54
		rel.sup=0.0022
		confidence=0.1053
			abs.sup=167
			rel.sup=0.0062
			confidence=0.1337
				abs.sup=667
				rel.sup=0.0163
				confidence=0.1966

restaurant -> fast_food -> restaurant	abs.sup=848
	rel.sup=0.0236
	confidence=0.1518
		abs.sup=61
		rel.sup=0.0025
		confidence=0.1083
			abs.sup=173
			rel.sup=0.0064
			confidence=0.1315
				abs.sup=730
				rel.sup=0.0179
				confidence=0.1927

bar -> restaurant	abs.sup=6180
	rel.sup=0.1723
	confidence=0.2039
		abs.sup=918
		rel.sup=0.0380
		confidence=0.0662
			abs.sup=1587
			rel.sup=0.0585
			confidence=0.1065
				abs.sup=5819
				rel.sup=0.1424
				confidence=0.2023

supermarket, restaurant -> restaurant -> restaurant	abs.sup=741
	rel.sup=0.0207
	confidence=0.1542
		abs.sup=65
		rel.sup=0.0027
		confidence=0.1042
			abs.sup=142
			rel.sup=0.0052
			confidence=0.1181
				abs.sup=823
				rel.sup=0.0201
				confidence=0.1992

Figure 7.15. The top ten discovered sequences of Turin according to the sum of the confidence of the sequences in each one of the selected areas by applying the spatial contextualization. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

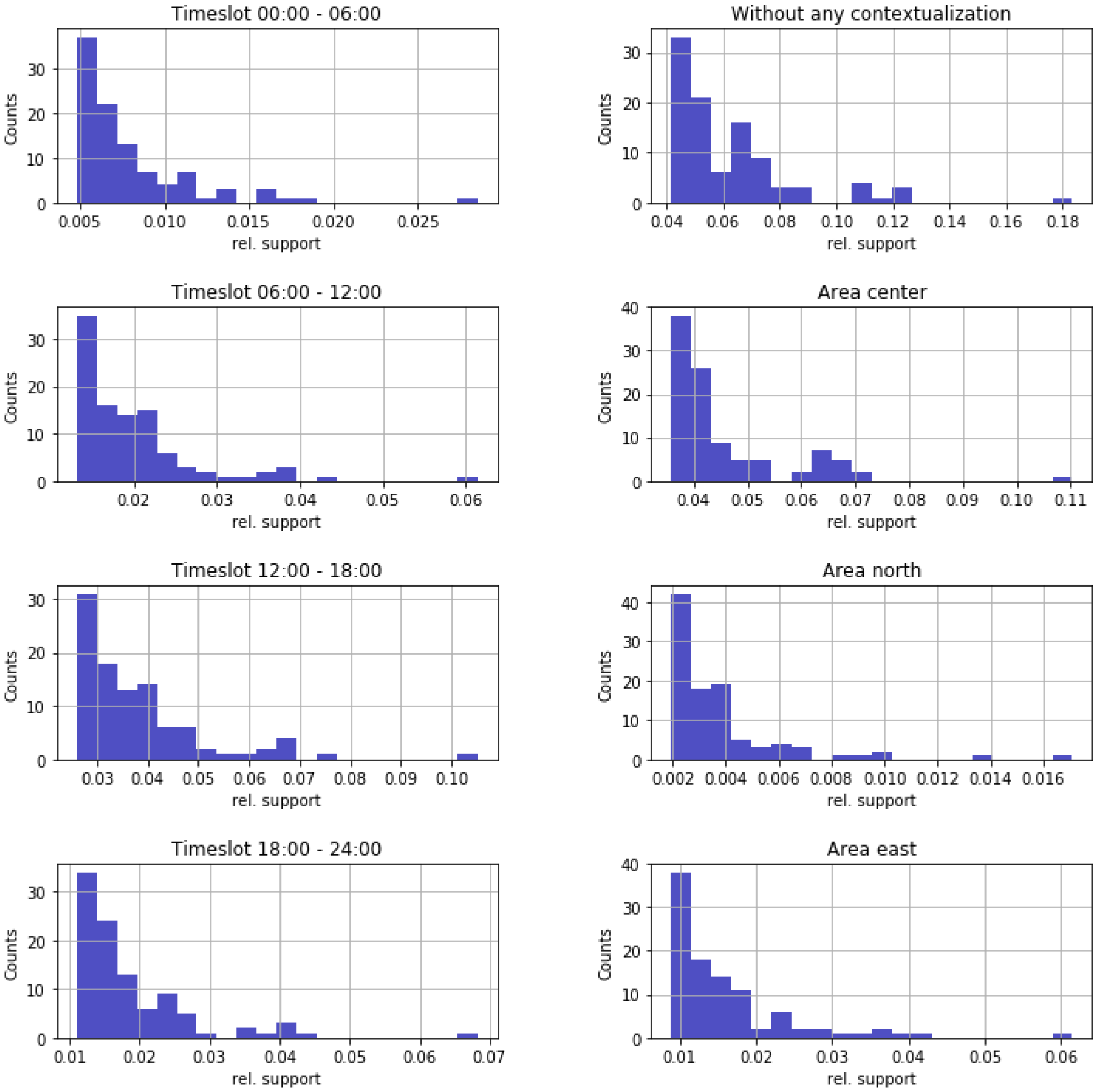


Figure 7.16. Distribution of the relative support values of the 100 sequences of Portland with the highest relative support for each temporal and spatial contextualization, as well as without any contextualization.

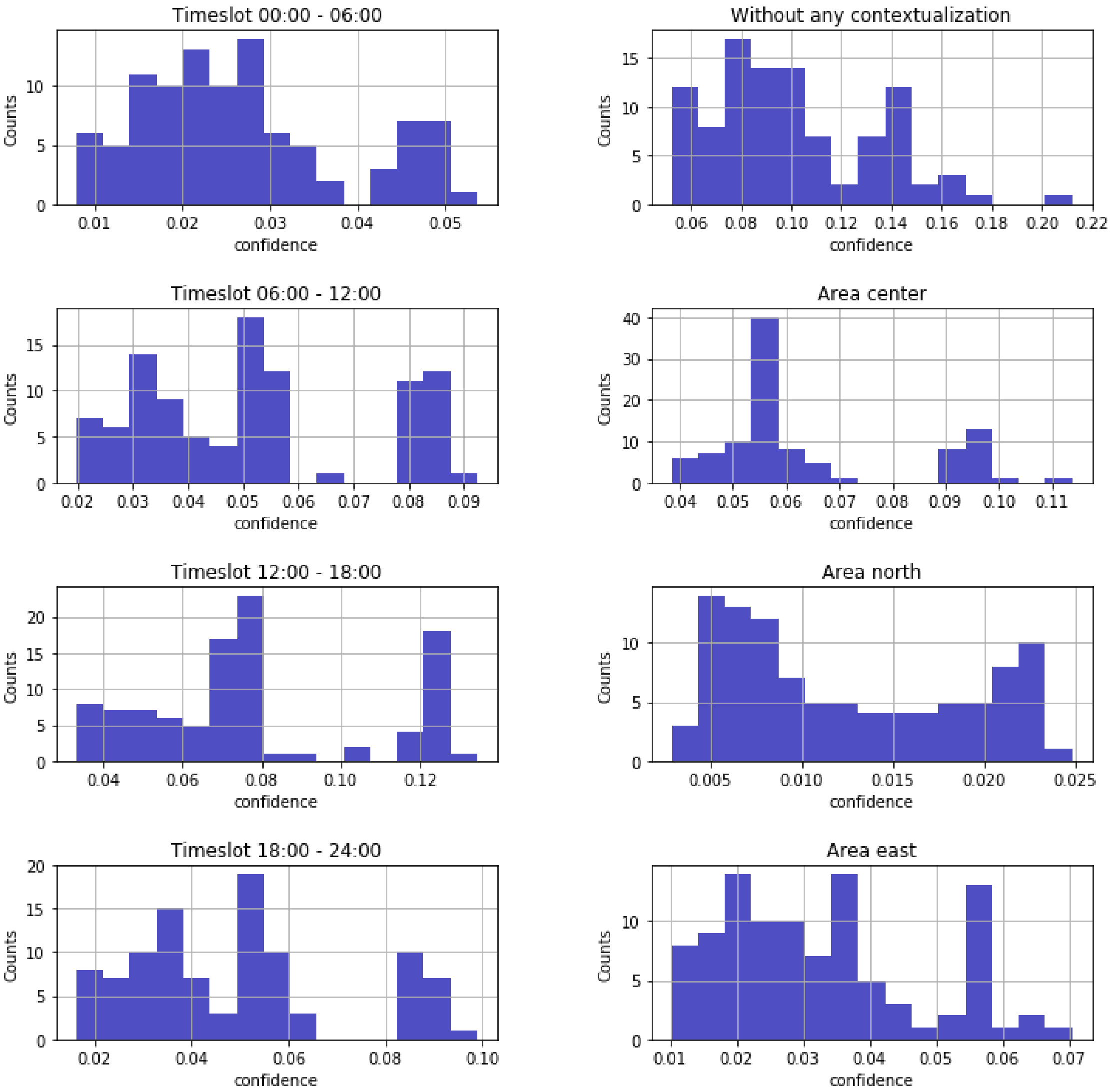


Figure 7.17. Distribution of the relative support values of the 100 most frequent sequences of Portland for each temporal and spatial contextualization, as well as without any contextualization.

Chapter 8

Conclusion and Outlook

8.1 Conclusion

In the first part of this thesis, we analyzed the possibility of predicting the availability of cars around a POI in the future. This can be interesting for car sharing users who want to know whether there will be a car available in the near future or for car sharing operators who want to relocate cars to regions with a predicted shortage.

In our experiments on the data of free-floating car sharing systems from Portland, Seattle, and Turin, we trained a random forest model for each POI on different feature sets and evaluated its performance by calculating the F1-score of the predictions on a hold-out test set which consisted of 50% of the database for each city. This works best using ten lags into the past, which contain the last five hours of recorded values.

Furthermore, we observed that the time of the day strongly impacts the performance of the random forest models, although this inner-day difference is diverse for each city.

Although the time series have a rather high stationarity, our results show that we can clearly outperform the baseline of predicting only the last recorded value. For ten randomly chosen shops in Seattle and a horizon of two hours we achieved an average F1-score of 79.46% (baseline: 71.74%). In Turin however, the difference between the random forest model and the baseline is with 80.80% and 76.59% slightly lower. Our classifier scored a F1-score of 72.01% in Portland, where the baseline scored only 70.22%.

With a horizon of 2.5 hours, the proposed classifier outperformed the baseline by 6.08 percentage points for Seattle, and by 5.8 and 2.65 percentage points for Turin and Portland, respectively.

In the last experiment of this part, we have shown that for all cities but Seattle, the random forest models also work for amenities and can therefore be trained on the data for multiple different types of POIs and used for their predictions.

In contrast to previous research, this work is among the first to study the impact of different variables purely based on on automatically collected car sharing data.

While sequence pattern mining has been applied to multiple domains, such as biology and network data, to the author's best knowledge, this is the first work proposing to use sequence pattern mining for extracting behavioral patterns of car sharing usage. We also showed the possibility of using these behavioral patterns to model general mobility patterns of citizens.

Multiple experiments were carried out on free-floating car sharing data from Portland and Turin. The car sharing data was preprocessed into sequences, each of which containing the trip information of one car on one day. Then, we applied a state-of-the-art sequence mining algorithm, called cSPADE, which allows us to set up constraints in order to reduce the extremely large search space.

Our experiments have shown that we can discover multiple sequences with respect to both specific POIs and POI categories, which have varying support and confidence values. We found several common sequences in the datasets of Turin and Portland, e.g. users traveling from a restaurant to a bar. However, a larger amount of sequences containing fast food venues was discovered in Portland compared to Turin.

Furthermore, we have compared the extracted sequences from the car sharing data with sequences discovered in a check-in data set. The latter ones have a higher certainty, because the users explicitly share that they are visiting a venue at a given time on the social network. The results show that there are matched sequences in both datasets. This shows the validity of the sequences extracted from car sharing data.

Additionally, we have applied a temporal as well as a spatial contextualization to the data. By dividing the time of the day into four six-hour timeslots, several differences in the support and confidence values were found. For instance the support of the sequence *restaurant* → *clothes* in Portland is with only 1.09% lower between 00:00 to 06:00 than later in the day, where it peaks at 6.64%. This is reasonable, because most people are shopping for clothes in the afternoon. After that, the values decrease again for the last timeslot.

The spatial contextualization also revealed interesting sequences and their differences in each region of the operator's area. For example, in the north area of Portland, a member who started his or her trip in the neighborhood of a restaurant is more likely to end it near a fast food venue than near a clothing store, while in the east area it is vice versa.

Our work has led us to the conclusion that applying sequence pattern mining on car sharing data reveals interesting sequences of car sharing usage. Furthermore, it was shown that these sequences also represent general mobility patterns and can therefore be used as an alternative to expensive studies or scarse user-based data from location-based social networks.

8.2 Outlook

There are several interesting and promising ideas to be investigated in future research.

First of all, the sequence mining algorithm can be applied to other car sharing data sets of other cities in order to gain more knowledge about the different sequences in different cities.

Apart from that, sequences with other time gaps can be discovered by changing the parameters of the cSPADE algorithm. For instance, we can extract seqeuences of longer car sharing trips with a minimum gap of 45 minutes and a maximum gap one of 120 minutes to travel. This can reveal further details about car sharing usage.

In order to apply the cSPADE algorithm with a higher itemset size or sequence length to the datasets, it is possible to expand the set of constraints of the algorithm by supporting constraints which include the POI categories. This would not only decrease the amount of work for preprocessing the dataset but might also increase the quality of the discovered sequences by not allowing uninteresting subsequences.

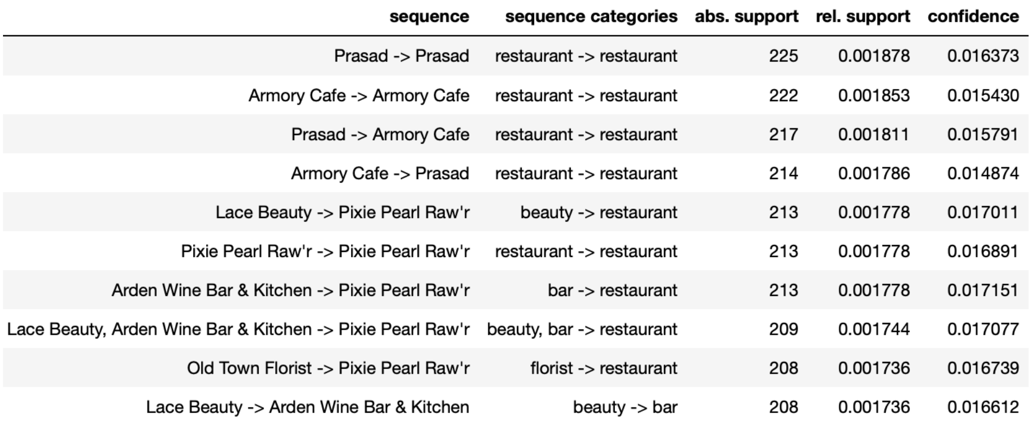
Additionally, similar to the comparison with the sequences from a check-in database in this work, more comparisons with other public movement data, such as taxi service data or car-hailing data, can be conducted. This can improve the reliability of the discovered sequences and further show that the car sharing data is a good indicator of society's behavioral patterns.

Moreover, additional approaches on how to increase the validity of a user visiting a POI when ending a car sharing trip in its neighborhood should be explored. For instance, we could do a majority voting of all POIs within a certain radius of the trip's destination to determine the POI category with higher certainty.

Furthermore, an experiment should be carried out which correlates the sequences found for different times with other time-dependent data, e.g. weather data. This approach can reveal further variables impacting when members tend to use car sharing as their transportation method.

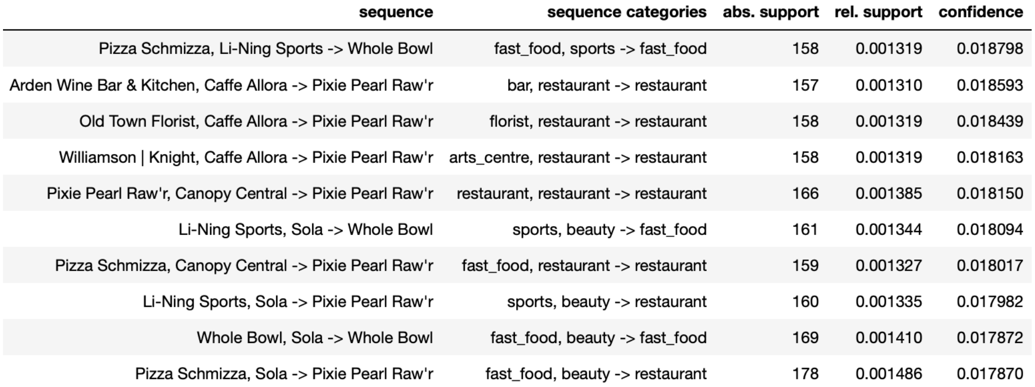
Appendix A

Discovered sequences in the car sharing dataset



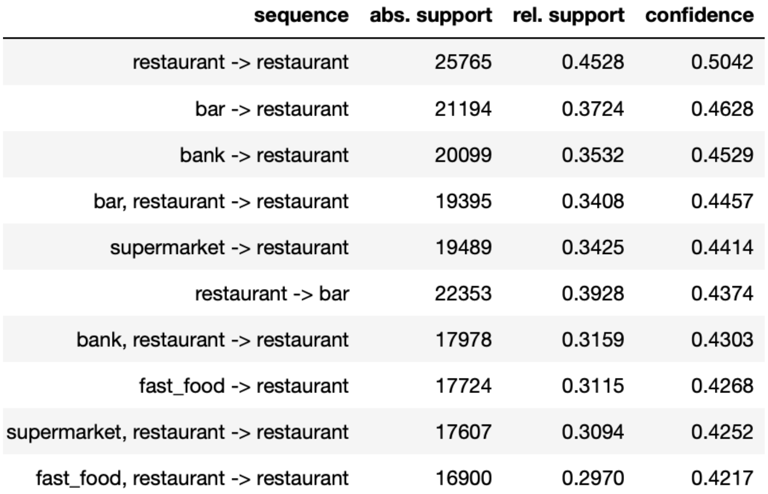
sequence	sequence categories	abs. support	rel. support	confidence
Prasad -> Prasad	restaurant -> restaurant	225	0.001878	0.016373
Armory Cafe -> Armory Cafe	restaurant -> restaurant	222	0.001853	0.015430
Prasad -> Armory Cafe	restaurant -> restaurant	217	0.001811	0.015791
Armory Cafe -> Prasad	restaurant -> restaurant	214	0.001786	0.014874
Lace Beauty -> Pixie Pearl Raw'r	beauty -> restaurant	213	0.001778	0.017011
Pixie Pearl Raw'r -> Pixie Pearl Raw'r	restaurant -> restaurant	213	0.001778	0.016891
Arden Wine Bar & Kitchen -> Pixie Pearl Raw'r	bar -> restaurant	213	0.001778	0.017151
Lace Beauty, Arden Wine Bar & Kitchen -> Pixie Pearl Raw'r	beauty, bar -> restaurant	209	0.001744	0.017077
Old Town Florist -> Pixie Pearl Raw'r	florist -> restaurant	208	0.001736	0.016739
Lace Beauty -> Arden Wine Bar & Kitchen	beauty -> bar	208	0.001736	0.016612

Figure A.1. The discovered sequences of Portland on POI level from the car sharing data, sorted by the support. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.



sequence	sequence categories	abs. support	rel. support	confidence
Pizza Schmizza, Li-Ning Sports -> Whole Bowl	fast_food, sports -> fast_food	158	0.001319	0.018798
Arden Wine Bar & Kitchen, Caffe Allora -> Pixie Pearl Raw'r	bar, restaurant -> restaurant	157	0.001310	0.018593
Old Town Florist, Caffe Allora -> Pixie Pearl Raw'r	florist, restaurant -> restaurant	158	0.001319	0.018439
Williamson | Knight, Caffe Allora -> Pixie Pearl Raw'r	arts_centre, restaurant -> restaurant	158	0.001319	0.018163
Pixie Pearl Raw'r, Canopy Central -> Pixie Pearl Raw'r	restaurant, restaurant -> restaurant	166	0.001385	0.018150
Li-Ning Sports, Sola -> Whole Bowl	sports, beauty -> fast_food	161	0.001344	0.018094
Pizza Schmizza, Canopy Central -> Pixie Pearl Raw'r	fast_food, restaurant -> restaurant	159	0.001327	0.018017
Li-Ning Sports, Sola -> Pixie Pearl Raw'r	sports, beauty -> restaurant	160	0.001335	0.017982
Whole Bowl, Sola -> Whole Bowl	fast_food, beauty -> fast_food	169	0.001410	0.017872
Pizza Schmizza, Sola -> Pixie Pearl Raw'r	fast_food, beauty -> restaurant	178	0.001486	0.017870

Figure A.2. The discovered sequences of Portland on POI level from the car sharing data, sorted by the confidence. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.



sequence	abs. support	rel. support	confidence
restaurant -> restaurant	25765	0.4528	0.5042
bar -> restaurant	21194	0.3724	0.4628
bank -> restaurant	20099	0.3532	0.4529
bar, restaurant -> restaurant	19395	0.3408	0.4457
supermarket -> restaurant	19489	0.3425	0.4414
restaurant -> bar	22353	0.3928	0.4374
bank, restaurant -> restaurant	17978	0.3159	0.4303
fast_food -> restaurant	17724	0.3115	0.4268
supermarket, restaurant -> restaurant	17607	0.3094	0.4252
fast_food, restaurant -> restaurant	16900	0.2970	0.4217

Figure A.3. The discovered sequences of Turin on category level from the the car sharing data, sorted by the confidence. Abs. sup. is the absolute support, while rel. sup. stands for the relative support of the sequence.

Bibliography

[1] Downtown Portland. https://en.wikipedia.org/wiki/Downtown_Portland,_Oregon. Accessed: 2019-06-30.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, 1995.

[3] J. Ali, R. Khan, N. Ahmad, and I. Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues(IJCSI)*, 9, 2012.

[4] H. Becker, F. Ciari, and K. Axhausen. Comparing car-sharing schemes in switzerland: User groups and usage patterns. *Transportation Research Part A: Policy and Practice*, 97:17–29, 2017.

[5] H. Becker, A. Loder, B. Schmid, and K. Axhausen. Modeling car-sharing membership as a mobility tool: A multivariate probit approach with latent variables. *Travel Behaviour and Society*, 8:26–36, 2017.

[6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. 1984.

[7] L. Caggiani, R. Camporeale, and M. Ottomanelli. A dynamic clustering method for relocation process in free-floating vehicle sharing systems. *Transportation Research Procedia*, 27:278 – 285, 2017.

[8] C. Celsor and A. Millard-Ball. Where does carsharing work?: Using geographic information systems to assess market potential. *Transportation Research Record*, 1992(1):61–69, 2007.

[9] C. Cong and C. P. Tsokos. *Theory and Applications of Decision Tree with Statistical Software*. 2009.

[10] A. Cutler, D. Cutler, and J. Stevens. *Random Forests*, volume 45, pages 157–176. 2011.

[11] J. Firnkorn and M. Müller. What will be the environmental effects of new free-floating car-sharing systems? the case of car2go in ulm. *Ecological Economics*, 70(8):1519–1528, 2011.

[12] A. Garcia Asuero, A. Sayago, and G. Gonzalez. The correlation coefficient: An overview. *Critical Reviews in Analytical Chemistry*, 36:41–59, 2006.

[13] M. Garofalakis, R. Rastogi, and K. Shim. Spirit: Sequential pattern mining with regular expression constraints. *VLDB*, 99, 01 2000.

[14] E. Goel and E. Abhilasha. Random forest: A review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 7:251–257, 2017.

[15] U. Haefeli, D. Matti, C. Schreyer, and M. Maibach. Evaluation car-sharing. Federal Department of the Environment, Transport, Energy and Communications, Bern, 2006.

[16] K. Hatonen, M. Klemettinen, H. Mannila, P. Ronkainen, and H. Toivonen. Knowledge discovery from telecommunication network alarm databases. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 115–122, 1996.

[17] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is np-complete. *Inf. Process. Lett.*, 5:15–17, 1976.

[18] J. Kang, K. Hwang, and S. Park. Finding factors that influence carsharing usage: Case study in seoul. *Sustainability*, 8:709, 2016.

[19] T. H. Kang, J. S. Yoo, and H. Y. Kim. Mining frequent contiguous sequence patterns in biological sequences. *IEEE 7th International Symposium on BioInformatics and BioEngineering*, pages 723–728, 2007.

[20] S. Le Vine, M. Lee-Gosselin, A. Sivakumar, and J. Polak. A new approach to predict the market and impacts of round-trip and point-to-point carsharing systems: Case study of london. *Transportation Research Part D: Transport and Environment*, 32:218–229, 2014.

[21] H. R. Lewis and C. H. Papadimitriou. Elements of the theory of computation. *SIGACT News*, 29:62–78, 1981.

[22] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997.

[23] C. D. Manning, P. Raghavan, and H. Schütze. Chapter 8: Evaluation in information retrieval. In *Introduction to Information Retrieval*, 2009.

[24] J. Müller and K. Bogenberger. Time series analysis of booking data of a free-floating carsharing system in berlin. *Transportation Research Procedia*, 10:345–354, 2015.

[25] J. Müller, G. Homem de Almeida Correia, and K. Bogenberger. An explanatory model approach for the spatial distribution of free-floating carsharing bookings: A case-study of german cities. *Sustainability*, 9:1290, 2017.

[26] G. B. M. of Defence (Navy) and G. B. A. M. of navigation. *Admiralty Manual of Navigation: General navigation, coastal navigation, and pilotage.* Admiralty Manual of Navigation. H.M.S.O., 1987.

[27] O. Pauly. *Random Forests for Medical Applications.* 2012.

[28] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[29] L. E. Raileanu and K. Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41:77–93, 2002.

[30] S. Schmöller and K. Bogenberger. Analyzing external factors on the spatial and temporal demand of car sharing systems. *Procedia - Social and Behavioral Sciences*, 111:8 – 17, 2014.

[31] S. Shaheen, A. Cohen, and M. Chung. North american carsharing: A ten-year retrospective. *Institute of Transportation Studies, UC Davis, Institute of Transportation Studies, Working Paper Series*, 2008.

[32] R. W. Sinnott. Virtues of the haversine. *skytel*, 68:158, 1984.

[33] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, 1996.

[34] T. Stillwater, P. Mokhtarian, and S. Shaheen. Carsharing and the built environment: A gis-based study of one u.s. operator. *Institute of Transportation Studies, UC Davis, Institute of Transportation Studies, Working Paper Series*, 2008.

[35] E. M. Voorhees. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999.

[36] S. Wagner, T. Brandt, and D. Neumann. Data analytics in free-floating carsharing: Evidence from the city of berlin. *48th Hawaii International Conference on System Sciences*, pages 897–907, 2015.

[37] S. Wagner, T. Brandt, and D. Neumann. In free float: Developing business analytics support for carsharing providers. *Omega*, 59, 2015.

[38] S. Wagner, C. Willing, T. Brandt, and D. Neumann. Data analytics for location-based services: Enabling user-based relocation of carsharing vehicles. In *ICIS*, 2015.

[39] S. Weikl and K. Bogenberger. Relocation strategies and algorithms for free-floating car sharing systems. *15th International IEEE Conference on Intelligent Transportation Systems*, pages 355–360, 2012.

[40] C. Willing, K. Klemmer, T. Brandt, and D. Neumann. Moving in time and space - location intelligence for carsharing decision support. *Decision Support Systems*, 2017.

[41] D. Yang, D. Zhang, L. Chen, and B. Qu. Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *Journal of Network and Computer Applications*, 55:170–180, 2015.

[42] D. Yang, D. Zhang, and B. Qu. Participatory cultural mapping based on collective behavior in location based social networks. *ACM Transactions on Intelligent Systems and Technology*, 2015.

[43] D. Yang, D. Zhang, and B. Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM TIST*, 7:30:1–30:23, 2016.

[44] M. J. Zaki. Efficient enumeration of frequent sequences. In *CIKM*, 1998.

[45] M. J. Zaki. Sequence mining in categorical domains: Incorporating constraints. In *CIKM*, 2000.

[46] M. J. Zaki, N. Lesh, and M. Ogihara. Planmine: Sequence mining for plan failures. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, NY, USA*, pages 369–374, 1998.