An Ensemble Learning Approach for the Kaggle Taxi Travel Time Prediction Challenge

Thomas Hoch

Software Competence Center Hagenberg GmbH
Softwarepark 21, 4232 Hagenberg, Austria
Tel.: +43-7236-3343-831
thomas.hoch@scch.at

Abstract. This paper describes the winning solution to the Taxi Trip Time Prediction Challenge run by Kaggle.com. The goal of the competition was to build a predictive framework that is able to predict the final destination and the total traveling time of taxi rides based on their (initial) partial trajectories. The available data consists of all taxi trips of 442 taxis running in the city of Porto within one year. The presented solution consists of an ensemble of expert models combined with a spatial clustering approach. The base classifiers consist of Random Forest Regressors where as the expert models for each test trip where based on a combination of gradient boosting and random forest. The paper shows how these models can be combined in order to generate accurate predictions of the remaining traveling time of a taxi.

Keywords: taxi-passenger demand, GPS data, ensemble learning, random forest regressor, gradient boosting, spatial clustering, machine learning

1 Introduction

The goal of the Taxi Trip Time Prediction Challenge run by Kaggle.com was to build a predictive framework that is able to predict the final destination and the total traveling time of a taxi. Due to change from VHF-radio dispatch system to an electronic dispatch system in Porto in recent years, most drivers don't indicate the final destination of their current trip. With a predictive framework the taxi central is able to optimize the efficiency of their electronic dispatch system.

Multiple works in literature have investigated operational dynamics of taxi services (see [2] for a survey). The goal is to use the taxi trajectories to look for common patterns, which can be used to optimize the taxi service. For example, Liu et al. used spatiotemporal patterns to explore driving behavior differences

* The research reported in this paper has been partly supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH

between top and ordinary drivers [8]. The work presented in [10] uses short-term forecast models to predict passenger demand patterns over a period of time in order to increase the profitability of the taxi industry.

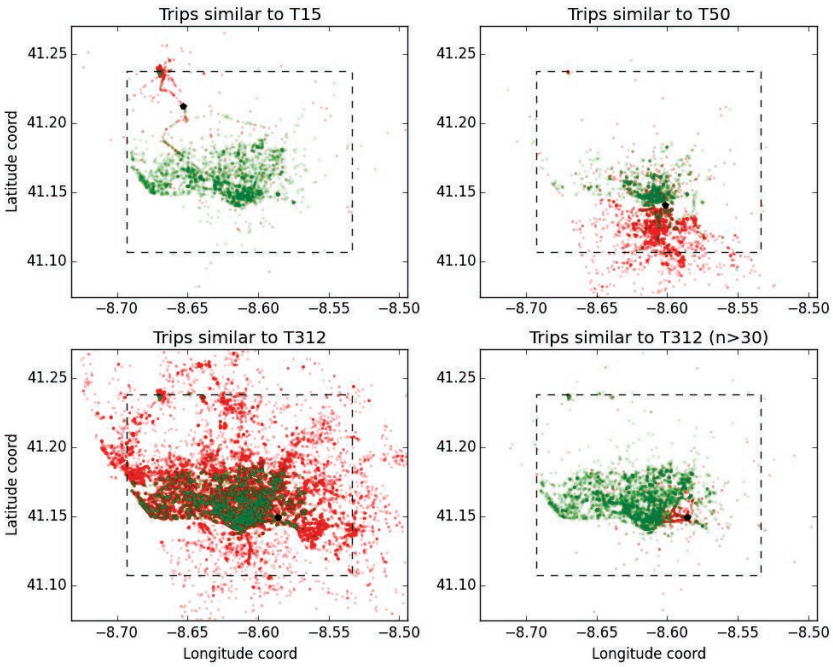
In general, the driving behavior of people is very repetitive because of daily routines and therefore the final location of a trip can be predicted most of the time with high accuracy [5]. On the other hand, taxi drivers serve a diversity of passengers, and the final destination of the taxi can be anywhere in the city. Thus, the prediction of the final destination of a taxi based on the initial trajectory is quit difficult. However, there are some patterns which most drivers obey. For example, most taxi drivers use the fastest path to the final destination. Given only a part of the trajectory, the continuation of the track is subject to a range of factors, including the current position, the type of the road the taxi is (e.g. highway or not), the current time, weather conditions, and calendar effects. Some of these factors can be inferred from the trajectory itself (e.g. speed is an indicator if the taxi is currently on a highway), others could be derived from the separately delivered meta data. Consequently, the task was to find high level features which are a good representation of a partial taxi trip.

2 Problem Statement and Main Idea

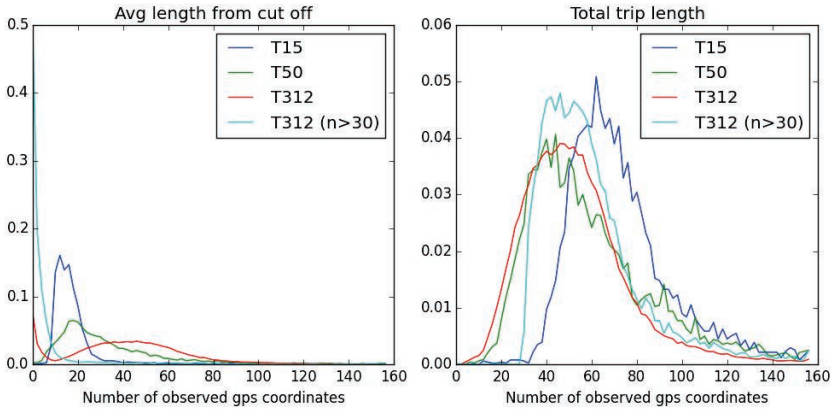
The final destination of a taxi trip and also the remaining driving time depends strongly on the last position of partial trajectory of the trip. For example, the last known position of the taxi for trip 15 of the test set is on the highway close to airport and it is very likely that the final position of the taxi will be at the airport. The top left plot of Figure 1a shows the end locations of all taxi trips (red) in the training set which cross the end position of the partial trajectory of T15 at some point in time. The plot shows that most but not all of these trips are indeed going to the airport. A precise prediction of the final destination as well as for the remaining traveling time (see Fig. 1b) is feasible.

Unfortunately, not all positions are equal predictive. For trip T50 (upper left plot in in Fig. 1), the dependency between the last known taxi position (black dot) and the final destination (red) is not as strong as for trip T15 and therefore only some broader tendency can be inferred. The left bottom plot shows the final destinations for trip T312. Taxis which cross the marked position (black) basically go to all parts of the city. A precise prediction is not feasible. Although a closer look at the data reveals, that tracks with a certain initial length (e.g. number of points > 30) show a clear spatial pattern (right bottom plot in Fig. 1). This is also true for the prediction of the remaining traveling time, as Fig. 1b shows. For trip T15 the histogram of the average length of the trip to the end position shows a clear peak, whereas for the other three trips the distribution is much broader.

From these observations the following framework is derived. It consists of a hierarchy of expert models where in the first layer an expert model for every trip in the test set is generated. On the next level a combination of some base



(a) Taxi trip end positions



(b) Travel time

Fig. 1. a) Start (green) and end (red) positions of all taxi trips in the training set, where the trip crosses the lsat known position (black) of the selected test trip. The dotted rectangle shows the main area of Porto. b) Histogram of the total trip length and the remaining trip length of the test trip shown in a).

models is generated and used when the size of the training set of the models in the first layer is too small. In particular the following were trained:

- Expert models for each test trip (e.g. trained on tracks which cross the test trip at the last known position).
- General base model: Based on a data set, where the features were extracted from all the tracks in the training set, and longer tracks were sampled more frequently than shorter ones.
- General expert models for short trips (e.g. only 1, 2 or 3 positions of the initial trajectory are known).

2.1 Methodology

On many competitions on Kaggle.com as well as in the literature it has been shown, that an ensemble of learning algorithms achieves a better performance than any single one in the ensemble [3, 12]. The framework of this paper follows the same approach and integrates different base models and track dependent models for the prediction of the remaining traveling time. As base classifiers either a random forest regression [1] or a gradient boosting regression [4] has been used. All models are trained using a 5 fold cross-validation technique. For both classifiers the implementation within the python package `scikit-learn`[11] has been used.

The ensemble prediction is modeled as follows:

$$Y_i(x_i) = w_i E_i(x_i) + \sum_{j=1}^{3} v_{ij} S_j(x_i) + u_i B(x_i) \tag{1}$$

where

- $E_i(x_i)$ is the expert prediction for the remaining traveling time or the final destination trained for the last position of the sample track x_i, respectively;
- $S_j(x_i)$ is the prediction of the general short trip expert classifier, trained on all trips in the training data set using only the j first GPS positions;
- $B(x_i)$ is the prediction of the general base model trained on sampled trips from the data set;
- w_i, v_{ij}, and u_i are the corresponding weight factors.

Ideally, the weight factors are tuned on a hold out test set with *Bayesian Optimization* as proposed in [7]. However, because of time constrains, the weight factors for the winning submission are set after carefully inspection of the cross validation plots in the following (heuristic) way:

- For all test trips with a sufficient large training set for the expert model, the prediction of the expert model was used (e.g. for the final submission to the contest $w_i = 1$ and $v_{ij}, u_i = 0$ for all trips where the number of samples in the training set was above 1000 and $w_i = 0$ otherwise).
- For all other test trips the prediction was a blend of the different base models. For the final submission the prediction was the average of all four models if the trajectory length was below 15 (e.g. $v_{ij} = 0.25, u_i = 0.25$) and otherwise the prediction of the general base model $B(x_i)$ with weight $u_i = 1$.

2.2 Data Acquisition and Preprocessing

The training dataset used for this competition is available from [9]. It consists of all the trajectories of 442 taxis running in the city of Porto within one year (from 01/07/2013 to 30/06/2014). Each taxi has an telematic system installed, which acquires the current GPS position and some additional meta data: (1) `CALL_TYPE`, which identifies the way the taxi service is demanded. (2) `ORIGIN_CALL`: Unique id to identify the caller of the service. (3) `ORIGIN_STAND`: Unique id to identify the taxi stand. (4) `TAXI_ID`: Unique id of the taxi driver. (5) `TIME_STAMP`: Start time of the trip. (6) `DAY_TYPE`: Identifier for the day type of the trip's start. (7) `MISSING_DATA`: Indicates if partial data of the trip are missing. A detailed description of the dispatching system and the data acquisition process can be found in [10]. For the time prediction task only the `LATITUDE` and `LONGITUDE` coordinates of the taxi trips and the `TIME_STAMP` attribute are used.

The training set contained a lot of very short trips as can be seen in the left plot of Figure 2, which shows a histogram of the total trip length. The high fraction of trips less than 4 does not follow the general type of the distribution, and were therefore excluded them from the analysis. Another type of error that occurred frequently was misread GPS coordinates, which increased the cumulative trip length considerable. They were excluded by cutting of very long distance trips. The threshold was set such that 0.1 percent of the longest trips were removed. The remaining trips were used for the generation of the model specific training sets.

The following features were used to describe a taxi trip in the training set. (1) `WORKING_DAY`: Derived from the time stamp attribute it indicates the week day the trip started (0-6: Mon-Sun). (2) `HOUR`: The current hour the trip started (from 0 to 23). (3) `TRIP_LENGTH`: The number of GPS readings. (4) `XS`: Latitude coordinate of the trip start location. (5) `YS`: Longitude coordinate of the trip start location. (6) `XC`: Latitude coordinate of the current taxi position (e.g. cut-off location). (7) `YC`: Longitude coordinate of the current taxi position. (8) `DIST_CC`: Harversine distance from the taxi start position to the city center. (9) `DIRECTION_CC`: Direction from the start position to the city center (in degrees). (10) `DIST_TX`: Harversine distance from the city center to the current taxi position. (11) `DIRECTION_TX`: Direction from the city center to the current position of the taxi (in degrees). (12) `CUM_DIST`: Cumulative distance of the taxi trajectory from start to current location. (13) `MED_V`: Median velocity of the taxi from start to current position. (14) `VEL`: Current velocity of the taxi. (15) `HEADING`: Heading of the car at the current position (in degrees).

The training sets of the different models were generated as follows:

- Base model: The training set contained all trips. The current position of the taxi was determined by randomly cut-off the trajectory in between (uniformly). The right plot in Figure 2) shows the distribution of the trip length up to the cut-off position (blue curve) in comparison with the trip length in the test data set (black dotted line). The number of short trips is considerable higher as in the test set, because the test set contains a snapshot of the current network status on 5 specific time points and is therefore more

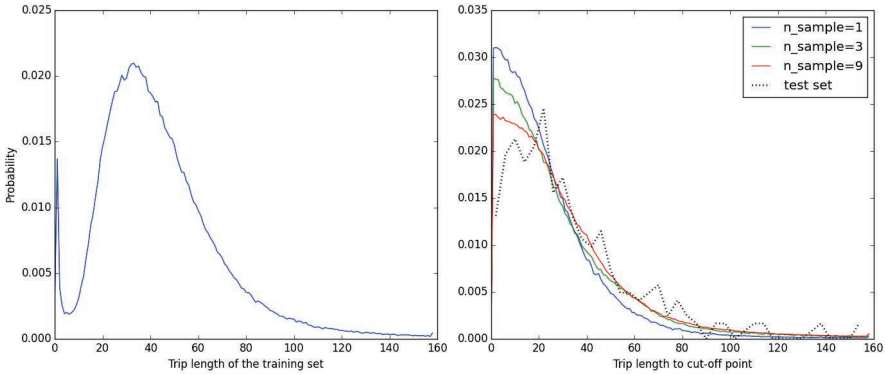


Fig. 2. Left plot shows the distribution of trip lengths. Right plot shows the trip length of the training set after sampling. More frequent sampling of longer trips decreases the number of short trips in the training set and the resulting distribution more similar to the test set distribution (black dotted line).

likely to include longer trips. To correct for the different sampling mechanism, more samples were drawn from longer trips. Figure 2 shows that by increasing the sampling frequency for longer tracks linearly with trip length, the frequency of short trips can be reduced. The resulting distribution is closer to the one of the test set (black dotted line).

– Expert models for short trips: For every expert (trip length is 1,2,3 or 4) a separate training set was build using only the first few GPS readings of all trips. For the data set which is based only on the start position (trip length = 1), some of the features can not be calculated (e.g. velocity) and were therefore excluded from the data set.
– Expert models for each test trip: Here a spatial clustering approach was used in order to select all the trips which were close to the current position of the taxi [6]. A trip in the training set was selected if there was a GPS position in the trajectory with a distance smaller than 50m to the last known taxi position. The trajectory up to this position was than used to calculate the track features.

3 Results and Discussion

3.1 Taxi Trip Time Prediction

The performance of the different models is measured using the Root Mean Squared Logarithmic Error (`RMSLE`) on the left out fold of the cross validation. The `RMSLE` is calculated as

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2} \tag{2}$$

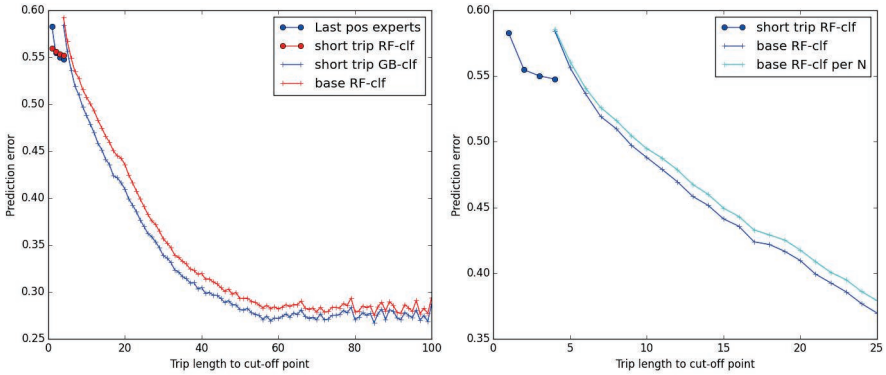


Fig. 3. Left plot shows the cross validated prediction error of the base model as a function of trip length. The Random Forest Regressor (RFR, blue) performs slightly better than the Gradient Boosting Regressor (GBR, red). The predictions of the short trip experts (blue (RFR)/red (GBR) circles) are considerably lower than the predictions of the base model. The right plot shows that models trained on a selection of the training set with fixed trip length (cyan curve) do not perform better than the actual base model trained on the whole data set (see text for further details).

where

- n is the number of trips in the left out fold of the cross validation;
- p_i is the predicted time of taxi trip i in seconds;
- a_i is the actual time of taxi trip i in seconds;
- log is the natural logarithm.

Figure 3 summarizes the result. The left plot shows the prediction error of the Random Forest Regressor (blue curve) and the Gradient Boosting Regressor (red curve) as a function of the initial trip length for the base model. The error decreases fast to values below 0.3 because of the logarithmic transformation of the predicted travel times in the error function. The prediction error is especially high for very short trips, because very little information can be gained from the first few points. Because of the sampling strategy used to generate the training set for the base model, the number of short tracks is rather slow. It makes sense to leverage the information of the whole data set by training specialist models utilizing only the beginning of all the tracks. The prediction error of these models is shown with blue (RFR) and red (GBR) circles in the plot. The prediction error is considerably lower, which indicates that the classifier is able to generalize more strongly from the increased size of the data set.

Because of the superior performance of the short trip models the question arises whether the same performance could be achieved with the base model, if for all the trips with the same length in the data set a single model would have been trained. The cyan curve in the top right plot of Fig.3 shows the predictions of these single random forest regressors. Although the training size

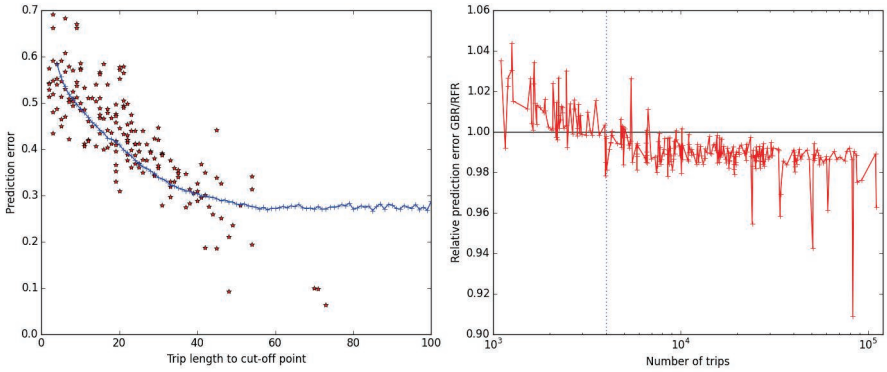


Fig. 4. The blue curve in the left plot shows the cross validated prediction error of the base model (RFR) as a function of trip length. The red points show the prediction error of a simple average prediction, if the training set is constrained on the last position of the test trip (see text for further details). The right plot shows increase of the GBR prediction error relative to the RFR error for the models trained on the last known taxi position. For a training set size above 4000 (blue line) the GBR is slightly better than the RFR.

of these individual RFR was around 10^6, the RFR trained on all the data (1.6 million) performed slightly better. Thus, a higher number of data points in the training set allows for a better approximation of the posterior distributions and thus for a more precise prediction.

Interestingly, a different behavior was found for the experts based on the last observed position. Since the last position of the taxi can be very indicative for the final destination (see left top plot in 1a), it makes sense to investigate this relationship further. A simple base line estimator for the total traveling time would be the current traveling time plus t_i the expected mean of the remaining traveling time of all trips with the same length.

$$\hat{y}_i = t_i + \frac{1}{n}\sum_{j=1}^{n}(y_j - t_j) \tag{3}$$

Figure 4 shows in the left plot the prediction error of the base line estimators (red) against current trip length. For comparison, the prediction error of the base model (RFR) is added in blue. The Fig. shows nicely, that for many trips the simple baseline estimator outperforms the base model. Unfortunately, because of the restriction on the last position and the trip length, the data set size is greatly reduced. Therefore the expert models were trained only on the last position constrain.

For the base models the random forest regressor performed slightly better then the gradient boosting regressor for all but the single position case (see Fig.3). However, for the models trained on the end position of the trajectory the

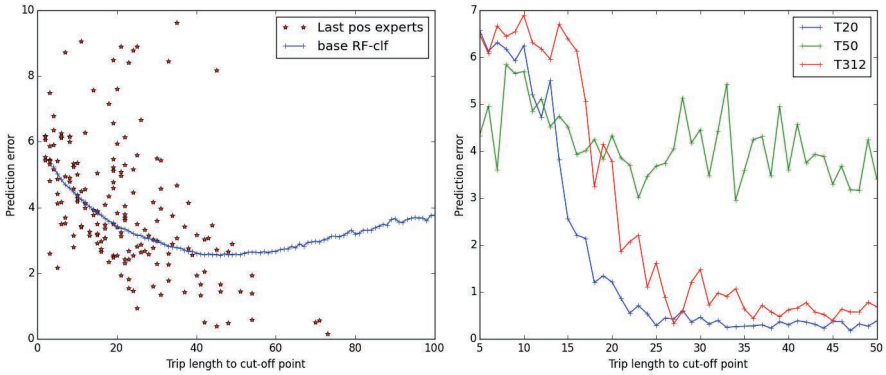


Fig. 5. Left plot shows the cross validated prediction error of the base model (RFR) as a function of trip length. The red stars indicate the prediction error of the base line estimators. The right plot shows the prediction error for three last position expert models plotted against initial trip length. At some cut-off positions a longer initial track can lead to dramatically reduction of the prediction error (blue and red curve). See text for further details.

gradient boosting regressor performed slightly better if the size of the training set was above 4000 (see right plot in Fig.4). Since the difference in performance is very small, no further investigations in this direction have been carried out.

3.2 Taxi Trajectory Prediction

The framework for the final destination prediction was the same as for the travel time prediction. For the trips with sufficient training data the expert model based on the last known position is used. For the other trips, the predicted position was the weighted average of the single base models. The evaluation metric for this task was the Mean Haversine Distance between the predicted and the true position. It was calculated as follows:

$$a = \sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) \tag{4}$$

$$d = 2\,r\,\arctan\left(\sqrt{\frac{a}{1-a}}\right) \tag{5}$$

where ϕ is the latitude, λ is the longitude, r is the Earth's radius, and d is the distance between the two locations, respectively.

Figure 5 summarizes the result. The left plot shows the prediction error of the Random Forest Regressor (blue curve) as a function of the initial trip length for the base model. The error decreases fast to values below 3.0 because very little information can be gained from the first few points. Differently to the time prediction case, the error start to increase around a trip length of 50

because longer trips are more likely to go outside the main city center leading to considerable higher prediction errors. Similar to the previous section, the red stars in the left plot show the prediction error of a simple baseline estimator trained on a selection of the training set (e.g. based on the last position and trip length). For some test trips the predicted error decreases dramatically compared to the base model. Further investigations revealed, that for certain trips the initial trip length is a very strong indicator for the final destination. The right plot of 5 shows the dependency of the prediction error on the initial trip length. Especially for positions close to a sightseeing location, shorter trips indicate taxis going away from it, where as longer trips are likely to end up there (see also Fig. 1a).

4 Conclusion

The remaining traveling time of a taxi depends mainly on the current position and heading of the taxi. For some parts of the city the prediction of final destination can be very precise, for others only a tendency can be obtained. The specific nature of the used error function in the Kaggle competition made it necessary, to predict very short trips with high precision since they were weighted considerably higher in the final score. For the optimization of a taxi dispatching systems, however, the very short trips are of less importance, since the information gained based on some additional observed points is high. Thus it is very likely that there is a taxi cab close by with a considerable longer trajectory, which allows for a more precise prediction in this case.

Acknowledgements

I would like to thank Kaggle.com, GeoLink, and the ECML workshop organizers for their work in making the competition a success, and my co-workers at the Knowledge-based Vision Group at the Software Competence Center Hagenberg for numerous useful comments and suggestions.

References

1. Breiman, L.: Random forest. Machine Learning **45**(1), 5–32 (2001). DOI 10.1023/A
2. Castro, P.S., Zhang, D., Telecom, I.M.t.: From Taxi GPS Traces to Social and Community Dynamics : A Survey. ACM Computing Surveys **46**(2), 34 (2013)
3. David Opitz, R.M.: Popular Ensemble Methods: An Empirical Study. J. Artif. Intell **11**, 169–198 (1999)
4. Friedman, J.H.: Stochastic gradient boosting. Computational Statistics and Data Analysis **38**(4), 367–378 (2002)
5. Froehlich, J., Krumm, J.: Route Prediction from Trip Observations. Proceedings of SAE World Congress **2193**(2008-01-0201), 53 (2008)
6. Han, J., Kamber, M., Tung, A.K.H.: Spatial Clustering Methods in Data Mining: A Survey. Geographic Data mining and knowledge discovery **2** (2001)

7. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. Lecture Notes in Computer Science (LNCS) **6683**, 507–523 (2011)
8. Liu, L., Andris, C., Biderman, A., Ratti, C.: Uncovering Taxi Driver's Mobility Intelligence through His Trace. SENSEable City Lab, Massachusetts Institute of Technology, USA. (2009)
9. Moreira-Matias, L., Azevedo, J., Mendes-Moreira, J., Ferreira, M., Gama, J.: The Geolink Taxi Service Prediction Challenge. ECML/PKDD (2015)
10. Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: Predicting taxi-passenger demand using streaming data. IEEE Transactions on Intelligent Transportation Systems **14**(3), 1393–1402 (2013)
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
12. Rokach, L.: Ensemble-based classifiers. Artificial Intelligence Review **33**(1-2), 1–39 (2010)