Using Metalearning for Prediction of Taxi Trip Duration Using Different Granularity Levels

Mohammad Nozari Zarmehri(✉) and Carlos Soares

INESC TEC, Faculdade de Engenharia, Universidade do Porto,
Rua Dr. Roberto Frias, 378, Porto, Portugal
{mohammad.nozari,csoares}@fe.up.pt

Abstract. Trip duration is an important metric for the management of taxi companies, as it affects operational efficiency, driver satisfaction and, above all, customer satisfaction. In particular, the ability to predict trip duration in advance can be very useful for allocating taxis to stands and finding the best route for trips. A data mining approach can be used to generate models for trip time prediction. In fact, given the amount of data available, different models can be generated for different taxis. Given the difference between the data collected by different taxis, the best model for each one can be obtained with different algorithms and/or parameter settings. However, finding the configuration that generates the best model for each taxi is computationally very expensive. In this paper, we propose the use of metalearning to address the problem of selecting the algorithm that generates the model with the most accurate predictions for each taxi. The approach is tested on data collected in the Drive-In project. Our results show that metalearning can help to select the algorithm with the best accuracy.

Keywords: Metalearning · Data mining · Machine learning · Trip duration prediction

1 Introduction

With fast-growing of Intelligent Transportation Systems (ITS) and Advanced Travelers Information Systems (ATIS), data collected by those systems can be useful to understand and improve processes in taxi companies and other organizations dealing with transportation, i.e. public transportation companies, logistics companies, and local government.

An example of a problem that can benefit from the analysis of data is trip duration in taxi companies; Especially knowing the estimated trip time duration beforehand can be very informative for taxi companies, drivers, and passengers to make the right decision for the scheduling and route planning. Data concerning the taxi trips (essentially GPS data) collected by taxis can be used for that purpose.

E. Fromont et al. (Eds.): IDA 2015, LNCS 9385, pp. 205–216, 2015.
DOI: 10.1007/978-3-319-24465-5_18

Data mining approaches can be used for the prediction of the trip duration. Using the data collected by taxis, these approaches relate trip duration with several variables describing the trip like origin, destination, time of day, day of week, and the weather.

Several algorithms have been introduced and can be used for the prediction of trip duration. But their predictive performance varies and causes several challenges. An important challenge for using data mining is to find out which algorithm has the best performance for a specific problem. But it has already been shown that there is no commonly best algorithm for a broad problem domain [1]. Algorithm selection for a specific problem is either based on a trial-and-error approach or expert advice. Neither way is thoroughly acceptable for the end user who wishes to access the technology cost-effectively [2]. An approach to deal with this problem is metalearning [3]. Metalearning uses a machine learning approach to relate the performance of machine learning algorithms with the characteristics of the data.

The problem of algorithm selection is more complex in applications with multiple sources of data (e.g., multiple taxis). In this case, it may be expected that the best algorithm varies for different sources. For instance, the best algorithm to predict trip duration may vary for different taxis, due to differences in the brand of the vehicle, its usage, and driving habits. Therefore, algorithm selection should be made not at the global level but at a lower one, such as taxi itself.

On the other hand, in applications with multiple sources of data in which the data schema is the same, it is possible that the quality of the model for a given source can be improved by training it with data from other sources. Therefore, the problem of algorithm selection is also extended to the dataset granularity selection. For the purpose of trip duration prediction, each taxi can use its data, data from its neighbors, data collected at the nearest road-side unit, or whole dataset which is collected centrally throughout the city.

In this paper, we investigate the use of a metalearning approach to the problem of algorithm selection in a case study of predicting trip duration for a taxi company. The taxi dataset is obtained from the Carnegie Mellon Portugal project, DRIVE-IN (Distributed Routing and Infotainment through Vehicular Inter-Networking) [4]. Selection is made between four different machine learning algorithms and two levels of granularity; Two levels of granularity are taxi itself and the collected data in whole month. Four machine learning algorithms used at the base-level are: random forest, support vector machines (SVMs), linear regression and decision tree. The experiment is done on the data from five months in 2013, from February to June. In each month, the data is collected by 440 taxis.

The approach is evaluated at the meta-level (i.e. the ability of choosing the most accurate base-level algorithm) and at the base-level (i.e. the base-level performance of the algorithm selected by the metalearning approach). The results obtained are positive at both levels.

2 Background

We start by discussing approaches to predict trip duration (Sect. 2.1) and then metalearning (Sect. 2.2).

2.1 Trip Duration

There has been a significant amount of research on trip duration prediction. Kwon et al. [5] use the flow and occupancy data from single loop detectors and historical trip duration information to forecast trip duration on a freeway. Using real traffic data, they found out that simple prediction methods can provide a good estimation of trip duration for trips starting in the near future (up to 20 min). On the other hand, for the trips starting more than 20 min away, better predictions can be obtained with historical data. The same approach is used by Chien et al. [6]. Zhang et al. [7] propose using a linear model to predict the short-term freeway trip duration. Trip duration is a function of departure time. Their results show that for a small dataset, the error varies from 5 % to 10 % while for a bigger dataset, the variation is between 8 % and 13 %.

Support Vector Regression (SVR) is used for prediction of trip duration by Wu et al. [8]. They utilize real highway traffic data for their experiments. They suggest a set of SVR parameter values by trial-and-error which lead to a model that is able to outperform a baseline model. Balan et al. [9] propose a real-time information system that provides the expected fare and trip duration for passengers. They use historical data consisting of approximately 250 million paid taxi trips for the experiment.

Considering the rapid change of behavior of vehicular networks, using the same algorithm for forecasting the travel time over a long period and for different vehicles, will eventually end in unreliable predictions. Therefore, it is important to find the best algorithm for each context. One possibility is to use a trial and error approach. This means finding out the algorithm that fits best to the specific dataset (i.e. for a specific vehicle and for a specific period) by evaluating multiple algorithms and choosing the best one [10]. This approach would be very time consuming, given the amount of alternatives available. One alternative approach is metalearning which is still missing.

2.2 Metalearning

The algorithm selection problem was formally defined by Rice in 1976 [11]. The main question was to predict which algorithm has the best performance for a specific problem.

The first formal project in this area was MLT project [12]. The MLT project creates a system called *Consultant-2* which can help to select the best algorithm for a specific problem.

Over the years, metalearning research has addressed several issues [13]. It may be important to select the best base-level algorithm not for the whole dataset, but rather for a subset of the examples [14] or even for individual examples [15].

Tuning the parameters of base-level algorithms is another task that metalearning can be helpful to (e.g. the kernel width of SVM with Gaussian kernel [13,16]). Rijn et al. [17] have investigated the use of metalearning for algorithm selection on data streams. The metafeatures are calculated on a small data window at the start of the data stream. Metalearning uses this metafeatures to predict which algorithm is the best in the next data windows.

3 Methodology

In this section, the data used in this work (Sect. 3.1), the metalearning approach (Sect. 3.3) and the evaluation methodology (Sect. 3.4) are presented.

3.1 Taxi Dataset

The dataset is obtained from a large-scale scenario [4], one of the taxi companies in the city of Porto. Porto is the second largest city in Portugal, with an area of 41.3 km^2, and comprises 965 km of roads. It is the central city in a metropolitan area with more than one million inhabitants. There are 63 taxi stands in the city and the main taxi union has 441 vehicles. Each taxi has an on-board unit with a GPS receiver and collects the travel log. The provided dataset by the project [4] consists of five months in 2013 for all the vehicles. The dataset contains 13 variables characterizing events in the data:

id (ID): Event identifier.
driver (D): Taxi driver identifier.
ts (T): Timestamp of the event. It is a UNIX timestamp, in seconds.
st (ST): Taxi state (Offline = 0, Pause = 1, InStand = 2, Free = 3, OnPickup = 4, OnPickupAfterACall = 5, Busy = 6, Login = 7).
Taxi ID (TID): Taxi identifier.
pst (PST): Previous state identifier. This is the same as 'st', but it refers to the state of the previous event.
track (TR): GPS track, encoded with polyline algorithm.
src (S): GPS coordinates of the source position.
dst (DST): GPS coordinates of the destination position.
dd (DD): Distance between src and dst (meters).
n (N): Name of the taxi stand, only if the state is 2 (i.e. if it is stopped in a stand).
pos (P): Location of the taxi stand, only if the state is 2 (i.e. if it is stopped in a stand).
dt (DT): Duration of the trip (seconds).

3.2 Base-level Approach

In this section the methodology which is used at the base-level is presented. In the traditional data mining, each entity E_i is described by a set of features,

X_i, and there is a target variable, Y_i. So the dataset used for the traditional data mining is like $DB = \{E_i, X_i, Y_i\}, \forall i \in \{1, ..., n\}$, while n is the number of entities.

At the base-level, the same scheme is used. The features used at the base-level are described in Sect. 3.1. Each taxi is represented by an entity in the scheme, $E_i = T_i$. The target variable is the trip duration ($Y_i = DT$). So the base-level scheme is like $DB = \{T_i, X_i, DT_i\}, \forall i \in \{1, ..., n\}$. Four algorithms are applied on the dataset (DB) at the base-level to predict the target variable: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and Linear Regression (LM).

3.3 Metalearning at Different Granularity Levels

In this section the metalearning methodology is presented. The taxi application introduces an interesting challenge for metalearning. Each taxi generates enough data to learn its own model. However, it can be expected that, in some cases, the quality of the model generated from the full set of data, i.e. concerning all taxis, can be better than the model generated solely with "local" data. Therefore, besides selecting an algorithm to learn the best model for a taxi, a decision can be made also concerning whether only data from the taxi or global data.

In terms of the metalearning approach, the possibility of generating meta-examples at different levels of granularity of the data, adds another dimension to the meta-dataset. So for each entity, instead of having just one set of X_i, other feature sets can be generated for different levels or categories of the data, $C_i^1, C_i^2, C_i^3, ..., C_i^k$, where k is the number of levels or categories. Therefore the meta-dataset for using in the metalearning process is $DB = \{T_i, C_i^j, Y_i\}, \forall i \in \{1, ..., n\}, \forall j \in \{1, ..., k\}$.

The proposed model used in this article is shown in Fig. 1.

In the proposed model, there are two different levels: taxi itself and the data for whole month. At the level one, each taxi (T_i) creates a unique category, $C_i^1, \forall i \in \{1, ..., n_1\}$ where n_1 is the number of taxis. The level two has only one category joining all the data from 440 taxis.

So after organizing the dataset in customized format, $DB = \{T_i, C_i^j, Y_i\}, \forall i \in \{1, ..., 440\}, \forall j \in \{1, 2\}$, it is delivered to the performance evaluation process. In this stage, each taxi is evaluated by different algorithms, applying in different levels. As result, for each taxi, there are different performance indicators: P_{ig}^k which means the performance of the algorithm g at level k for taxi i.

$$P_{iw}^j : \quad \forall w \in \{1, \ldots, 4\}, \forall j \in \{1, 2\}, \forall i \in \{1, \ldots, 440\} \tag{1}$$

Where w stands for the algorithms, i indicates taxis, and j shows levels.

On the other hand, the metafeatures are calculated for each taxi and at different levels. In general mf_i^j is the calculated metafeatures for taxi i at the level j. For each taxi, the best performance obtained from the performance evaluation part is selected according to the Eq. 2:

$$P_{best\,i} = \max_{w,j}(P_{iw}^j) \ , \quad \forall w \in \{1, \ldots, 4\}, \forall j \in \{1, 2\} \tag{2}$$

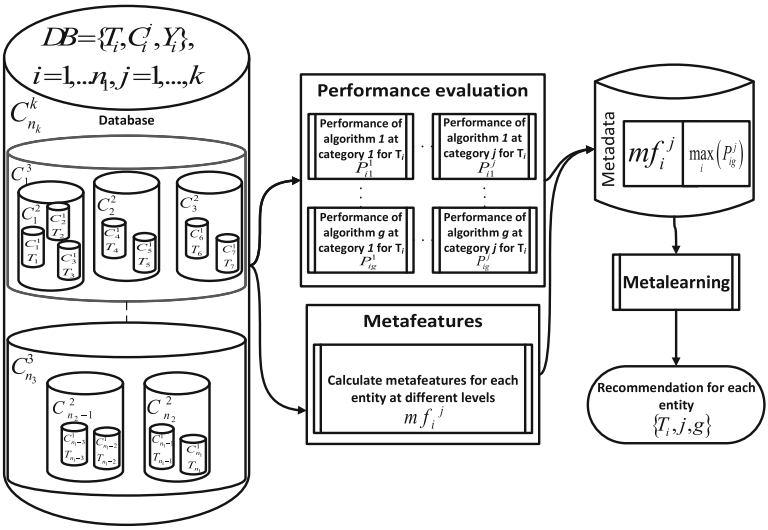


Fig. 1. Proposed methodology used for metalearning

Finally the metadata structure for each taxi consists of the taxi identification, metafeatures for the first and the second level and the best performance obtained from Eq. 2.

$$T_i,\ mf_i^1,\ mf_i^2,\ P_{best\,i} \tag{3}$$

The main idea in metalearning is to find out the best algorithm and the best level to apply the algorithm depending on the metafeatures obtained at different levels. Consequently, the metalearning maps the extracted features from the original datasets to the best performance obtained at different levels by applying different algorithms on the original dataset.

Our model recommends a level and an algorithms for each taxi in which, applying the recommended algorithm on the recommended level produces the best performance with high probability (see Eq. 4).

$$Model\ Output\ :\ \{\underbrace{T_i}_{taxi},\ \underbrace{j}_{recommended\,level},\ \underbrace{g}_{recommended\,algorithm}\ \} \tag{4}$$

3.4 Evaluation

Base-level Evaluation. At the base-level, the problem of prediction of the trip duration is a regression problem. Each algorithm is applied on the dataset and tried to predict the trip duration. This prediction is evaluated by the Normalized Root-Mean-Square Error (NRMSE). RMSE is a frequently used measure which shows the differences between the predicted value by a model and the actual

observed value. In results, the NRMSE is the RMSE divided by the standard deviation of the variable being predicted (See Formulas 5 and 6). Using R [18], the package hydroGOF [19] is used for calculation of NRMSE. The standard deviation is used for the normalizing the RMSE.

$$RMSE = \sqrt{\frac{\sum (\hat{Dt}_i - Dt_i)^2}{n_1}} \tag{5}$$

$$NRMSE = 100 * \frac{RMSE}{\sigma} \tag{6}$$

Where n_1 is the length of the predicted values, σ is the standard deviation of the predicted variable, Dt_i is the actual trip duration, and $\hat{Dt}_i$ is the predicted trip duration. Having the NRMSE for all the possible runs, the algorithm with the best NRMSE (the lowest one) is selected as the best algorithm for each taxi to be used at the meta-level.

Meta-level Evaluation. At the meta-level, the proposed model predicts a base-level algorithms along the level of granularity which will have the best performance (lowest NRMSE) for a given taxi and month. Therefore, the problem in this level is a classification problem. This decision is taken based on metafeatures describing the dataset characteristics.

The performance of the proposed model is evaluated by the accuracy of the prediction. In addition, we also evaluate the performance of the proposed model relative to the possible range of base-level performance. $Scaled_{error}$ shows the relative NRMSE of the metalearning model with respect to the best and the worst NRMSE of the base-level. It is shown in the following equation:

$$Scaled_{error} = \frac{NRMSE_{ML} - NRMSE_B}{NRMSE_W - NRMSE_B} \tag{7}$$

Where $NRMSE_{ML}$ is the NRMSE of the proposed metalearning model, $NRMSE_B$ is the best NRMSE obtained by the base-level algorithms, and $NRMSE_W$ is the worst NRMSE obtained by the base-level algorithms. The range of $Scaled_{error}$ is between 0 and 1. In addition, the lower the $Scaled_{error}$ the better performance is expected for the meta-level experiment.

Metafeatures. The extracted metafeatures noted above, are described briefly in this section. A comprehensive study was done by Peng et al. [20] for feature selection. Totally 31 metafeatures were proposed to describe the structure of the dataset. These metafeatures are selected based on the regression problem. Their effectiveness through extensive experiments were evaluated. A list of all metafeatures that we used for this study with a brief description is provided in Table 1. The detail description of each metafeature is explained in [20].

Table 1. Extracted metafeatures used in metalearning

No.	Feature description
1	Number of examples
2	$log(10)$ of the number of examples
3	Number of attributes
4	Ratio of number of examples by number of attributes
5	$log(10)$ of the ratio of number of examples by number of attributes
6	Number of continuous attributes
7	Number of symbolic attributes
8	Number of binary attributes
9	Proportion of continuous attributes
10	Proportion of symbolic attributes
11	Proportion of binary attributes
12	Correlation between continuous attributes
13	Average absolute correlation between continuous attributes
14	Minimum absolute correlation between continuous attributes
15	Maximum absolute correlation between continuous attributes
16	The ratio between the standard deviation and the standard deviation of alpha trimmed mean
17	Number of continuous attributes with outliers
18	Proportion of continuous attributes with outliers
19	Correlation matrix between attributes and target
20	Average correlation continuous attribute/target
21	Minimum correlation continuous attribute/target
22	Maximum correlation continuous attribute/target
23	Check if standard deviation is larger than mean
24	Ratio of the standard deviation and the mean of the target attribute
25	Sparsity based on the coefficient of variation
26	Sparsity based on the absolute coefficient of variation
27	Standard deviation of the proportions of a histogram with 100 bins of target values
28	textith.outlier value, as calculated for the continuous attributes
29	Outlier detection based on the notion of outliers used for continuous attributes
30	Mean distance between each target value and its two neighbors (sorted by value)
31	Average mean distance between each target value and its two neighbors (sorted by value)

4 Results

4.1 Meta-level Results

The overall results of the calculated $Scaled_{error}$ for each month are shown in Fig. 2. The results seem interesting while the $Scaled_{error}$ is very low and near zero. It shows that the performance of the meta-level is close to the best performance obtained by the base-level.

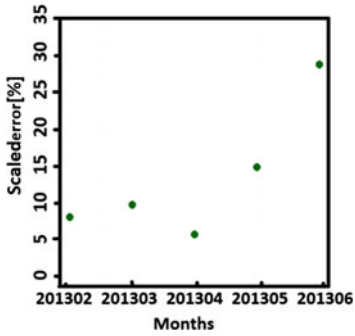


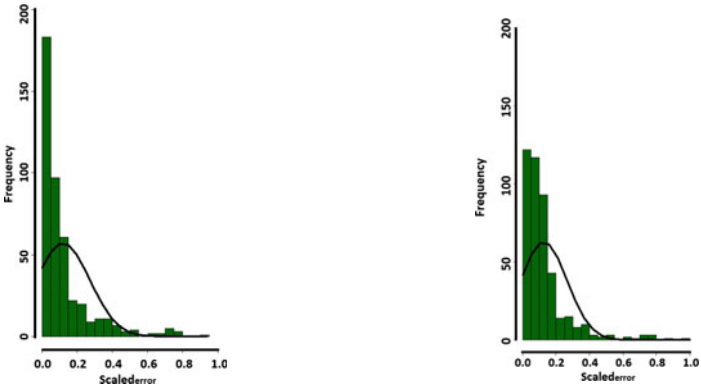
Fig. 2. The average $Scaled_{error}$[%] over all taxis for each month

This result also illustrates the usefulness of using metalearning. By using the dataset characteristics, the metalearning can guess the algorithm with the best performance at the base-level that should be used. It reduces the cost of running several algorithms on probably large datasets to find the one with the best performance at the base-level.

The distribution of calculated $Scaled_{error}$ for each taxi is shown in Fig. 3. As we expected, the density concentrated around zero. This results show that the metalearning is useful because the results of metalearning are almost near the best performance obtained at the base-level. The normal distribution for RF and DT algorithms (black lines) show that on average the $Scaled_{error}$ is less than 0.2 in both cases. Although the density of $Scaled_{error}$ for RF algorithm has high concentration near the origin.

4.2 Base-level Results

To know the performance of the base-level, Fig. 4 shows the box-plot of calculated NRMSE for different taxis and in the different levels of granularity in each month. It can be seen that the NRMSE for all months is less than 5 %. The average NRMSE for each month is around 1 %. So the base-level error on average is 1 % which sounds considerably good. This means that the base-level algorithms can predict the trip duration very precisely.



(a) RF algorithm (b) DT algorithm

Fig. 3. Distribution of $Scaled_{error}$ over each taxi

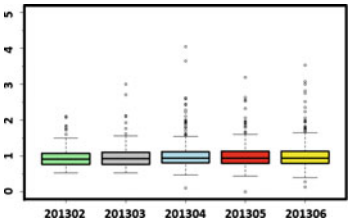


Fig. 4. NRMSE[%] for different months

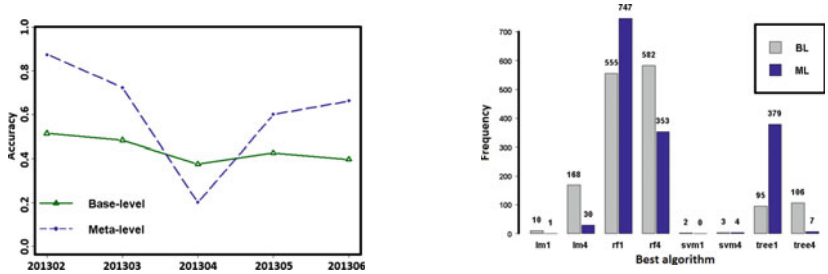
4.3 Base-level vs. Meta-level Results

In metalearning one of the most important metric for evaluation is the accuracy. The comparison of accuracy between the base-level and the meta-level is presented in Fig. 5a. According to this result, the performance of the meta-level outperforms the base-level for most of the months. In April 2013, due to the lack of enough observations for calculating the metafeatures, the performance of metalearning is dropped.

The accuracy of the base-level is calculated based on the majority algorithm with the best performance at the base-level. Although, the accuracy of the metalearning is calculated by considering the algorithm with the best performance at the base-level. On average, the meta-level accuracy is 17 % higher than the base-level accuracy that can be converted to 39 % improvement on the base-level.

To obtain the algorithm with the best performance at the base-level, performing a lots of algorithms is required. Therefore, the computational cost is considerably high. But by using metalearning, the algorithm with the best performance is found by high probability and lower computational cost.

In addition, the prediction of the best algorithm by metalearning is almost followed the best algorithm obtained by the base-level (Fig. 5b).



(a) Accuracy (b) The best performance algorithm

Fig. 5. Base-level (BL) vs. meta-level (ML)

5 Conclusion

We proposed the use of metalearning for prediction of trip duration. The experiments are performed on the taxi dataset from Drive-In project. The machine learning and data mining algorithms are performed at two different levels of granularity: taxi and month levels. The results show that the metalearning can help predicting the algorithm with the best performance at the base-level with high accuracy. Furthermore the performance of the base-level itself is also considerably applicable. Therefore, the overall results show that the metalearning predicts the trip duration with the error rate less than 5 %.

Acknowledgment. This work is financed by the ERDF - European Regional Development Fund through the COMPETE programme (operational programme for competitiveness) within project GNOSIS, cf. "FCOMP-01-0202-FEDER-038987". It is also funded by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT) through projects "NORTE-07-0124-FEDER-000057" and "NORTE-07-0124-FEDER-000059". The work is also financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within projects "FCOMP-01-0124-FEDER-037281" and "SFRH/BD/71438/2010". The research leading to these results has also received funding from the ECSEL Joint Undertaking, the framework programme for research and innovation horizon 2020 (2014–2020) under grant agreement n° 662189-MANTIS-2014-1.

References

1. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)

2. Giraud-Carrier, C., Vilalta, R., Brazdil, P.: Introduction to the special issue on meta-learning. Mach. Learn. **54**(3), 187–193 (2004)
3. Brazdil, P., Giraud-carrier, C., Soares, C., Vilalta, R.: Metalearning: applications to data mining. In: Cognitive Technologies. Springer, Heidelberg (2009)
4. Cmuportugal.org: Drive-in: Distributed routing and infotainment through vehicular inter-networking (2014)
5. Kwon, J., Coifman, B., Bickel, P.: Day-to-day travel-time trends and travel-time prediction from loop-detector data. Transp. Res. Rec.: J. Transp. Res. Board **1717**(1), 120–129 (2000)
6. Chien, S.I.J., Kuchipudi, C.M.: Dynamic travel time prediction with real-time and historic data. J. Transp. Eng. **129**(6), 608–616 (2003)
7. Zhang, X., Rice, J.A.: Short-term travel time prediction. Transp. Res. Part C: Emerg. Technol. **11**(3), 187–210 (2003)
8. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. **5**(4), 276–281 (2004)
9. Balan, R.K., Nguyen, K.X., Jiang, L.: Real-time trip information service for a large taxi fleet. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, MobiSys 2011, pp. 99–112. ACM, New York (2011)
10. Brazdil, P., Soares, C., Costa, J.D.: Ranking learning algorithms: using IBL and meta-learning on accuracy and time results. Mach. Learn. **50**, 251–277 (2003)
11. Rice, J.R.: The algorithm selection problem. In: Rubinoff, M., Yovits, M.C. (eds) Advances in Computers, vol. 15, pp. 65–118. Elsevier (1976)
12. Kodratoff, Y., Sleeman, D., Uszynski, M., Causse, K., Craw, S.: Building a machine learning toolbox (1992)
13. Rossi, A.L.D., de Leon Ferreira de Carvalho, A.C.P., Soares, C., de Souza, B.F.: MetaStream: a meta-learning based method for periodic algorithm selection in time-changing data. Neurocomputing **127**, 52–64 (2014)
14. Brodley, C.: Recursive automatic bias selection for classifier construction. Mach. Learn. **20**(1–2), 63–94 (1995)
15. Todorovski, L., Džeroski, S.: Combining classifiers with meta decision trees. Mach. Learn. **50**(3), 223–249 (2003)
16. Soares, C., Brazdil, P.B., Kuba, P.: A meta-learning method to select the kernel width in support vector regression. Mach. Learn. **54**(3), 195–209 (2004)
17. van Rijn, J.N., Holmes, G., Pfahringer, B., Vanschoren, J.: Algorithm selection on data streams. In: Džeroski, S., Panov, P., Kocev, D., Todorovski, L. (eds.) DS 2014. LNCS, vol. 8777, pp. 325–336. Springer, Heidelberg (2014)
18. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2014)
19. Zambrano-Bigiarini, M.: hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series (2014) R package version 0.3-8
20. Peng, Y.H., Flach, P.A., Soares, C., Brazdil, P.B.: Improved dataset characterisation for meta-learning. In: Lange, S., Satoh, K., Smith, C.H. (eds.) DS 2002. LNCS, vol. 2534, pp. 141–152. Springer, Heidelberg (2002)