Feature Selection Issues in Long-Term Travel Time Prediction

Syed Murtaza Hassan[1], Luis Moreira-Matias[1], Jihed Khiari[1](✉),
and Oded Cats[2]

[1] NEC Laboratories Europe, 69115 Heidelberg, Germany
{syed.hassan,luis.matias,jihed.khiari}@neclab.eu
[2] Department of Transport and Planning, TU Delft, 2600 Delft, Netherlands
o.cats@tudelft.nl

Abstract. Long-term travel time predictions are crucial for tactical and operational public transport planning in schedule design and resource allocation tasks. Similarly to any regression task, its success considerably depend on an adequate feature selection framework. In this paper, we approach the myopia of the State-of-the-Art method RReliefF on mining relevant inter-relationships of the feature space relevant for reducing the entropy around the target variable on regression tasks. A comparative study was conducted using baseline regression methods and LASSO as a valid alternative to RReliefF. Experimental results obtained on a real-world case study uncovered the bias/variance reduction obtained by each approach, pointing out promising ideas on this research line.

Keywords: Travel time prediction · Machine learning · Regression · Feature selection

1 Introduction

One of the most common research problems in transportation is travel time prediction (TTP). The literature on this topic is extensive and covers different application domains such as fleet management, monitoring, control, mass transit and individual navigation [1]. Hereby, we focus on public transport in general and buses in particular. It is possible to distinguish short and long-term travel time prediction problem based on the prediction horizon (e.g. threshold of 2–3 h). Operational tasks (e.g. timetable design) or resource allocation (e.g. vehicle and crew scheduling) requires long-term TTP.

A traditional approach to TTP is regression analysis. It comprises a large number of techniques to estimate the relationship between a set of predictors (i.e. features) and a dependent variable:

$$\hat{f} : x_i, \theta \rightarrow \mathbb{R} \text{ such that } \hat{f}(x, \theta) = f(x_i) = y_i, \forall x_i \in X, y_i \in Y \quad (1)$$

where $f(x_i)$ denotes the true unknown function which is generating the samples' target variable and $\hat{f}(x_i, \theta) = \hat{y}_i$ be an approximation dependent on the feature

H. Boström et al. (Eds.): IDA 2016, LNCS 9897, pp. 98–109, 2016.
DOI: 10.1007/978-3-319-46349-0_9

vector x_i and an unknown parameter vector $\theta \in \mathbb{R}^n$ (given by some induction model M). Notorousily, this approximation will be as good as the adequacy of M to the dependence structure of f as well as the relevancy of the input feature space X. If it has a low number of features, it may not explain the variance of Y, thus leading M to biased models. Coversely, for a large set of features, we may be using features with a low predictive power. In consequence, M may output very complex models which lead to optimal fits on the input dataset (i.e. *local minima*) but a considerably lower ones when tested in any generic inference task. These phenomenons are known as *underfitting* and *overfitting*, respectively.

Automatic Feature Selection [2] is a subfield of study focused on developing algorithms capable of defining adequate feature spaces for supervised learning problems. The idea is to find the feature subset that guarantees solutions (i.e. models) close to the global minima of our generalization error by defining which features to use and which to drop on a particular regression/classification problem. There are mainly two types of feature selection algorithms: (i) *filters*, where the induction model is not take into account to select an adequate feature subset and (ii) *wrappers*, where the feature subset selection process takes into account the induction model (typically through an encapsulated optimization framework). In this paper, we are focused on discussing issues around this topic (i), as well as its impact in the context of long-term TTP tasks.

In transportation science, it is known that the main determinants of bus running times are route length, passenger activity at stops and the number of traffic signals (e.g. [3,4]). Other studies also added driver response to the deviation from the schedule as an explanatory variable [5,6]. However, all of those have estimated linear regression models to identify the impact of potential explanatory variables on bus running times. Consequently, the resulting models often have very limited predictive power.

Attaining better bus travel time predictions can have significant consequences for passenger delays, operator's performance fines and the efficiency of its resource allocation. The inherently complex and uncertain operational environment in which urban bus service operate call for the development of more sophisticated models that can capture non-linear relations between system variables. To the authors' best knowledge, the literature to handle this specific issue is scarce. Mendes-Moreira *et al.* [7] compared Random Forests (RF), Support Vector Machine Regression (SVR) and Progression Pursuit Regression (PPR). On the other hand, the well-known RReliefF [8] was proposed to do an adequate feature selection for each route. As many other methods from the RELIEF*-family, RReliefF is an instance-based learning method which leverages on the concept of neighborhood to define features that can (or cannot) contribute significantly to the entropy reduction on estimating the target variable Y. Consequently, as many other instance-based methods (e.g. k-nearest neighbors), it is highly dependent on an adequate setting of a distance metric that serves this specific purpose (which can easily vary from problem to problem). Moreover, it also has limitations on evaluating inter-relationships among the feature set X which can lead to this effect.

This paper is focused on studying the effects of RReliefF *myopia* to unrealistic distance functions and/or interrelationship on the feature set relevant for predicting the target variable value. To do it so, we propose an the Least Absolute Shrinkage and Selection Operator (LASSO) as a simple and yet valid alternative to RReliefF for this particular domain. The idea is to leverage on the priority that LASSO gives on the bias error reduction - in contrast to RReliefF. Consequently, our contributions are twofold: (1) a practical demonstration on RReliefF limitations through the study of its impact on particular application area; (2) the introduction of LASSO as a valid alternative to this problem due to the high number of relevant interactions among different predictors/features that can reduce bias error. Experimental results of applying the same baseline predictors to a particular real-world case study uncovered the potential of our novel approach.

The remaining of this paper is organized as follows: we start by describing the case study and related data sources. The methodology section presents the feature selection algorithms studied as well as a brief description of the baseline regressors employed. The experimental setup is detailed in Sect. 4, followed by a result report and a comprehensive discussion. We conclude with final remarks and future research directions.

2 Case Study

Our case study is a large urban bus operator in Sweden. We collected data from four high-frequency (maximum planned headway of 11 min between 7:00–19:00) routes A1/A2/B1/B2, i.e. two bus lines A/B. Line A connects residential areas to a public transport interchange hub as well as major shopping areas. B connects the southern parts of the city to the city center, traversing through an interchange, major hospitals as well as a logistic center. The bus operator defines two schedules; a summer schedule taking effect from June 19th till December 14th and a winter schedule taking effect from December 15th till June 18th. Our study covers a period of six months between August 2011 and January 2011 thus including both schedules.

As part of the preprocessing step, a trip pruning was performed by removing trips where more than 80 % of link travel times were missing. In addition, we performed data imputation on the remaining samples by following the interpolation procedure suggested in [9]. The dwell times were also pruned by using the 99 % percentile to remove erroneous measurements. Table 1 presents an overview of the resulting dataset, detailed per route. It contains the (i) total number of trips (NT), (ii) number of stops and (iii) Round Trip Times (RTT).

2.1 Feature Generation

The original features are schedule departure time, daytype and vehicle ID. Unlike RF, SVR and PPR do not support categorical values. Therefore, it is required to

Table 1. Statistics per route. The values are as mean ± s.d. Times in seconds.

	NTrips	Stops	Daily trips	Round trip times
A1	17953	33	134 ± 27	3017 ± 425
A2	16353	33	133 ± 30	2755 ± 480
B1	16280	25	137 ± 23	2607 ± 465
B2	16353	25	134 ± 22	2746 ± 448

generate new features based on the original ones. For the type of day, we use one-hot encoding which generates 7 numerical features corresponding to the day type. Vehicle ids associated with less than 0.5 % of total number of trips were grouped into a single cluster. The remaining vehicle ids were clustered using a clustering technique described in the experiments section. This procedure resulted in four additional features.

Figure 1 illustrates the clustering results for route A2. It illustrates the clustering plot (top-left) and the kernel density estimations for the vehicle ids within each cluster. We used the Bayesian Information Criterion (BIC) to determine the best number of clusters $k = 3$ from the interval $K = [2{:}20]$. We note that the three clusters are characterized by slightly different p.d.f. This justifies mapping the ids into three distinct features. Since driver rosters are typically assigned

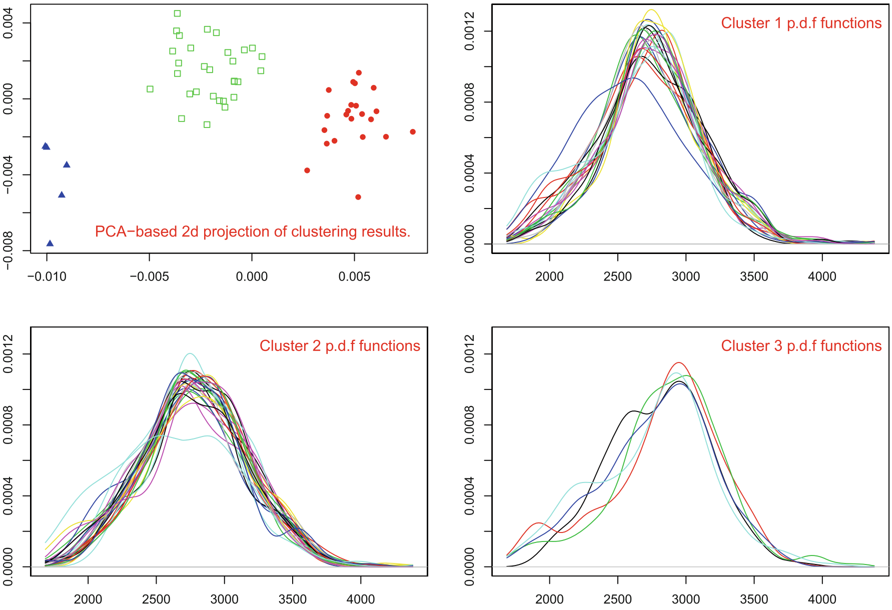

Fig. 1. Clustering results of vehicle ids for route A2.

to individual vehicles throughout their shift, vehicle travel times reflect driving style as well as the propagation of delays from one trip to later ones.

3 Methodology

Feature selection consists in eliminating redundant or non-informative features. Applying feature selection can not only lead to more interpretable models but also attain better results. Redundant features can negatively affect the predictions of models that do not inherently perform such a task. This is also relevant for our TTP framework, where we seek to determine the best set of features for generating predictions. The state-of-the-art method for this domain (proposed by Mendes-Moreira *et al.* [7]) is RreliefF [8]. This instance-based learning algorithm is able to determine features relevance on determining the target variable value. It can handle interdependences on the feature space, missing data and/or different type of functional forms for the dependences. However, its success depends largely on an adequate definition of a distance metric. Moreover, it is focused on reducing variance-type error, neglecting the inter-relationships that can potentially reduce the bias-type one.

Hereby, we compare RReliefF to LASSO as filter feature selection method to highlight why the first is not adequate for this task on long-term TTP problems. This section elaborates formally on the two methods as well on the three used baseline regressors: PPR, RF and SVR.

3.1 Feature Selection Methods

RReliefF was introduced by Kira and Kendell [10]. Its key idea is to rank features based on how well they separate classes. Given a randomly selected instance R_i, this is achieved by searching for its two nearest neighbors, one from the same class called nearest hit H and the other from the different class called nearest miss M. Since a good feature separates different classes, it should have a small distance to H and a large distance to M. The estimate of feature A quality i.e. $W[A]$ is adjusted accordingly. The whole process is repeated for m iterations-where m is a user defined parameter. Finally, features that have a higher value than a given threshold ϕ are selected. Similarly, the ReliefF algorithm [11] deals with classification problems with more than two classes, by considering k *hits* and *misses* rather than two. In regression problems, the predicted value is continuous so we cannot determine if two instances are part of the same class or not. To solve this issue, Robnik-Šikonja and Konokenko [8] introduced RreliefF: a probability measure modeled with the relative distance between the predicted values of the two instances. Similarly to ReliefF, a random instance R_i and its k nearest instances are selected in order to iteratively calculate the weights of input variables based on an user-defined distance metric.

LASSO is a shrinkage and selection method for linear regression introduced by Tibshirani [12]. Similarly to other shrinkage methods, it aims to improve the least-squares estimator by adding constraints on the value of coefficients noted

b. Given an input data matrix of size $N \times p$ (i.e. N samples defined by $p-1$ features and a target y), the LASSO estimate is defined by

$$\hat{b}^{lasso} = \underset{b}{\operatorname{argmin}} \sum\nolimits_{i=1}^{N} (y_i - b_0 - \sum\nolimits_{j=1}^{p} x_{i,j} b_j)^2 \tag{2}$$

subject to

$$\sum\nolimits_{j=1}^{p} |b_j| \leqslant t, t \geqslant 0 \tag{3}$$

The equivalent *Lagrangian form* is

$$\hat{b}^{lasso} = \underset{b}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum\nolimits_{i=1}^{N} (y_i - b_0 - \sum\nolimits_{j=1}^{p} x_{i,j} b_j)^2 + \lambda \sum\nolimits_{j=1}^{p} |b_j| \right\} \tag{4}$$

The L_1-norm penalty of LASSO $\sum_{j=1}^{p} |b_j|$ constrains the solution space to go for simpler, low-coefficient models by forcing some of the $n-1$ features to be shrunk out of the final model. The tuning parameter λ controls the **strength** of the penalty. As it increases, more coefficients are set to zero and hence, less variables are selected. λ is typically set by using a cross-validation search technique over a grid of admissible values.

3.2 Regression Methods

RreliefF and LASSO were tested as filter-type feature selection methods to cope with three baseline regressors: RF, SVR and PPR.

Random Forests is an ensemble method based on classification and regression trees (CART [13]) that was introduced by Leo Breiman in 2001 [14]. The trees are grown by randomly choosing a set of candidate predictors at every node for a sample of the data and then producing the split by choosing the best splitter available. RF combines this with a random selection of samples to train the trees which is referred to as bootstrap aggregating or bagging. RF's hyperparameters are (i) the number of randomly selected predictors to choose from at each split *mtry* and the number of grown trees *ntree*.

Support Vector Machines were introduced by Cortes *et al.* in 1995 [15]. They are primarily binary classifiers that perform their task by constructing hyperplanes in a multidimensional space able to separate instances either linearly on non-linearly. In ϵ-SVM, these hyperplanes are constructed in a way to ensure the largest minimum distance to the training examples. This distance (ϵ) is denominated as *margin*. SVMs can be adapted for regression with a quantitative response by sequentially optimizing an error function where we seek to maximize the geometrical distance between the two hyperplanes $\frac{1}{||w||}$ which is equivalent to minimizing $\frac{1}{2}||w||^2$. To allow examples to be in the margin or to be misclassified, slack variables $\xi_i >= 0$ are introduced. The optimization problem becomes:

$$\underset{w,b}{\arg\min} \frac{||w||^2}{2} + C \times \sum_{i=1}^{n} \xi_i \tag{5}$$

where $C > 0$ is a constant that sets the relative importance of maximizing the margin and minimizing the amount of slack. Kernels are typically used in SVMs to map the data points into higher dimensional feature space, where a linear separation allow a non-linear boundary to be drawn in the original one. Typical kernel include polynomial and radial basis functions. The choice of the kernel depends on the problem and different functions may depend on different hyperparameters.

Projection Pursuit Regression is an additive model that consists of linear combinations of non-linear transformations of linear combinations of explanatory variables (so-called *ridge functions*) [16]. It firstly projects the data matrix of explanatory variables in the optimal direction before applying smoothing functions to those. If *maxterms* (i.e. the number of linear combinations) is sufficiently large, PPR can be considered a universal approximator with considerable similarities to the so-called feed forward neural networks. However and similarly to the latter, complexity constraints need to be formulated to avoid overfitting. The algorithm starts by adding *maxterms* ridge functions. Then, it removes iteratively the least important term until *nterms* terms remain, which is the number of terms in the final model. Both *maxterms* and *nterms* are hyperparameters that need to be tuned beforehand. *optlevel* is a third hyperparameter which controls how thoroughly the models are refitted during this process. To smooth the ridge functions, we use by default Friedman's 'super smoother' *supsmu* which requires to fit the bass/span control.

4 Experiments

The experiments were conducted using the R Software [17]. Data was divided into two sets: a training set and a test set (i-e 70 %/30 %). Statistical independence was assumed to be in place among the routes. Consequently, we ended up having a total of 4 data sets. Vehicle ids were categorized into four groups: one containing all vehicle ids having less than 0.5 % of the total number of trips and 3 obtained through a three-step clustering procedure. First, kernel density estimation was used to generate the p.d.f. for every unique vehicle id. Second, this p.d.f. were clustered by a Gaussian Mixture Model trained using the Expectation-Maximization algorithm. Finally, the Bayesian Information Criterion was used to select the best model.

Package `FSelector` [18] was used for RReliefF. The value used for neighbour.count (the number of nearest examples) in [7] was 10. For robustness reasons, we used neighbour.count = 50 with $m = 100$ iterations. For illustrative purposes on this particular issue, we used 0.1 % of total data set as sample size. Similar results were found for a sample size of 0.5 % and 1.0 % of total data set length. A minimum weight threshold was set as $\phi = 0.01$. The default distance metric of `FSelector`'s implementation of RReliefF was used.

We used `glmnet` [19] procedures for fitting LASSO. The best λ was selected using cross validation.

4.1 Hyperparameter Tuning

Package `caret` [20] was utilized for hyperparameter tuning of RF, SVR and PPR. The two methods used in our experiments for hyperparameter optimization are (i) Grid Search (e.g. [7]) and (ii) Random Search [21]. (i) Grid Search exhaustively considers all the parameter combinations specified in a grid of parameter values. Hence, a high computational effort is required for large grids. A valid alternative introduced by Bergstra and Benghio [21] is Random Search. It consists on conducting independent draws from a uniform density using the same configuration space as the one defined by a regular grid. This approach only evaluates a random subsample of grid points - set to 60 in our case - and presents similar results to the grid one on an efficient manner [21].

PPR has five different hyperparameters: nterms, max.terms, optlevel, bass and span (the two latter for *supsmu*). Random Search was used for tuning nterms Package `kernlab` [22] was used for SVR. SVR has six different hyperparameters: kernel, C (for all kernels), epsilon (for all kernels), sigma (for Radial kernel), scale and degree (only for polynomial kernel). Random Search was used for tuning C, sigma, scale and degree. Finally, Package `randomForest` was used for RF. Grid search was used for tuning both hyperparameters, as well as the ones non explicitly mentioned above.

The three abovementioned base learners were evaluated based on the three resulting feature spaces: (1) feature set with all features (12 features), and as well as the ones given by (2) LASSO and (3) RReliefF. The obtained results were compared using two metrics of interest: RMSE and MAE.

4.2 Results and Discussion

The optimal hyperparameter values for the three distinct setups are displayed in Table 2 for RF, PPR and SVR. Figure 3 shows the results of RReliefF for each of the routes. x-axis is the feature set. y-axis is the weight; boxplots. It is evident that only the departure time has a predictive power accordingly with RReliefF. We therefore select departure time as the only feature from RReliefF method for each route. Figure 2 shows the results of LASSO plots for each of the routes. x-axis are different $\log(\lambda)$ values while y-axis are the coefficients. Features after the cut-off are selected to be the most suitable ones.

Finally, the evaluation of SVR, PPR and RF for the three feature sets for each of the routes are presented in Tables 3 and 4 respectively. The tables clearly show that LASSO performs better than RReliefF on this particular task. RF is the algorithm that benefits less of the feature selection process since this task is inherent of its own modelling process. Figure 3 illustrates the myopia of RReliefF on identifying some of the daytypes as relevant for reducing the bias-error around the target variable. As result, underfitted models (using only scheduled departure time) produce bad results - especially for PPR and SVR. These effects are depicted in Fig. 4, where the deficiency of the models output by either PPR and SVR during the peak hours when fed by RReliefF feature subspaces is highlighted. This effect happens because the daytype variables do

Table 2. Optimal hyperparameters setting.

		PPR			SVR		RF	
		nterms	max.terms	optlevel	σ	C	mtry	ntree
LASSO	A1	3	3	1	1179.65	909.04	1	500
	A2	3	3	1	997.91	335.15	1	500
	B1	3	3	1	1369.19	0.509	1	500
	B2	3	3	1	1008.36	72.25	1	700
RreliefF	A1	7	7	7	65.42	2.84	3	700
	A2	8	8	3	329.67	3.30	3	900
	B1	6	6	3	17.87	22.80	3	900
	B2	7	7	3	71.46	0.076	3	900
ALL	A1	8	8	3	0.15	909.04	6	900
	A2	9	9	3	0.19	318.79	6	500
	B1	11	11	3	0.17	312.95	6	900
	B2	5	5	3	0.24	72.25	6	900

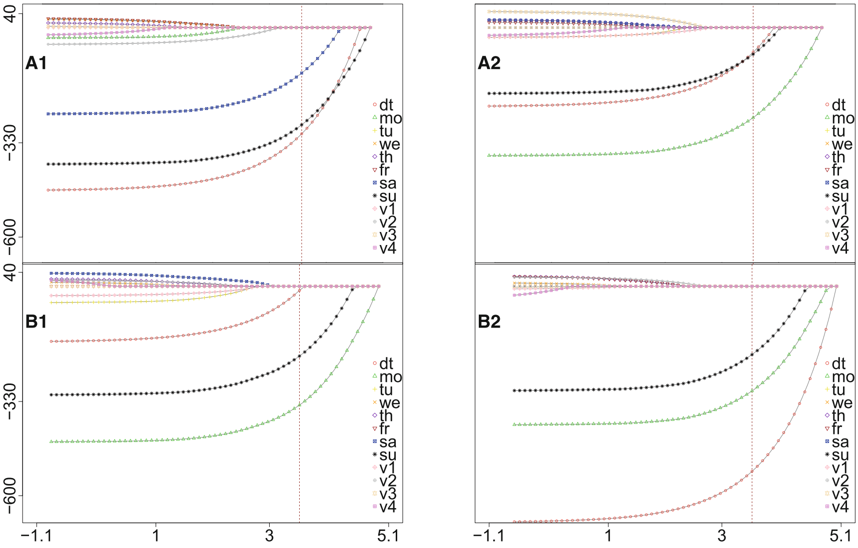

Fig. 2. LASSO results for all routes. A vertical red dashed line is drawn at the best log λ value. This serves as cut.off point. (Color figure online)

not have a particular effect on the variance-error reduction - but mainly only on the bias one. In the authors' opinion, these results illustrate that RReliefF is not the best technique to handle the feature selection task on this particular problem.

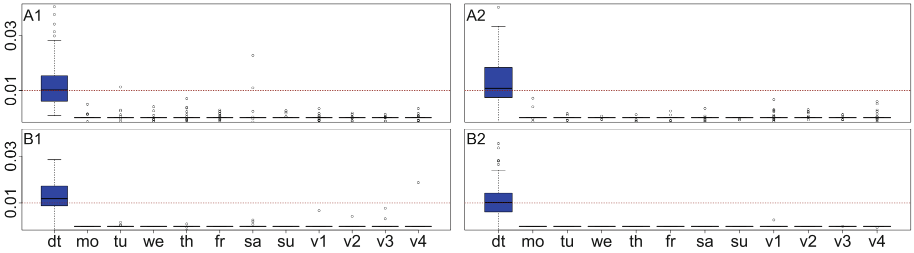

Fig. 3. RReliefF results for all routes. A horizontal red line is drawn at y = 0.01. (Color figure online)

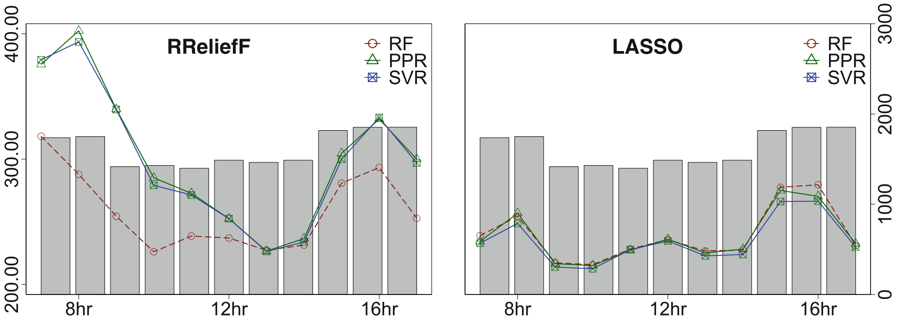

Fig. 4. RRelieF and LASSO comparative analysis (y-axis) using RMSE (scaled on RRelief side) along different scheduled departure times (x-axis). Bars denote the sample size on each timespan (scaled on LASSO side).

Table 3. SVR results for initial, LASSO and RrF-RReliefF feature sets.

Route	RMSE RrF	MAE RrF	RMSE LASSO	MAE LASSO	RMSE ALL	MAE ALL
A1	293.294	224.455	**228.192**	**182.554**	244.888	196.851
A2	260.567	192.483	**196.843**	**154.453**	228.977	180.843
B1	309.361	224.084	**244.650**	**180.188**	281.480	205.387
B2	311.037	231.711	**255.853**	**204.383**	268.029	211.830
ALL	293.564	218.183	**231.384**	**180.394**	255.843	198.477

5 Concluding Remarks

Feature selection is a relevant task in any real-world data mining project. Long-term TTP for public transport planning and/or operational purposes is not an exception. Hereby, we discussed the limitations of RReliefF - the state-of-the-art for this problem. A comprehensive comparison with LASSO was conducted using a real-world case study from a bus operator in Sweden. The obtained results illustrated how dependent RReliefF is on an adequate distance metric that gives different relevance for distinct features - thus leading to a proper normalization of the RReliefF output weights and/or different selection thresholds for each

Table 4. RF and PPR results for initial, LASSO and RrF-RReliefF feature sets.

	RF						**PPR**					
Route	**RMSE RrF**	**MAE RrF**	**RMSE LASSO**	**MAE LASSO**	**RMSE ALL**	**MAE ALL**	**RMSE RrF**	**MAE RrF**	**RMSE LASSO**	**MAE LASSO**	**RMSE ALL**	**MAE ALL**
A1	240.20	190.18	**227.66**	**181.44**	232.72	187.96	311.66	242.49	**231.52**	**186.83**	232.94	188.16
A2	235.65	180.76	**199.84**	**158.91**	203.11	161.73	263.04	195.19	**197.05**	**158.04**	198.80	159.45
B1	266.32	198.87	249.78	189.70	**248.95**	**188.71**	311.68	226.25	**247.69**	**188.21**	248.33	189.21
B2	283.59	223.68	266.62	215.42	**264.51**	**213.91**	316.31	233.50	264.21	213.77	**261.05**	**211.29**
ALL	256.44	198.37	**235.97**	**186.37**	237.32	188.08	300.67	224.36	**235.13**	**186.71**	235.28	187.03

feature accordingly to each one's contribution on model's bias reduction. As future work, we intend to explore further supervised filters for dimensionality reduction purposes on this task - such as Auto-Encoders.

References

1. Moreira-Matias, L., Mendes-Moreira, J., de Sousa, J.F., Gama, J.: Improving mass transit operations by using AVL-based systems: a survey. IEEE Trans. Intell. Transp. Syst. **16**(4), 1636–1653 (2015)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
3. Mishalani, R., McCord, M., Forman, S.: Schedule-based and autoregressive bus running time modeling in the presence of driver-bus heterogeneity. In: Hickman, M., Mirchandani, P., Voß, S. (eds.) Computer-Aided Systems in Public Transport, pp. 301–317. Springer, Heidelberg (2008)
4. Berkow, M., El-Geneidy, A., Bertini, R., Crout, D.: Beyond generating transit performance measures. Transp. Res. Rec. J. Transp. Res. Board **2111**(1), 158–168 (2009)
5. El-Geneidy, A., Horning, J., Krizek, K.: Analyzing transit service reliability using detailed data from automatic vehicular locator systems. J. Adv. Transp. **45**(1), 66–79 (2011)
6. Mazloumi, E., Rose, G., Currie, G., Sarvi, M.: An integrated framework to predict bus travel time and its variability using traffic flow data. J. Intell. Transp. Syst. **15**(2), 75–90 (2011)
7. Mendes-Moreira, J., Jorge, A., de Sousa, J., Soares, C.: Comparing state-of-the-art regression methods for long term travel time prediction. Intell. Data Anal. **16**(3), 427–449 (2012)
8. Robnik-Šikonja, M., Kononenko, I.: An adaptation of relief for attribute estimation in regression. In: Proceedings of the Fourteenth International Conference on Machine Learning, ICML 1997, pp. 296–304 (1997)
9. Mendes-Moreira, J., Moreira-Matias, L., Gama, J., de Sousa, J.: Validating the coverage of bus schedules: a machine learning approach. Inf. Sci. **293**, 299–313 (2015)
10. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Proceedings of the Ninth International Workshop on Machine Learning, pp. 249–256 (1992)
11. Kononenko, I.: Estimating attributes: analysis and extensions of RELIEF. In: Bergadano, F., Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994). doi:10.1007/3-540-57868-4_57

12. Tibshirani, R.: Regression shrinkage and selection via the LASSO. J. Roy. Stat. Soc. Ser. B (Methodol.) **58**(1), 267–288 (1996)
13. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press, New York (1984)
14. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
15. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
16. Friedman, J., Stuetzle, W.: Projection pursuit regression. J. Am. Stat. Assoc. **76**(376), 817–823 (1981)
17. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation, Vienna (2012)
18. Romanski, P.: Fselector: selecting attributes. R package version 0.19 (2009)
19. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**(1), 1 (2010)
20. Kuhn, M.: Caret package. J. Stat. Softw. **28**(5), 1–26 (2008)
21. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(1), 281–305 (2012)
22. Zeileis, A., Hornik, K., Smola, A., Karatzoglou, A.: kernlab-an S4 package for kernel methods in R. J. Stat. Softw. **11**(9), 1–20 (2004)