COMPARATIVE EVALUATION OF ALGORITHMS FOR GPS DATA IMPUTATION

Tao Feng, Eindhoven University of Technology, t.feng@tue.nl

Harry J. P. Timmermans, Eindhoven University of Technology

ABSTRACT

GPS data collection has been increasingly considered as an alternate means of data collection, replacing the traditional travel survey methods. Several algorithms which vary from informal ad-hoc approaches to advanced machine learning methods have been reported in the literature to extract activity and travel information from GPS traces. However, the differences in the performance of different algorithms are scarcely addressed. In this paper we evaluate the relative performance of different imputation algorithms for GPS data imputation by incorporating the naive Bayesian, Bayesian network, logistic regression, multilayer perception, support vector machine, decision table and C4.5. The accuracy of imputation results of various methods are compared using the GPS data collected in The Netherlands. Results show that the Bayesian network has a better performance than other algorithms according to the correctly identified instances and Kappa values for both training data and test data. Especially, the Bayesian network shows a stronger capability than other methods in the aspect of prediction generalisation.

Keywords: Classification algorithm, Bayesian, decision tree, rules

INTRODUCTION

The research issue of GPS data imputation and data collection have been increasingly discussed in recent years to further investigate the possibility of replacing the traditional survey methods, i.e. paper-based questionnaire, telephone calls, in a scale of either limited samples (Moiseeva, et al., 2010) or the national travel survey (Feng and Timmermans, 2011). One of the most important issues of GPS data imputation is how to generate as accurate as possible the activity and travel data in space and time. Although in many cases a prompt recall survey has been applied as a compensation to obtain the respondent-confirmed agendas, in the meantime, it involves much additional efforts and potential human errors (Bonsall, et al., 2011; Feng and Timmermans, 2013). Therefore, in practice, a well-performed imputation algorithm extracting the activity-travel data which are highly closed to the reality is extremely important.

Several algorithms have been reported in the literatures to extract activity and travel information from GPS traces. Algorithms vary from the informal ad-hoc approaches to advanced machine learning methods, such as neural networks, support vector machines, and Bayesian belief networks (Stopher and Wargelin, 2010; Moiseeva, et al., 2010; Rudloff and Ray, 2010). The machine learning algorithms have been long recognized as a promising method to replace the traditional ad hoc rules because the latter can be difficult in the situation of the increasing number of rules and the problem complexity. Unlike the ad hoc rules, machine learning methods are flexible in providing more accurate predictions through a learning process.

Basically, the imputation of activity episodes and transportation modes is in general a classification issue where most of machine learning algorithms should apply. Although a few different algorithms have been applied in the past for GPS data imputation, there are scarce comparisons of the success of these different methods. Among many of the existing research related to GPS data imputation, one of the exceptions is conducted by Bonsall et al. (2010), who presented a system for collecting and profiling commuter data. Although the main purpose of the paper is rather than the comparison of the performances of different algorithms, four methods were presented comparatively by using the results of the percentage of correctly detected modes. However, discussions on the performance of different algorithms were not addressed.

Therefore, in this paper we evaluate the relative performance of different algorithms for GPS data imputation. Seven different algorithms are considered, including the naive Bayesian (NB), Bayesian Network (BN), logistic regression (LR), multilayer perception (MP), support vector machine (SVM), decision table (DT) and C4.5 (C45). The imputation output includes most of the available transportation modes and the activity episode. Imputation results of various methods are compared using a sample of GPS data collected in The Netherlands. The overall error rates and hit ratios are adopted to assess their relative performance.

The remainder of the paper is organized as follows: Section 2 represents the different algorithms we are going to investigate and the configuration settings for each algorithm. Section 3 will then briefly describe the GPS data and sample selection. Section 4 will present the results in details for different algorithms with a purpose of comparison. Section 5 will conclude this paper and points out some future research potentials.

ALGORITHMS

In general, the imputation of activity episodes and transportation modes can be simulated as a nonlinear problem where many algorithms for classification should be applicable. Here, without loss of generality, we selected seven types of different algorithms for comparison. The algorithms vary in different theoretical bases, including the Bayesian inference, regression model, neural network, support vector machine, rule-based inference and decision trees. A full list of adopted algorithms is shown in Table 1.

Table 1 List of algorithms and parameter settings

Id	Algorithms
1	Bayesian Network (BN)
2	Naive Bayesian (NB)
3	Logistic regression (LR)
4	Multilayer Perception (MP)
5	Decision Table (DT)
6	Support Vector Machine (SVM)
7	C4.5 (C45)
8	CART (CART)

Naive Bayesian

A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. A naive Bayes classifier considers all these features to contribute independently to the probability.

Abstractly, the probability model for a classifier is a conditional model over a dependent class variable *C* with a small number of outcomes or classes, conditional on several feature variables F_1 through F_n.

$$\mathrm{P}(C|F_1, \dots, F_n) \tag{1}$$

The problem is that if the number of features *n* is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. Therefore a more tractable model can be reformulated using the Bayes theorem as follows:

$$\mathrm{P}(\mathrm{C}|F_1, \dots, F_n) = \frac{P(C)P(F_1,\dots,F_n)}{P(F_1,\dots,F_n)} \tag{2}$$

Bayesian network

Bayesian (also called Belief) network (BN) is a graphical representation of probabilistic causal information incorporating sets of probability conditional tables. It can be treated as an enhanced naïve Bayesian model by relaxing the assumption of independent distribution in that BN considers the joint probability of an attribute with its parent attributes, while the naive Bayesian assume all variables are independent. Thus a BN represents all factors deemed potentially relevant for observing a particular outcome, indicating that with BN it is possible to articulate expert beliefs about the dependencies between different variables.

The network is represented as a directed graph, together with an associated set of probability tables. In our case, the Bayesian network measures the interrelationship between spatial and temporal factors (input), and activity-travel pattern (output), i.e. transportation modes and activity episode. As shown in Figure 1, all the input variables are considered as the child nodes of the MODE. The parameter is estimated by using the maximum likelihood method when the network structure is determined.

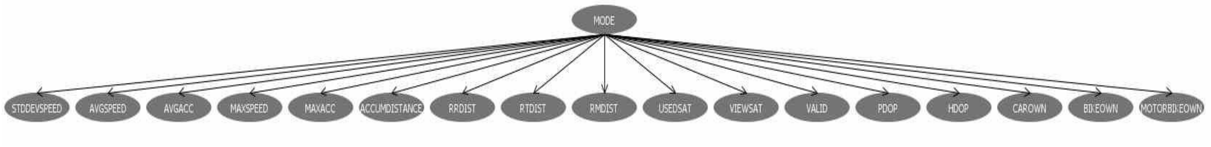


Figure 1 Structure of the Bayesian net

Logistic regression

Logistic regression is a type of regression analysis used for predicting the outcome of a categorical dependent variable based on one or more predictor variables. The probabilities describing the possible outcome of a single trial are modeled as a function of explanatory variables using a logistic function. In the past, different types of models have been developed as an extension of the basic logistic regression model. The multinomial logistic regression model is such a model which generalizes logistic regression by allowing more than two discrete outcomes. That is, it is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables.

Assumed there are *k* classes for *n* instances with *m* attributes, the probability for class *j* with the exception of the last class is

$$P_j(x_i) = \frac{\exp(x_i b_j)}{\sum_{j=1}^{k-1} \exp(x_i b_j) + 1} \tag{3}$$

The last class has probability

$$1 - \sum_{j=1}^{k-1} P_j(x_i) = \frac{1}{\sum_{j=1}^{k-1} \exp(x_i b_j) + 1} \tag{4}$$

The (negative) multinomial log-likelihood is thus:

$$\mathrm{L} = -\sum_{i=1}^{n} \sum_{j=1}^{k-1} \left(Y_{ij} \ln \left(P_j(x_i) \right) \right) + \left(1 - \sum_{j=1}^{k-1} Y_{ij} \right) \cdot \ln(1 - \left(\sum_{j=1}^{k-1} P_j(x_i) \right) + ridge \cdot B^2 \tag{5}$$

The *ridge* is a parameter which needs to be given in advance in the log-likelihood function. In order to find the matrix *B* for which *L* is minimized, a Quasi-Newton Method is used to search for the optimized values of the *m*(k-1)* variables.

Multilayer perception

A multilayer perception (MP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate output. An MP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MP utilizes a supervised learning technique called back-propagation for training the network.

Due to the fact the neural network with one hidden layer is in principle able to simulate all types of nonlinear problems we set one hidden layer in the network model. The activation function used the sigmoid function, as follows

$$\emptyset(y_i) = \frac{1}{(1+e^{-v}i)} \tag{6}$$

where y_i is the output of the i^{th} node (neuron) and v_i is the weighted sum of the input synapses.

Since the weights are obtained through an iterated calculation process, some parameters is Regarding setting the network training, we set the momentum and learning rate as 0.2 and 0.3, respectively. The training time was set as 500, which means the calculation stops when the number of epoch reaches 500 times.

Decision table

A decision table is a two-dimensional table that shows the action to be taken following a series of related decisions. In general, a decision table is composed of rows and columns, presented as a matrix. Each column corresponds to a single rule, with the rows defining the conditions and actions of the rules.

In the aspect of searching the best combinations, different algorithms apply. Here, we use an algorithm which searches the space of attribute subsets by greedy hill-climbing augmented with a backtracking facility. The performance of attribute combinations used in the decision table is evaluated through the overall root mean squared error and the accuracy of different classes.

Support vector machine

Support vector machines (SVM) are supervised learning models with associated learning algorithms that analyse data and recognize patterns. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, a SVM training algorithm builds a model that assigns new examples into one category or the other. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Here, we used a sequential minimal optimization algorithm to train a support vector classifier. The algorithm globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes, which results in the coefficients in the output based on the normalized data rather than original data. Multi-class problems are solved using pairwise classification. To obtain proper probability estimates, the option that fits logistic regression models to the outputs of the support vector machine is used. In the multi-

Table 2 Attribute variables for GPS data imputation

	Variable names	Content
Input	STDDEVSPEED	Standard deviation of speed
	AVGSPEED	Average speed
	AVGACC	Average acceleration
	MAXSPEED	Maximum speed
	MAXACC	Maximum acceleration
	ACCUMDISTANCE	Accumulated distance
	RRDIST	Distance to road line
	RTDIST	Distance to tram line
	RMDIST	Distance to metro line
	USEDSAT	Number of used satellites
	VIEWSAT	Number of viewed satellites
	VALID	GPX fix type
	PDOP	Position accuracy of 3d coordinate
	HDOP	Horizontal accuracy of 2d coordinate
	CAROWN	Yes if the respondent has a car, no otherwise
	BIKEOWN	Yes if the respondent has a bike, no otherwise
	MOTORBIKEOWN	Yes if the respondent has a motorbike, no otherwise
Output	MODE	Activity episode, train, walk, bike, car, bus, motorbike, running, tram, metro

class case the predicted probabilities are coupled using Hastie and Tibshirani's pairwise coupling method.

C45

C4.5 builds decision trees from a set of training data using the concept of information entropy. The training data is a set $S = s_1, s_2, \dots, s_n$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, \dots, x_{p,i})$, where the x_j represent attributes or features of the sample, as well as the class in which s_i falls.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recuses on the smaller sub-lists.

DATA

The data used in this paper are mainly collected from a small group of individuals as reported in a pilot study (Anastasia, et al., 2010). Eight individuals living in Eindhoven, The Netherlands, carried the GPS logger Bluetooth A+ during 6-8 weeks. In addition, to include more transportation modes in applications, we collected the activity and travel data specifically for the trips by tram and metro in the city of Rotterdam. Two colleagues contributed to this data collection and kept a very detailed diary of location and time data. In total, there are 53258 data points for model calibration and validation.

The GPS devices were configured to record data in every 3 seconds. The recorded information include: date, time, longitude, latitude, speed, distance, accuracy of the measurement (like PDOP, HDOP, etc.), and number of satellites. To impute transport modes in a certain time period, the three seconds epoch data were averaged within a time window. Furthermore, some additional statistic indicators were generated as to be the input variables of the prediction models. We incorporated variables relevant to speed, spatial distance to networks, accuracy of the GPS log measurement, and personal profiles. A detailed list of variables is shown in Table 2.

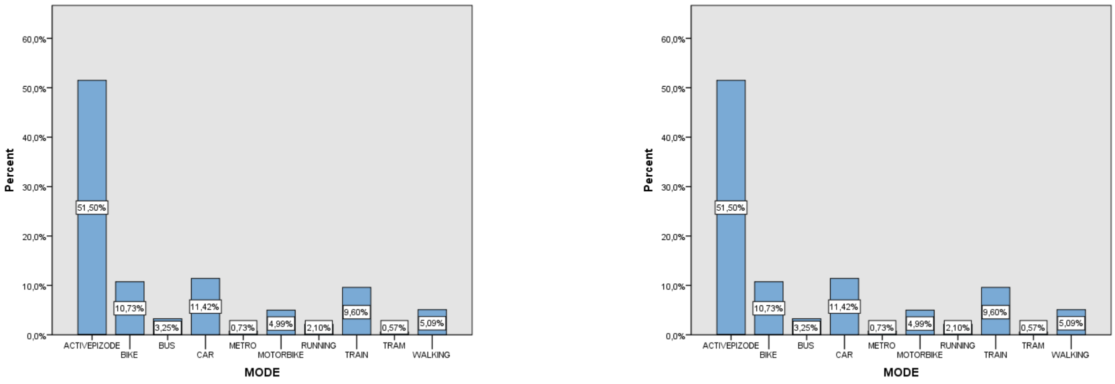
To make sure the comparison among different algorithms is reasonable in the sense of consistency, we use the same input variables and output variables for all algorithms. In concrete, for each model, there are eighteen input variables and one output variable which is named as MODE which has ten types of transportation modes and the activity episode.

The data were divided into two sets: training data and test data. We apply an algorithm to ensure selecting a random sub-sample from the whole dataset. Here, we set the selection criteria as non-replacement. More specifically, the algorithm was applied only once to randomly draw 75% of the full samples as the training dataset. Then the training dataset was used to compare with the full dataset, where those which are not selected yet (the left 25%), are taken as the test dataset. Moreover, before applying these algorithms, both of the training and test datasets are organized in a random order. Table 3 shows the partitions of the sample.

Table 3 Selection of training and test datasets

	Count	Percentage
Training data	39,942	75%
Test data	13,316	25%
Total	53,258	100%

Figure 2 shows the distribution of the transportation modes and the activity episode for the training data and the test data. As you can see that the two datasets have the same composition of different transportation modes and activity episodes.



(a) Training data

(b) Test data

Figure 2 Distribution of transportation modes and the activity episode

RESULTS AND ANALYSES

In order to evaluate the relative performance of different algorithms, we compare different indicators in terms of training data and test data, respectively. For each dataset, we use the indicators of the correctly classified instances (CCI), incorrectly classified instances (ICI) and Kappa value (Kappa). We use the software of Weka to implement these different methods (Hall et al., 2009).

Correctly identified instances and Kappa values

Table 3 presents the details of the prediction accuracy. As expected that for all the methods, the CCI for test data are lower than that for training data. All models, except for the NB, result into a Kappa value higher than 0.9.

In the case of training data, C45 and BN got a higher level of CCI (99.825% and 99.805%) than others. The Kappa values for C45 and BN are also quite high, 0.997 and 0.998, respectively. The kappa statistic measures the agreement of prediction with the true class, with 0 and 1 signifies incomplete and complete agreement, respectively. This means both of the two models, BN and C45, yield high prediction accuracy for the training data.

Level of CCI for the test data shows that BN has the highest score (99.474%) and NB has the lowest score (86.684%). Different from the results of training dataset that both BN and NB have a similar level of CCI, the CCI of the test dataset for C45 is 99.309%, slightly lower than BN. This probably indicates the BN has a stronger capability than C45 in the aspect of prediction generalisation.

Apart from the prediction accuracy, the levels of the complexity of different models also differ. In our cases, the NB model results in a simple network structure which can be easily represented. While the C45 algorithm results into complicated decision trees, with 214 leaves and 413 trees in total.

Among all the algorithms, the lowest level of CCI is from NB (86.966%), with the Kappa value as 0.822. In addition, the LR model (94.865%) results into a similar level of CCI to the SVM model (94.667%) for training data. The DT model (98.886%) has a higher level of CCI than the MB (97.118%) model for training data. These comparative conclusions also apply for the test data.

Hit ratios

The results of hit ratios show the prediction accuracy for each transportation mode and the activity episode. Table 5 presents the results for the training data. It can be found that the BN predicts the transportation modes of bike, motorbike, tram and metro with a 100% of correctness, with others equal to or higher than 99.7%. The level of the hit ratio of BN model is comparable with other methods.

The C45 and the DT also obtain a high accuracy regarding the level of hit ratios, but not as same as that of the BN model. Other models seem to have obvious falling down of the accuracy for different transportation modes. For example, the NB model has a low rate for Bike mode (0.799) and the SVM has low rates for the Running mode (0.654) and the Walking mode (0.76). In addition, both the LR (0.758) and the MP (0.743) lost the accuracy in the prediction of Bus mode, and the LR also did not predict well for the Running mode (0.76).

Table 6 presents the results for the test data. Some similar conclusions for above comparative analyses can be also found from this result. In addition, the comparison between the hit ratios for each transportation mode and the activity episode in the training data and the test data shows some interesting results. It can be found that the hit ratios of all transportation modes and the activity episode for the test data do not have to be lower than that for the training data, except for the models of BN and C45. This indicates that, in real predictions, the models of BN and C45 may perform more stable than others.

Table 3 Prediction accuracy and model performance

Algorithms	Training Data			Test Data		
	CCI (%)	ICI (%)	Kappa	CCI (%)	ICI (%)	Kappa
BN	99.805	0.195	0.997	99.474	0.526	0.993
NB	86.966	13.034	0.822	86.648	13.352	0.818
LR	94.865	5.135	0.926	94.510	5.490	0.921
MP	97.118	2.882	0.958	96.816	3.184	0.954
DT	98.886	1.114	0.984	98.100	1.900	0.973
SVM	94.667	5.333	0.923	94.458	5.542	0.920
C45	99.825	0.175	0.998	99.309	0.691	0.990

Table 4 Hit ratios for training data by transportation mode and activity episode

	A	B	C	D	E	F	G	H	I	J
BN	0.997	0.997	0.999	1	0.999	0.999	1	0.999	1	1
NB	0.848	0.969	0.934	0.799	0.836	0.926	0.949	0.98	1	0.983
LR	0.989	0.991	0.818	0.928	0.891	0.758	0.947	0.76	1	1
MP	0.998	0.974	0.916	0.926	0.965	0.743	0.989	0.985	1	1
DT	0.999	0.971	0.958	0.985	0.979	0.99	0.991	0.974	0.982	0.98
SVM	0.987	0.999	0.76	0.925	0.876	0.888	0.971	0.654	1	1
C45	1	0.999	0.993	0.997	0.997	0.994	0.998	0.999	0.996	0.99

Note: A-Activity episode; B-Train; C-Walking; D-Bike; E-Car; F-Bus; G-Motorbike; H-Running; I-Tram; J-Metro

Table 5 Hit ratios for test data by transportation mode and activity episode

	A	B	C	D	E	F	G	H	I	J
BN	0.996	0.993	0.988	0.997	0.994	0.977	0.999	1	1	0.983
NB	0.849	0.964	0.942	0.789	0.826	0.9	0.946	0.963	1	0.975
LR	0.99	0.994	0.815	0.915	0.882	0.733	0.935	0.752	1	1
MP	0.998	0.976	0.896	0.926	0.962	0.708	0.987	0.974	1	1
DT	0.998	0.948	0.939	0.973	0.97	0.973	0.982	0.963	0.892	0.959
SVM	0.987	0.998	0.763	0.931	0.869	0.844	0.968	0.641	0.985	1
C45	0.998	0.998	0.974	0.992	0.987	0.98	0.991	0.956	1	0.992

Note: A-Activity episode; B-Train; C-Walking; D-Bike; E-Car; F-Bus; G-Motorbike; H-Running; I-Tram; J-Metro

CONCLUSIONS

Imputation of GPS data for extracting the activity-travel data has been an important issue in the data collection using the technique of ICT. Various algorithms including the traditional ad hoc rules and machine learning algorithms have been developed, but few of them addressed the performance of different algorithms. Assessing the difference of these classifications algorithms in the context of GPS data imputation can provide theoretical bases for the predication capability of different methods and contribute to the selection of a well-performed algorithm in real applications.

Since the comparison in a more general sense in a different research field does not apply for the GPS data, in this paper, we evaluated the relative performance of different imputation algorithms for GPS data imputation by incorporating the naive Bayesian, Bayesian network, logistic regression, multilayer perception, support vector machine, decision table and C4.5. The accuracy of imputation results of various methods are compared using the GPS data collected in The Netherlands.

Results show that the Bayesian network has a better performance than other algorithms according to the correctly identified instances and Kappa values for both training data and test data. Especially, the Bayesian network shows a stronger capability than other methods in the aspect of prediction generalisation. In addition, the BN resulted into a higher level of hit ratios for all transportation modes and the activity episode than other methods.

Comparison of the hit ratios between the training data and the test data showed that the hit ratios of all transportation modes and the activity episode for the test data do not have to be lower than that for the training data, except for the models of BN and C45. This indicates that, in real predictions, the models of BN and C45 may perform more stable than others.

It should be noted that each of the methods included in this paper has a potential to be further improved and provides better results in that the outputs might be influenced by the settings of different parameters. However, such difference should not be too big in principle. As an alternative, it might be useful to further check on the sensitivity of the algorithms for GPS data imputation. Moreover, future research may include more methods and more training and test data set. More importantly, examination on the sequence data of individuals in combination of the predictions in the epoch is an interesting direction.

REFERENCES

Bonsall, P., Schade, J., Roessger, L. and Lythgoe, B. (2011) Can we believe what they tell us? factors affecting people¡¦s engagement with survey tasks. International Steering Committee for Travel Survey Conferences, Chile, 2011.

Feng, T. and Timmermans, H.J.P. (2013) Analysis of Error in Prompted Recall Surveys. The XII NECTAR International Conference, 16-18 June, 2013, São Miguel Island, Azores, Portugal.

Feng, T., Moiseeva, A. and Timmermans. H.J.P. (2011) Processing of National Travel Survey GPS Pilot Data: A Technical Report Prepared for the Department for

Transport. A Technical Report prepared on behalf of the Department of the Transport, UK, 2011.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, 11(1).

Moiseeva, A., Jessurun, J. and Timmermans, H.J.P. (2010) Semiautomatic Imputation of Activity Travel Diaries. Transportation Research Record: Journal of the Transportation Research Board, 2183, 60-68.

Rudloff, C. and Ray, M. (2010) Detecting Travel Modes and Profiling Commuter Habits Solely Based on GPS Data. Transportation Research Board, January 10, Washington D.C.

Stopher, P.R. and Wargelin, L. (2010) Conducting a household travel survey with GPS: Reports on a pilot study. Proceedings of the 12th WCTRS, July 11-15, 2010, Lisbon, Portugal.