Neurocomputing 167 (2015) 3–7

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling

Fabio Moretti [a,b,*], Stefano Pizzuti [a,b], Stefano Panzieri [b], Mauro Annunziato [a]

[a] *ENEA (Italian Energy New technologies and sustainable Economic development Agency)*
[b] *Università degli Studi "Roma Tre", Computer Science and Automation Department*

ARTICLE INFO

Article history:
Received 7 November 2013
Received in revised form
16 July 2014
Accepted 11 August 2014
Available online 12 May 2015

Keywords:
Traffic flow forecasting
Ensembling
Bagging
Neural networks

ABSTRACT

In this paper we show a hybrid modeling approach which combines Artificial Neural Networks and a simple statistical approach in order to provide a one hour forecast of urban traffic flow rates. Experimentation has been carried out on three different classes of real streets and results show that the proposed approach outperforms the best of the methods it puts together.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Transportation is a wide human-oriented field with diverse and challenging problems waiting to be solved. Characteristics and performances of transport systems, services, costs, infrastructures, vehicles and control systems are usually defined on the basis of quantitative evaluation of their main effects. Most of the transport decisions take place under imprecision, uncertainty and partial truth. Some objectives and constraints are often difficult to be measured by crisp values. Traditional analytical techniques were found to be ineffective when dealing with problems in which the dependencies between variables were too complex or ill-defined. Moreover, hard computing models cannot deal effectively with the transport decision-makers' ambiguities and uncertainties. In order to come up with solutions to some of these problems, over the last decade there has been much interest in soft computing applications of traffic and transport systems, leading to some successful implementations [1]. The use of soft computing methodologies for modeling and analyzing traffic and transport systems is of particular interest to researchers and practitioners due to their ability to handle quantitative and qualitative measures, and to efficiently solve complex problems which involve imprecision, uncertainty and partial truth. Soft computing can be used to bridge modeling gaps of normative and descriptive decision models in traffic and transport research. Transport problems can be classified into four main areas: traffic control and management, transport planning and management, logistics, design and construction of transport facilities. The first category includes traffic flow forecasting which is the topic tackled in this work. This issue has been faced by the soft computing community since the nineties [4–10] up today [12–14,11] with Artificial Neural Networks (ANNs) [2,3]. As example, among the most recent work [14] focuses on traffic flow forecasting approach based on Particle Swarm Optimization (PSO) with Wavelet Network Model (WNM). Pamula et al. [11] review neural networks applications in urban traffic management systems and presents a method of traffic flow prediction based on neural networks. Bucur et al. [12] propose the use of a self-adaptive fuzzy neural network for traffic prediction suggesting an architecture which tracks probability distribution drifts due to weather conditions, season, or other factors. All the mentioned applications have one feature in common: they use one single global model in order to perform the prediction. Therefore, the main novelty of the proposed work is to combine different heterogeneous models in order to get a meta-model capable of providing predictions more accurate than the best of the constituent models. In our work we firstly composed of a neural networks ensemble with a simple statistical model and compare the results over the one hour forecast, then we improved ensembling model with BAGGING. Results shown highlight a remarkable decrease of error through the BAGGING learning phase.

* Corresponding author.
E-mail addresses: fabio.moretti@enea.it (F. Moretti), stefano.pizzuti@enea.it (S. Pizzuti), stefano.panzieri@uniroma3.it (S. Panzieri), mauro.annunziato@enea.it (M. Annunziato).

http://dx.doi.org/10.1016/j.neucom.2014.08.100
0925-2312/© 2015 Elsevier B.V. All rights reserved.

2. Methods

2.1. Basic model

In order to perform a meaningful comparison for the forecasting, a basic model should be introduced in order to quantify the improvement given by more intelligent and complex forecasting techniques. For seasonal data a basic model might be defined as

$$x_t = x_{t-s} \tag{1}$$

with S being the appropriate seasonality period. This model gives a prediction at time *t* presenting the value observed exactly a period of S steps before. For this work we put the value of S = 1 which corresponds to the previous hour. It means that to predict the flow rate of the following hour it is used the current flow measure.

2.2. Statistical

One of the simplest and most widely used models when dealing with regular time series (as urban traffic flows) is to build an average weekly distribution of the traffic flow sampled hourly. Thus, from the data we compute for each day the average flow rate hour by hour in such a way that we get an average distribution made of 24.7 = 168 points.

2.3. Neural network ensembling

Models ensemble is a technique where many prediction models cooperate on the same task. The aggregation of multiple prediction of the same variable may lead to better results and generalization than using a single model prediction. In order to increase generalization capability, the model learning phase is crucial. The goal obtains better predictive performance than could be obtained from any of the constituent models. In the last years, several ensembling methods have been carried out [17,15,16]. The first one, also known as Basic Ensemble Method (BEM), is the simplest way to combine M neural networks as an arithmetic mean of their outputs. This method can improve the global performance [20,21] although it does not take into account that some models can be more accurate than others. This method has the advantage to be very easy to apply. A direct BEM extension is the Generalised Ensemble Method (GEM) in which the outputs of the single models are combined in a weighted average where the weights have to be properly set, sometimes after an expensive tuning process. Bagging (Bootstrap AGGregatING) [18] technique improves generalization: for each learner replaces part of the training data set with a random combination of training data itself. Thus each dataset may contain duplicated entries of the same sample or not at all. Improvement occurs especially when small changes in dataset may lead to a large changes in prediction. Adaboosting [19] introduces weights on the training points.

2.4. Hybrid model

Hybrid models are an extension of the ensembling approach in the sense that the final goal is to combine different models in such a way that the accuracy of the composition is higher than the best of the single models. The difference is that the combination is performed among highly heterogeneous models, that is models generated by different methods with different properties and thus the composition among them is a complex rule taking into account the peculiarities of the models and/or of the problem itself. Therefore, in this work we propose a novel hybrid model which combines an ANN ensemble with the statistical model. The composition rule is the following : "IF the statistical model has a high error (meaning that for some reason we are out of a normal situation) THEN use the neural model ELSE use the statistical one". This criterion is based on the absolute error of the statistical model, thus the composition rule turns into

$$|x^t - y^t| > \epsilon \Rightarrow y^{t+1} = y_n^{t+1} \tag{2}$$

$$|x^t - y^t| \le \epsilon \Rightarrow y^{t+1} = y_s^{t+1} \tag{3}$$

where y^{t+1} is the outcome (one hour prediction) after the composition rule, y_n^{t+1} is the prediction of the neural ensemble, y_s^t is the current outcome of the statistical model and y_s^{t+1} is its prediction. This basically means that if we are in normal statistical conditions (where the statistical model makes a small error) then use as prediction model the statistical one (which is very accurate in this condition), else (when out of normal statistical situations) take the neural ensembling estimation.

3. Experimentation

In this paragraph we test and compare the methods presented in the previous section. The test case has concerned the short term traffic flow rate of three different streets, shown in Table 1, located in the town of Terni (about 90 km north of Rome). The data set is made of 3 months (13 weeks) of measurement corresponding to 2184 hourly samples.

The data set has been partitioned into training/testing and validation made respectively of 10 and 3 weeks each. We firstly present the result obtained using hybrid model based on Neural Network Basic Ensemble model and statistic model and we show an improvement on the forecasting, then we replace Basic Ensemble Model with a Bagging based one. Results show a further improvement on the forecasting.

3.1. Neural network setup

The ANN are feed-forward MLP with 10 hidden neurons and one output (the one hour flow forecast) with sigmoid as activation function for all the neurons. The number of inputs *N* has been

Table 1
Street parameters.

Street	Maximum traffic flow rate
Street 1	600
Street 2	800
Street 3	950

Table 2
History length selection.

N (h)	Street 1	Street 2	Street 3
3	5.72	6.88	5.81
5	3.90	5.07	3.99
8	3.29	3.43	3.02
10	3.54	4.12	3.74

Table 3
Hybrid model parameter ϵ tuning. Errors percentage of hybrid model at different values of ϵ parameter.

Street	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 30$	$\epsilon = 40$	$\epsilon = 50$	$\epsilon = 60$
Street 1	2.98	2.83	2.81	2.80	2.88	2.99
Street 2	2.85	2.69	2.65	2.66	2.68	2.75
Street 3	3.25	3.13	3.08	3.04	3.03	3.04

Download English Version:

https://daneshyari.com/en/article/406264

Download Persian Version:

https://daneshyari.com/article/406264

Daneshyari.com