Transportation Research Part C 43 (2014) 143–157

Contents lists available at ScienceDirect

Transportation Research Part C

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

Short-term traffic volume forecasting: A *k*-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm

Zuduo Zheng *, Dongcai Su

Civil Engineering & Built Environment School, Science and Engineering Faculty, Queensland University of Technology, 2 George St, GPO Box 2434, Brisbane, Qld 4001, Australia

ARTICLE INFO

Article history:
Received 28 February 2013
Received in revised form 26 October 2013
Accepted 13 February 2014

Keywords:
Short-term traffic volume forecasting
KNN
Kalman filter
Principle component

ABSTRACT

To enhance the performance of the *k*-nearest neighbors approach in forecasting short-term traffic volume, this paper proposed and tested a two-step approach with the ability of forecasting multiple steps. In selecting *k*-nearest neighbors, a time constraint window is introduced, and then local minima of the distances between the state vectors are ranked to avoid overlappings among candidates. Moreover, to control extreme values' undesirable impact, a novel algorithm with attractive analytical features is developed based on the principle component. The enhanced *KNN* method has been evaluated using the field data, and our comparison analysis shows that it outperformed the competing algorithms in most cases.

1. Introduction

As an indispensable component of intelligent transportation systems, short-term traffic volume forecasting (*STTVF*) has received enormous attentions over the past two decades. Consequently, many *STTVF* algorithms were developed using different approaches from various perspectives. Specifically, in terms of modeling, these algorithms are either parametric or non-parametric: the former explicitly and quantitatively formulates the relationship between the input and the output (the forecasted) via a parameterized function (model), while the latter is fully data driven and explores the implicit relationship between the forecasted data and input data without providing any well-defined function.

Implementing a parametric algorithm typically consists of two basic steps: estimating the parameters and forecasting the output by inputting new data into the calibrated model. Although a rich family of parametric *STTVF* algorithms with promising performances was developed in the literature (Ahmed and Cook, 1979; Levin and Tsao, 1980; Okutani and Stephanedes, 1984; Hamed et al., 1995; Williams et al., 1998; Williams and Hoel, 2003; Stathopoulos and Karlaftis, 2003; Xie et al., 2007), they inherently face model calibration, validation, and computational challenges, which makes them difficult to be implemented in real-time transportation systems. For example, although good performance of seasonal autoregressive integrated moving average model (*SARIMA*) was frequently reported (Williams and Hoel, 2003; Ghosh et al., 2005; Williams et al., 1998; Chung and Rosalion, 2001), estimating the parameters of *SARIMA* is quite computationally demanding even in the case of univariate, as indicated in Smith et al. (2002).

* Corresponding author. Tel.: +61 07 3138 9989; fax: +61 07 3138 1170.
E-mail address: zuduo.zheng@qut.edu.au (Z. Zheng).

http://dx.doi.org/10.1016/j.trc.2014.02.009
0968-090X/

On the other hand, non-parametric *STTVF* algorithms are also extensively studied and their good performances are often reported (Smith and Demetsky, 1997; Zhang et al., 1998). Compared with parametric *STTVF* algorithms, main advantages of non-parametric algorithms include: intuitive formulation, totally data-driven and thus free of assumptions on data distribution, high flexibility and easy extendibility (Clark, 2003). For example, k-nearest neighbor (*KNN*) algorithms can be easily extended to handle multivariate by simply adding data from multiple locations into the search space. More importantly, nonparametric algorithms are theoretically grounded. As an asymptotically optimal forecaster, when applied to a state space with m members, *KNN* approach will asymptotically be at least comparable to any mth order parametric model (Smith et al., 2002). Motivated by this attractive property, there is a steady stream of refining and extending *KNN* in the literature. This paper is along this line.

Most existing *KNN* algorithms are single-step (Smith and Demetsky, 1997; Smith et al., 2002; Davis and Nihan, 1991), which has two main disadvantages: (i) generating overlapping nearest neighbors when the method is extended to multiple-step forecasting as demonstrated later; (ii) sensitive to noisy neighbors. To remedy these serious limitations, this study develops an enhanced *KNN* algorithm (i.e., *KNN-LSPC*) with the ability of forecasting multiple steps. We have evaluated the algorithm's performance using loop detector data. Our analysis shows that the enhanced *KNN* algorithm outperformed the competing algorithms in most cases.

Note that for the convenience of discussion, this paper focuses on short-term volume forecasting. However, the algorithm can be easily adapted for forecasting other traffic flow measures (e.g., speed). The remaining of the paper is organized as follows. Section 2 defines the *STTVF* problem, and then introduces *KNN*; Section 3 presents the enhanced *KNN* algorithm; Section 4 evaluates the enhanced *KNN* algorithm's performance; Finally, Section 5 summarizes the main findings and discusses future research.

2. Background

2.1. Problem description

Without loss of generality, we define *STTVF* as follows:

For a given traffic volume series $\{vol^{(j)}(t), t_0 \leqslant t \leqslant t_c, 1 \leqslant j \leqslant m\}$, where t_0, t_c are the indices of the beginning and the current time intervals (note that for simplicity, time interval will be shortened as time unless it is otherwise stated), m is the number of locations (i.e., the number of loop detectors), and $vol^{(j)}(t)$ represents the traffic volume collected from the jth loop detector at time t, the problem is to forecast volume L steps ahead for the target location (denoted as j^*). More specifically, our task is to forecast the following vector $f(t)$:

$$f(t) = vol^{(j^*)}(t_c + 1 : t_c + L)$$

with $f(t, i) = vol^{(j^*)}(t_c + i), 1 \leqslant i \leqslant L$ represents its ith element. Note that we use the notation $vol(a{:}b)$, $b > a$, $a, b \in Z^+$ to denote a sub-time series from time a to time b of a traffic volume time series. For notational simplicity, we denote $vol(t) = vol^{(j^*)}(t)$ herein.

2.2. KNN algorithms

Like other data-driven approaches, *KNN* algorithm's performance is dependent on the representativeness and extensiveness of the data. The fundamental assumption of *KNN* algorithms is that future states to be forecasted are more or less similar to a neighborhood of the past. Smith et al. (2002) provided an excellent review on *KNN* forecasting algorithms.

A typical *KNN* framework consists of three basic elements: defining the state vector; measuring distance between two state vectors; and forecasting future state vectors by utilizing a collection of k-nearest neighbors (candidates).

For *STTVF*, a typical state vector can be defined as:

$$x(t) = [vol(t-2), vol(t-1), vol(t)] \tag{1}$$

The nearness of a state vector to another is commonly measured by the Euclidean distance, according to which neighbors are ranked and selected. Out of k neighbors that are nearest to the current state vector, future states can be forecasted using various methods. The simplest forecasting approach is to directly compute the average of the k nearest neighbors, while more sophisticated approach in the literature generate forecasts by weighting the k nearest neighbors according to their distances to the current state vector.

For illustration purpose, the *KNN* algorithms developed in Smith et al. (2002) are summarized below. These *KNN* algorithms are also used as part of the benchmark models in evaluating the new algorithm's performance.

The *KNN* forecasting algorithms proposed in Smith et al. (2002) were designed to predict one step ahead and its state vector and forecasted vector at time t are defined as in Eqs. (2) and (3), respectively.

$$x(t) = [vol(t-2), vol(t-1), vol(t), vol_{hist}(t), vol_{hist}(t+1)] \tag{2}$$

$$f(t) = vol(t+1) \tag{3}$$