Transportation Research Part C 72 (2016) 168–181

Contents lists available at ScienceDirect

Transportation Research Part C

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

An efficient realization of deep learning for traffic data imputation

Yanjie Duan, Yisheng Lv *, Yu-Liang Liu, Fei-Yue Wang

The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

A R T I C L E I N F O

Article history:
Received 28 April 2016
Received in revised form 27 September 2016
Accepted 27 September 2016

Keywords:
Traffic data imputation
Deep learning
Missing data

A B S T R A C T

Traffic data provide the basis for both research and applications in transportation control, management, and evaluation, but real-world traffic data collected from loop detectors or other sensors often contain corrupted or missing data points which need to be imputed for traffic analysis. For this end, here we propose a deep learning model named denoising stacked autoencoders for traffic data imputation. We tested and evaluated the model performance with consideration of both temporal and spatial factors. Through these experiments and evaluation results, we developed an algorithm for efficient realization of deep learning for traffic data imputation by training the model hierarchically using the full set of data from all vehicle detector stations. Using data provided by Caltrans PeMS, we have shown that the mean absolute error of the proposed realization is under 10 veh/5-min, a better performance compared with other popular models: the history model, ARIMA model and BP neural network model. We further investigated why the deep leaning model works well for traffic data imputation by visualizing the features extracted by the first hidden layer. Clearly, this work has demonstrated the effectiveness as well as efficiency of deep learning in the field of traffic data imputation and analysis.

1. Introduction

Traffic data are essential and fundamental for both transportation research and applications (Wang, 2010; Nan et al., 2008). Many institutes have established systems (Zhong et al., 2005), e.g. Caltrans (California Department of Transportation) PeMS (Performance Measurement System) (PeMS, 2014), to store and analyze traffic data like flow, occupancy and speed. With traffic data, engineers and researchers can understand and assess the performance of traffic systems, discover existent transportation problems, and make better decisions (Chen and Bell, 2002). However, traffic data collected from real traffic systems often have corrupted or missing data points (Li et al., 2014; Smith and Babiceanu, 2004) due to detector and communication malfunctions, adversely affecting traffic management (Sharma et al., 2004). Thus traffic data imputation is required in traffic data collection and storage.

Generally, traffic data imputation is to estimate the corrupted or missing traffic data. Because of the necessity of traffic data imputation, the complex patterns of traffic data and the diversity of application scenarios, there have been many researchers investigating this problem using a wide range of methods (Ferndez-Moctezuma et al., 2007; Tang et al., 2015; Tan et al., 2013; Castrillon et al., 2012; Zhang and Liu, 2009). These methods are mainly classified into three categories: prediction, interpolation and statistical learning (Li et al., 2014). Prediction methods typically use historical data collected

* Corresponding author.
E-mail addresses: duanyanjie2012@ia.ac.cn (Y. Duan), yisheng.lv@ia.ac.cn (Y. Lv), liuyuliangxp@126.com (Y.-L. Liu), feiyue@ieee.org (F.-Y. Wang).

http://dx.doi.org/10.1016/j.trc.2016.09.015
0968-090X/

from one site to build a prediction model and predict the values of corrupted or missing data points for the same site (Zhong and Sharma, 2006; Li et al., 2015; Gan et al., 2015). The autoregressive integrated moving-average (ARIMA) model (Nihan, 1997) is one of the commonly used methods to predict the imputed data points one by one. Interpolation methods replace the corrupted or missing data points with history data points or neighboring data points. One straightforward model is to fill a corrupted or missing data point with a known historical data collected at the same site at the same daily time interval but in previous days (Allison, 2001). This model is also known as the history model. Models using neighboring data points to interpolate corrupted or missing data points often use the data of neighboring sites or neighboring states to estimate the values of corrupted or missing data points for the current site. In these models, each corrupted or missing data point is estimated by the average or the weighted average of the neighboring data (Chen et al., 2003; Al-Deek et al., 2004; Van Lint et al., 2005; Kim and Lovell, 2006). A typical one is k-NN (Liu et al., 2008; Chang et al., 2012), of which the key work is to determine the neighbors by an appropriate distance metric. Statistical learning methods often use the observed data to learn a scheme, then inference the corrupted or missing data points in an iterated fashion (Smith et al., 2003; Wang et al., 2008; Qu et al., 2009; Li et al., 2013). A classical method is the Markov Chain Monte Carlo (MCMC) multiple imputation method (Ni and Leonard, 2005; Ni et al., 2005; Steimetz and Brownstone, 2005; Farhan and Fwa, 2013). The basic idea of MCMC multiple imputation method is to treat a corrupted or missing data point's value as a parameter of interest, and estimate the parameter by drawing a series of samples of the parameter. That means the imputation of the corrupted or missing data point is a combination of multiple imputed values instead of only one value. Neural network (Zhong et al., 2004; Ming et al., 2004; Lv et al., 2015) is a promising method to obtain better imputation performance than traditional imputation methods given more observed data. In addition, some researchers use traffic simulation models such as DynaSmart (Mahmassani et al., 1992), DynaMIT (Ben-Akiva et al., 1998), Vissim (Fellendorf, 1994), Paramics (Cameron and Duncan, 1996), TransWorld (Wang, 2010) to perform traffic data imputation (Muralidharan et al., 2009; Muralidharan and Horowitz, 2009). Whatever method a specific traffic data imputation realization adopts, the key idea is utilizing the potential temporal and spatial information in the data.

With the increasing quantity of traffic data, imputing corrupted or missing data points automatically and efficiently has become essential and critical. For this end, here we propose a deep learning model named denoising stacked autoencoders (DSAE) for traffic data imputation. This model treats traffic data containing observed normal data points, corrupted or missing data points as a corrupted vector, and transforms traffic data imputation into clean data recovering or corrupted data denoising. We firstly evaluate the impact of temporal and spatial factors on traffic data imputation, and then propose a hierarchically training algorithm to efficiently build the model. This algorithm can take advantage of inherent information embedded in large scale data collected from a traffic network and achieve better performance.

The rest of this paper is organized as the following: Section 2 describes the denoising stacked autoencoders. Section 3 presents the realization schemes for traffic data imputation based on DSAE, and discusses the model performance on solving random corruption. Section 4 further explores the model performance on solving continuous corruption. Section 5 concludes this paper.

2. Methodology

Deep learning has been successfully applied in image classification, speech recognition, natural language processing and computer gaming (LeCun et al., 2015; Silver et al., 2016; Wang et al., 2016). In this paper, we propose a novel deep learning model named denoising stacked autoencoders (DSAE) and would like to see how effective DSAE are at imputing traffic flow data. DSAE has two basic blocks including autoencoders (AEs) (Bengio et al., 2007) and denoising autoencoders (DAE) (Vincent et al., 2008). An AE can extract features from original input data. AEs can be stacked to form a deep network to obtain an abstract representation of the input in a gradual feature extraction way. A DAE is a stochastic version of the AE and can capture statistical dependencies between the inputs. Also DAEs can be connected to form a deep network named stacked denoising autoencoders (SDAE) (Vincent et al., 2010). SDAE has been proved to be able to learn inherent features and correlations in data and extract useful higher level representations. Both DAE and SDAE have the ability of cleaning or denoising data. Here, for the purpose of traffic data imputation we combine a DAE playing the role of corrupted traffic data denoising with stacked AEs helping extract features from high dimension traffic data to build a deep learning model named DSAE which retains the advantages of both DAE and stacked AEs. DSAE recovers data through feature extraction and statistical dependency learning. The following paragraphs introduce AE and DAE, and illustrate the construction of a DSAE.

2.1. AE

An AE is structured by its encoder part and decoder part as shown in Fig. 1. The encoder $f\theta$ maps an input vector $\mathbf{x}$ into hidden representation $\mathbf{h}$, i.e. $\mathbf{h} = f_\theta(\mathbf{x})$. f_θ is typically a nonlinear transformation in the following form:

$$f_\theta(\mathbf{x}) = s(\mathbf{x}\mathbf{W}^T + \mathbf{b}), \tag{1}$$