Low-dimensional Models for Compression, Compressed Sensing, and Prediction of Large-Scale Traffic Data

Nikola Mitrovic, *Student Member*, *IEEE*, Muhammad Tayyab Asif, *Student Member*, *IEEE*, Justin Dauwels, *Senior Member*, *IEEE* and Patrick Jaillet

***Abstract*—Advanced sensing and surveillance technologies can collect traffic information from various sources with high temporal and spatial resolution. Recorded data is essential for many real-time applications related to traffic management systems. However, the volume of the data collected severely limits the scalability of these systems for large networks. In this study, we consider the problems of compression, estimation and prediction in the context of large and diverse networks. Although methods such as principal component analysis (PCA) can efficiently compress traffic data sets, the low-dimensional models created by these methods are not readily interpretable. In this study, we propose an alternative approach to compress the recorded data and enhance the scalability of traffic management systems. We compress the network by representing it in terms of a small subset of the road segments present in the network. This formulation allows us to efficiently store collected data in an intuitive way. Furthermore, we utilize the compressed representation to estimate the current state of the network by collecting data from a small subset of road segments. Similarly, we perform traffic prediction for the whole network, by developing prediction models for only the representative subset of road segments. For the analysis, we consider a large network comprising 17,967 road segments. Numerical results show that our method can achieve competitive compression performance compared to PCA. Results further demonstrate significant reduction in prediction time without significant degradation in prediction performance.**

***Index Terms*—Low-dimensional models, prediction in large networks.**

I. Introduction

Intelligent Transportation Systems (ITS) collect real-time traffic information from various sources such as probe vehicles, smartphone devices and infrastructure based traffic sensors. With advancements in sensor technology, traffic data (e.g., volume and speed) can be recorded on a large scale and with high temporal resolution [1]. Recorded data is frequently used for historical analysis and traffic management operations. ITS use advanced compression techniques to efficiently store recorded data for historical traffic analysis [2]. Furthermore, ITS utilize simulation models and data driven techniques to perform tasks such as network monitoring, transportation planning and congestion avoidance [3]. These applications heavily rely on fast and accurate estimation of current and future network states.

Principal component analysis (PCA) is commonly deployed to compress large traffic data sets [2], [4]–[6]. PCA provides an effective low-dimensional representation in terms of latent variables. The latent variables may be hard to interpret since they represent a linear combination of (up to all) original dimensions [7]. In order to apply PCA, each link in the traffic network needs to be monitored. For large traffic networks this approach may not be possible. By contrast, we wish to consider a more practical option where only a subset of links is monitored.

Simulation (model) driven approaches are traditionally used to assess the current and future traffic states [8]. These approaches are deployed for traffic management operations at various levels of network granularity [8]–[10]. For large areas, macroscopic and mesoscopic simulation tools (e.g., DYNAMIT) have been used to build custom models, relying on historical speed-density link relationships for that specific network [10]. In recent years, high volumes of recorded data have served for extensive calibration of traffic dynamics and consequently have enhanced the performance of model-driven approaches. However, such models are not generic and cannot be translated from one network to another in straightforward manner. An alternative is to consider data driven methods. Data-driven methods offer greater flexibility due to their generic structure. Furthermore, these methods can be used to develop highly accurate traffic prediction and estimation models. The sheer volume of data combined with large network size, however, can limit the scalability of both data-driven and model-driven techniques [11]. Nonetheless, traffic conditions on different roads within the network tend to exhibit varying level of correlations. This trend can be seen even for large and diverse networks. We propose to utilize these relationships to obtain low-dimensional representations for traffic networks. Specifically, we use these low-dimensional representations to perform traffic estimation and prediction.

There are various data mining and machine learning techniques which have been applied for prediction of traffic data [12]–[15]. In all existing studies, these techniques

Nikola Mitrovic, Muhammad Tayyab Asif and Justin Dauwels are with the School of Electrical and Electronic Engineering, College of Engineering, Nanyang Technological University, Singapore, 639798; nikola001@e.ntu.edu.sg, muhammad89@e.ntu.edu.sg, jdauwels@ntu.edu.sg.

Patrick Jaillet is with the Department of Electrical Engineering and Computer Science, School of Engineering, and also with the Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is also with the Center for Future Urban Mobility, Singapore-MIT Alliance for Research and Technology, Singapore (e-mail:jaillet@mit.edu).

The research described in this project was funded in part by the Singapore National Research Foundation (NRF) through the Singapore MIT Alliance for Research and Technology (SMART) Center for Future Mobility (FM).

explicitly predict traffic variable at each link in the observed network [14], [16], [17]. These studies often consider small traffic networks or a few highways, and do not scale well to larger networks and real-time applications.

In order to handle large-scale traffic networks in real-time settings, we propose a low-dimensional representation of the network, explicitly expressed in terms of its traffic links (road segments). Unlike PCA, the proposed method based on a column based (CX) decomposition requires a small number of representative road segments to be monitored. We apply the CX decomposition to express the original network in terms of a small subnetwork. The latter is carefully selected so as to represent the entire original network. We refer to the small subnetwork as the compressed state of the original network. We also use the CX method to learn relationship between compressed and original (uncompressed) network by considering the recorded data. In this way, we can represent the collected data as a product of two low-rank matrices: (i) the subnetwork data matrix; (ii) the corresponding relationship matrix, inferred from recorded data set. We refer to this approach as compression.

We further use this low-dimensional representation to infer the present state of the large network from the current traffic state of the subnetwork. We refer to this approach as compressed sensing [18]. Unlike compression, compressed sensing requires training and testing data sets. We use the training data set to infer the relationship matrix. To assess the network state we multiply the subnetwork data matrix, obtained from the testing set, with the relationship matrix, inferred from the training data set. We rely on the observation that the traffic conditions tend to follow distinct patterns and that traffic variables often vary smoothly across the traffic network.

Similarly, we apply the CX method to infer the future state of the entire network. To this end, we first select the representative subnetwork, and learn its relationship with the large network. Then, we explicitly predict traffic parameters for the subnetwork. At last, we perform extrapolation to estimate the future state of the whole network. We refer to this approach as compressed prediction.

For our analysis, we consider the city-scale traffic network in Singapore, comprising $17,967$ road segments. The results show that the proposed methods can accurately compress the large traffic data. Further, our compressed methods can infer the current (estimation) and future (prediction) states of the network, while substantially reducing the processing speed of the underlying modeling algorithm. The reduction is proportional to the compression ratio, i.e., the ratio of the number of links in the subnetwork and the total number of links.

The paper is structured as follows. In Section II we briefly introduce the column based (CX) matrix decomposition method. In Section III we present three applications of the CX matrix decomposition methods in the realm of traffic modeling: compression, compressed sensing, and compressed prediction. In Section IV we describe the traffic data set analyzed in this paper, and introduce several performance measures to assess the proposed and existing algorithms. In Section V, we provide numerical results for our experiments and discuss further improvements. In Section VI we summarize our contributions and suggest topics for future research.

II. Column based (CX) decomposition

First let us briefly review the column based (CX) matrix decomposition:

Definition 1: Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a given matrix. Let $\mathbf{C} \in \mathbb{R}^{m\times c}$ be a matrix consisting of c columns of the matrix $\mathbf{A}$. The column-based (CX) matrix approximation $\hat{\mathbf{A}}$ of $\mathbf{A}$ is defined as $\hat{\mathbf{A}} = \mathbf{CX}$, where $\mathbf{X} \in \mathbb{R}^{m\times n}$ is a matrix that expresses every column of $\mathbf{A}$ in terms of the basis provided by the columns of $\mathbf{C}$ [19].

A. Column selection

In order to select the best subset of columns, for a given size c, one could test all possible combinations. However, the computational complexity of this brute-force approach is $O(n^c)$ [20]. Due to this complexity, testing all possible choices of c columns is typically not practical. Therefore, the selection of a subset of columns is still an interesting open problem [19]–[21]. To alleviate this problem, several randomized algorithms have been proposed [19], [21]. We briefly review the algorithm proposed by Drineas *et al.* [19]. Consider a set $\Theta_A = \{\mathbf{a}_i\}_{i=1}^n$ containing all the columns of matrix $\mathbf{A}$. The algorithm then selects c columns from Θ_A as follows. The algorithm calculates the Euclidean norm of top k right singular vectors of matrix $\mathbf{A}$ to assign score to each column [19]. This score is then converted into a probability and further used to sample the columns. The resulting reconstruction error can be upper bounded as follows [19]:

Theorem 1: *Given a matrix* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and an integer* k$\ll$*min*{m,n}*, there exist a randomized algorithm that selects exactly* $c = O(k^2 \log(1/\delta)/\varepsilon^2)$ *columns of* $\boldsymbol{A}$ *to construct* $\boldsymbol{C}$*, such that with probability at least* $1-\delta$,

$$\min \parallel A - CX \parallel_F = \parallel A - CC^+A \parallel_F \leq (1+\varepsilon) \parallel A - A_k \parallel_F, \quad (1)$$

where $\parallel \mathbf{A} \parallel_F = \left(\sum_i \sum_j a_{i,j}^2\right)^{\frac{1}{2}}$ is the Frobenius norm of the matrix $\mathbf{A}$. $\mathbf{C}$ contains the sampled columns of $\mathbf{A}$, $\mathbf{CC}^+\mathbf{A}$ is the projection of $\mathbf{A}$ on the subspace spanned by the chosen columns and ε is the acceptable deviation. $\mathbf{A}_k$ is the best rank-k approximation to $\mathbf{A}$, obtained by singular value decomposition (SVD). δ is the failure probability in the case of randomized algorithms. The algorithm performs column sampling $O(\log(\frac{1}{\delta}))$ times and returns the subset of columns such that $\parallel A - CC^+A \parallel_F$ is smallest over all $O(\log(\frac{1}{\delta}))$ trials [19].

Sun *et al.* proposed a random sampling algorithm by assigning a score to each column based on the Euclidean norm of the column [21]. The scores can be normalized to obtain the probability of selection for each column. This method significantly reduces the computational complexity and space requirements for sparse matrices such as the origin-destination (OD) internet traffic data [21].

In our application, we consider three sampling strategies. In the first strategy, we assign identical probability to each column of the matrix ($P_r(a_i) = \frac{1}{n}$). We refer to it as random sampling. In the second strategy we follow [21] and assign sampling probabilities based on the Euclidean norm of the column of matrix **A**. The assigned probability of the column is proportional to the energy (L2 norm) of that column (2). We refer to it as energy sampling method:

$$P_e(a_i) = \frac{\|\mathbf{a}_i\|_2^2}{\|\mathbf{A}\|_F^2} \quad \forall \; i = 1, ...n, \tag{2}$$

In the third strategy, we follow [19] and assign a weight to each column in proportion to the Euclidean norm of the top k right singular vectors of matrix **A** [19], [20], [22]. To approximate k we use the number of columns (links) that needs to be selected:

$$P_s(a_i) = \frac{1}{c}\sum_{j=1}^{c} v_{ij}^2 \quad \forall \; i = 1, ...n. \tag{3}$$

We refer to this scheme as SVD sampling method where v_{ij} is the i-th coordinate of j-th right singular vector. To compute the right singular vectors, we perform a singular value decomposition (SVD) of the original data matrix **A**:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T. \tag{4}$$

Since the matrix **V** is unitary, the quantities $P_s(a_i)$ sum to one. The computational complexity of the SVD sampling method is $O(n^3)$ [19], which can be too expensive for extremely large matrices [21]. For the energy sampling method, it is also $O(n^3)$ as it requires the computation of the Frobenius norm of the matrix. In contrast, probability assignment in random sampling takes $O(1)$ time.

B. Relationship matrix

For the sampled column matrix $\mathbf{C} \in \mathbb{R}^{m\times c}$, we compute the relationship matrix $\mathbf{X} \in \mathbb{R}^{c\times n}$, which will allow us to represent the columns of matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ in terms of columns of the matrix **C** [19]. The matrix **X** can be regarded as an extrapolation matrix that maps the subnetwork associated with **C** to the entire network represented by **A**.

For given matrices **C** and **A**, we compute the matrix **X** as:

$$\mathbf{X} = \mathbf{C}^{+}\mathbf{A}, \tag{5}$$

where $\mathbf{C}^{+}$ is Moore-Penrose pseudo-inverse of matrix **C** [23].

III. CX METHOD FOR TRAFFIC APPLICATIONS

In this section, we develop CX based methods for three different traffic applications: compression, compressed sensing, and compressed prediction of speed data. For this purpose, we consider the traffic data in the form of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ where the columns of the matrix $\{\mathbf{a}_i\}_{i=1}^{n}$ contain traffic data from different roads $\{s_i\}_{i=1}^{n}$. Rows represent time instances $\{t_i\}_{i=1}^{m}$ at which the traffic data is recorded. Each matrix cell (a_{ij}) shows the numerical value of an observed traffic variable (e.g., speed, volume) at location s_j during the interval of time $(t_i - T, t_i)$ where T is the sampling period (e.g., 5 or 15 minutes). Therefore, the i-th row vector $\alpha_i = [z(s_1,t_i)...z(s_n,t_i)]$ of **A** contains the observed traffic variable for the entire network at a particular time t_i. Similarly, the j-th column vector $\mathbf{a}_j = [z(s_j,t_1)...z(s_j,t_m)]^T$ of **A** contains the observed variable at location s_j during the entire recording period. Therefore, we can write traffic data matrix as $\mathbf{A} = [\mathbf{a}_1 \; ... \; \mathbf{a}_n]$. For the sake of simplicity, we use subscripts h, p and f in the rest of the paper to denote historical, present and future values, respectively.

A. Compression

In this section we explain how the column based decomposition method can be used for compression of recorded traffic data. Suppose that the matrix $\mathbf{C}_h$ contains the observed traffic parameter at c specific locations in the network, such that $\{\mathbf{c}_1,...,\mathbf{c}_c\} \subseteq \{\mathbf{a}_1,...,\mathbf{a}_n\}$. Then, we can approximate the data matrix $\mathbf{A}_h$ as

$$\hat{\mathbf{A}}_h = \mathbf{C}_h\mathbf{X}_h, \tag{6}$$

where the matrix $\mathbf{X}_h$ contains the relationships between the traffic parameter at different locations in the network. Hence, instead of storing the large matrix $\mathbf{A}_h$, we store the two smaller matrices $\mathbf{C}_h$ and $\mathbf{X}_h$. The compression ratio (CR) of such low-dimensional approximation is given by:

$$CR_h = \frac{mn}{mc + cn}, \tag{7}$$

where the numerator equals the total number of elements in the original (uncompressed) matrix $\mathbf{A}_h$. The denominator in (7) represents the total number of elements in the low-dimensional (compressed) approximation. It is noteworthy that the matrix **X** has to be stored for the compression of traffic data. Consequently, the compression ratio differs from the ratio of the total number of links (n) and the number of links in compressed network state (c).

The proposed method is easy to interpret. Unlike Principal Component Analysis (PCA), the proposed method stores the observed traffic parameter at specific locations in the network instead of as a linear combination of latent variables. Furthermore, a CX-compression scheme identifies the important links in the network and reveals the relationships between these links and the rest of the network, leading to practical benefits for traffic management.

B. Compressed sensing

So far, we have assumed that the matrix $\mathbf{X}_h$ is stored together with $\mathbf{C}_h$ leading to the compression of matrix $\mathbf{A}_h$. In this scenario, the matrix $\mathbf{X}_h$ is computed for a given data matrix $\mathbf{A}_h$ and a column matrix $\mathbf{C}_h$. Alternatively, one may precompute a matrix $\mathbf{X}_h$ and re-use the same matrix to infer **A** from any given **C**. We refer to that scenario as compressed sensing. The underlying assumption is that the traffic conditions are stationary, so that a fixed matrix **X** allows us to accurately reconstruct the original data matrix **A** from **C** [5], [6], [24]–[26]. Therefore, we can estimate the present network state ($\hat{\alpha}_p$) as:

$$\hat{\alpha}_p = \mathbf{c}_p\mathbf{X}_h, \tag{8}$$

where $\hat{\alpha}_p \in \mathbb{R}^{1\times n}$ is a row vector which represent the current state of the entire network. Row vector $\mathbf{c}_p \in \mathbb{R}^{1\times c}$ contains the information about current traffic conditions at c specific locations in the network. Matrix $\mathbf{X}_h$ is the relationship matrix, learned from a training data set. We define the compression ratio for compressed sensing CR_p as $(\frac{n}{c})$.

Large traffic networks contain diverse set of road segments. We want to explore whether homogeneous subnetworks can improve the overall performances of compressed sensing. We divide the traffic network into s mutually exclusive subnetworks such that $\alpha_p = [\alpha_1 ... \alpha_s]$ where $\alpha_i \in \mathbb{R}^{1\times n_i}$ $\forall$ $i = 1,...s$. Then, we perform compressed sensing for each subnetwork separately. At last, we merge the results of the clustered subnetworks in order to infer the traffic state of the entire network. Although different choices of temporal and/or spatial clustering can be applied, we only consider simple clustering based on different road categories in this study.

The overall performance of the proposed compressed sensing method is sensitive to the "compressibility" of the network and "non-stationarity" in the traffic data. For compression, we represent the traffic data as a product of two low-rank matrices, i.e., the subnetwork data matrix and the most appropriate relationship matrix. As the compression is lossy, we expect the reconstructed matrix $\hat{\mathbf{A}}_h$ to be different from the original matrix $\mathbf{A}_h$ (see (6)). The issue of non-stationarity is due to the fact that the matrix $\mathbf{X}_h$ is inferred from training (historical) traffic data instead of the current data. Indeed, in the setting of compressed sensing, data is only available at a subset of links corresponding to the matrix $\mathbf{C}_p$, where $\mathbf{C}_p = [\mathbf{c}_p^1 ... \mathbf{c}_p^m]^T$. The matrix $\mathbf{A}_p$ ($\mathbf{A}_p = [\alpha_p^1 ... \alpha_p^m]^T$) is not available, and the goal is to infer that matrix by extrapolating the matrix $\mathbf{C}_p$ according to the CX decomposition. Obviously, the matrix $\mathbf{X}_p$ cannot be extracted from the current data $\mathbf{A}_p$, since the matrix $\mathbf{A}_p$ is not available. Instead we determine $\mathbf{X}_h$ from training data set. Since traffic is not perfectly stationary, this approximation will induce an additional reconstruction error. We refer to it as the error due to non-stationarity of traffic spatial relationships. To quantify this error, let us call $\mathbf{B} = \mathbf{C}_p\mathbf{X}_p$ the reconstruction of the data matrix $\mathbf{A}_p$, assuming the latter is available to compute the CX decomposition. The reconstruction $\hat{\mathbf{A}}_p$ ($\hat{\mathbf{A}}_p = [\hat{\alpha}_p^1 ... \hat{\alpha}_p^m]^T$) in the scenario of compressed sensing is less accurate, since we need to replace $\mathbf{X}_p$ (determined from the test data matrix $\mathbf{A}_p$) by $\mathbf{X}_h$ (determined from training data matrix $\mathbf{A}_h$). The mean squared error (MSE) incurred for compressed sensing can be written as:

$$\frac{1}{mn}\|\mathbf{A}_p - \hat{\mathbf{A}}_p\|_F^2 = \frac{1}{mn}\|(\mathbf{A}_p - \mathbf{C}_p\mathbf{X}_p) - (\mathbf{C}_p\mathbf{X}_h - \mathbf{C}_p\mathbf{X}_p)\|_F^2, \quad (9)$$

$$= \frac{1}{mn}\|(\mathbf{A}_p - \mathbf{B}) - (\hat{\mathbf{A}}_p - \mathbf{B})\|_F^2, \quad (10)$$

$$= \frac{1}{mn}\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - b_{ij})^2 + \sum_{i=1}^{m}\sum_{j=1}^{n}(\hat{a}_{ij} - b_{ij})^2 - 2\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - b_{ij})(\hat{a}_{ij} - b_{ij})\Big), \quad (11)$$

where the first component of the error corresponds to the compressibility of the network and the second component is due to the non-stationarity of spatial patterns within the network. To make this interpretation more explicit, we rewrite (11) as:

$$\mathrm{MSE}_{\mathrm{est}} = \mathrm{MSE}_{\mathrm{com}} + \mathrm{MSE}_{\mathrm{ns}} - 2\xi_{\mathrm{est}}. \quad (12)$$

We will analyze the behavior of these errors for different compression ratios in Section V.

C. Compressed prediction

In the previous section, we inferred the condition of the entire traffic network from observations at a small subset of links. Here we will extend this approach to prediction; we aim to predict the state of the entire traffic network from past and present observations at a small subset of links. We predict the traffic parameter (e.g. speed) for a selected subset of locations, and then extrapolate the predictions to the rest of the network:

$$\hat{\alpha}_f = \hat{\mathbf{c}}_f\mathbf{X}_h, \quad (13)$$

where $\hat{\mathbf{c}}_f \in \mathbb{R}^{1\times c}$ is the row vector containing the predicted values of the traffic conditions at the selected locations, $\hat{\alpha}_f \in \mathbb{R}^{1\times n}$ contains the predictions for all locations, and $\mathbf{X}_h$ is the relationship matrix as in (8). The predictions in $\hat{\mathbf{c}}_f$ can be generated by means of any state-of-the-art prediction algorithm; we decided to use support vector regression, as that method performed well in our earlier studies [15], [27]. If the predictions $\hat{\mathbf{c}}_f$ would be identical to the true values $\mathbf{c}_f$, then the problem boils down to compressed sensing, which we discussed in the previous section. In practice, of course, the predictions have some inaccuracies. Therefore, we can write $\mathbf{c}_f = \hat{\mathbf{c}}_f + \Delta\mathbf{c}$ where $\Delta\mathbf{c}$ represents the prediction error. Furthermore, let $\mathbf{D} = \mathbf{C}_f\mathbf{X}_h$ be the estimated network profile, during the entire observational period, without any prediction error in $\mathbf{C}_f$ ($\mathbf{C}_f = [\mathbf{c}_f^1 ... \mathbf{c}_f^m]^T$). Consequently, the mean squared error (MSE) between predicted $\hat{\mathbf{A}}_f$ ($\hat{\mathbf{A}}_f = [\hat{\alpha}_f^1 ... \hat{\alpha}_f^m]^T$) and true future values $\mathbf{A}_f$ ($\mathbf{A}_f = [\alpha_f^1 ... \alpha_f^m]^T$) can be written as:

$$\frac{1}{mn}\|\mathbf{A}_f - \hat{\mathbf{A}}_f\|_F^2 = \frac{1}{mn}\|(\mathbf{A}_f - \mathbf{C}_f\mathbf{X}_h) - (\hat{\mathbf{C}}_f\mathbf{X}_h - \mathbf{C}_f\mathbf{X}_h)\|_F^2, \quad (14)$$

$$= \frac{1}{mn}\|(\mathbf{A}_f - \mathbf{D}) - (\hat{\mathbf{A}}_f - \mathbf{D})\|_F^2, \quad (15)$$

$$= \frac{1}{mn}\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - d_{ij})^2 + \sum_{i=1}^{m}\sum_{j=1}^{n}(\hat{a}_{ij} - d_{ij})^2 - 2\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - d_{ij})(\hat{a}_{ij} - d_{ij})\Big). \quad (16)$$

We refer to (16) as MSE for compressed prediction. We rewrite (16) in more explicit form:

$$\mathrm{MSE}_{\mathrm{total}} = \mathrm{MSE}_{\mathrm{est}} + \mathrm{MSE}_{\mathrm{pred}} - 2\xi_{\mathrm{pred}}. \quad (17)$$

By substituting (12) in (17), we obtain:

$$\mathrm{MSE}_{\mathrm{total}} = \mathrm{MSE}_{\mathrm{com}} + \mathrm{MSE}_{\mathrm{ns}} + \mathrm{MSE}_{\mathrm{pred}} - 2\xi_{\mathrm{est}} - 2\xi_{\mathrm{pred}}. \quad (18)$$

The total error of compressed prediction can be decomposed into four error components: (i) error due to compression; (ii) error due to changes in spatial relationships (non-stationarity); (iii) error due to inaccurate predictions; (iv) correlations among the previous error components.

The compressed prediction provides significant reduction in computational complexity by explicitly predicting the traffic variables for only a small subset of road segments in the network. Compressed prediction involves two computations: (i) prediction of the traffic variable at representative locations in the network and (ii) extrapolation of the predicted values to the entire network. In the former, the computational complexity depends on the underlying prediction algorithm, and is proportional to the number of locations c in the subnetwork. The second step (extrapolation) requires a single matrix-vector multiplication with complexity O(cn). Therefore, the total complexity is of order O(c+cn). In practice, the predictions at each link in the subnetwork are computationally complex. By contrast, the extrapolation can be executed much faster. Therefore, by performing prediction only for a small subnetwork, the computational complexity can be drastically reduced. The reduction is proportional to the compression ratio, i.e., the ratio of the total number of links and the number of links in the subnetwork.

IV. Experimental setup

In this section we explain the traffic data set considered in this study, and describe how we analyzed it for our different applications (compression, compressed sensing, and compressed prediction). We also briefly review support vector regression, which we use in our CX-based prediction approach. At last, we introduce several measures to assess the proposed methods.

A. Data set

We observe the nationwide traffic network in Singapore, comprising 57,747 road segments (links). The traffic network contains diverse types of roads, belonging to different categories (CAT A, B, ..., E). The variable of interest is the average traffic speed. The traffic speed is recorded every five minutes at each link in the network. Therefore, the reported speed represents the average speed of all vehicles which traverse a link during the given sampling interval. The Land Transportation Authority (LTA) of Singapore provided us speed data for a period of three months (August - October 2011). We represent the data set in the form of a matrix $\mathbf{A}_{\text{speed}} \in \mathbb{R}^{26,496\times 57,747}$, as explained in Section III.

The obtained data set contains missing values due to sensor malfunctions and other reasons. First, we remove the columns and the rows of the matrix $\mathbf{A}_{\text{speed}}$ which have more than 5% of missing values. Therefore, we only consider those links which have at least 95% of data during observational period and those time instances when at least 95% of network data is available. As a result, 17,967 road segments (columns) and 23,564 time instances (rows) are taken in consideration (see Fig.1). Most of the removed rows (time instances) belong to data from October. For the remaining matrix $\mathbf{A} \in \mathbb{R}^{23,564\times 17,967}$ we impute the missing values by following the procedure in [28].

We use matrix $\mathbf{A} \in \mathbb{R}^{23,564\times 17,967}$ to evaluate the performance of compression, for three different sampling schemes: random, energy, and SVD. Since the other two applications (compressed sensing and compressed prediction) require training for some parameters, we split the data set into training and testing subsets. As training set, we chose the speed data of the months August and September, 2011, representing 74% of all speed data. We use the training data set for three purposes: (i) to determine the subnetwork of c links, corresponding to the matrix $\mathbf{C}$; (ii) to learn relationships between the subnetwork, as defined in (i), and the entire network; (iii) to train the SVR predictors for the subnetwork defined in (i).

The testing set $\mathbf{A}_{test} \in \mathbb{R}^{6109\times 17,967}$ is composed of the speed data for the month of October 2011, representing 26% of all data. We use this data to evaluate the performance of compressed sensing and compressed prediction. For both applications, we use the relationship matrix $\mathbf{X}_h$ extracted from the training data set. For compressed prediction, we carefully predict traffic parameter at specific locations. Although any prediction algorithm can be applied, we use here support vector regression (SVR), which we briefly review.

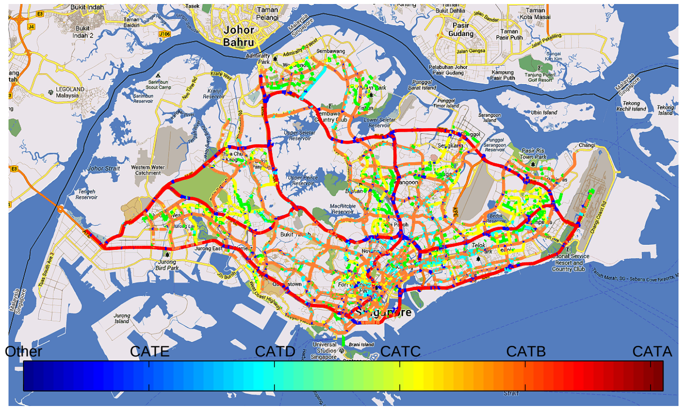


Fig. 1: City-scale network of Singapore with 17,967 road segments of different categories, from CAT A (freeways) to CAT E (local feeders).

B. Support Vector Regression

Support vector regression (SVR) is a data-driven prediction algorithm that is often applied for traffic data [14], [15], [29]. It efficiently uses current and past values of traffic parameter at a particular location in order to predict future value. Let us assume that we want to predict the future speed value y_{jk} at time $t_j + k\delta_t$ for link s_i. We write that as $y_{jk} = z(s_i, t_j + k\delta_t)$, where t_j is the current time. In our analysis, we define the corresponding input feature vector as $\mathbf{x}_j = [d(t_j),\ h(t_j),\ z(s_i,t_j)\ \ldots\ z(s_i,t_j - k\delta_t)]^T$ where $d(t_j)$ and $h(t_j)$ is the day and hour, respectively, of the particular t_j. $z(s_i,t_j)\ \ldots\ z(s_i,t_j - k\delta_t)$ are current and k past speed values for the particular link. We aim to infer relationship between y_{jk} and $\mathbf{x}_j$ such that $y_{jk} = f_k([d(t_j),\ h(t_j),\ z(s_i,t_j)\ \ldots\ z(s_i,t_j - k\delta_t)])$. We organize the training data set as r input-output pairs $\{(\mathbf{x}_j, y_{jk})\}_{j=1}^{r}$ which we use to train SVR and infer non-linear relationships.

SVR non-linearly maps the input speed data into higher dimensional feature space Φ [30], [31]. In order to avoid explicit mapping in Φ, SVR utilizes the Kernel trick. Kernel trick replaces dot products in the feature space by the relation $\kappa(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, where κ is the desired kernel function [31]. In our analysis, we consider Radial Basis Function (RBF) kernel $\kappa_{\mathbf{x}_i,\mathbf{x}_j} = \exp - \frac{\| \mathbf{x}_i - \mathbf{x}_j \|^2}{2\gamma^2}$ which is highly effective in mapping non-linear relationships [32]. Therefore, the function f_k will be [30], [31]:

$$f_k(\mathbf{x}) = \sum_{j=1}^{r} (\alpha_j - \alpha_j^*)\mathbf{x} \cdot \mathbf{x}_j + b. \tag{19}$$

where α_j, α_j^* are the Lagrange multipliers. We use (19) to train SVR and perform speed prediction. For SVR implementation we utilize matlab package LIBSVM [33]. In our analysis, we deploy ν-SVR for ν=1 and default values for other parameters [30], [33].

C. Performance measures

We now briefly explain the three performance measures considered in this paper: percent root mean distortion (PRD), mean absolute percentage error (MAPE), and mean squared error (MSE). We use percent root mean distortion (PRD) error to evaluate the reconstruction error of low-dimensional models [34]. The percent root mean distortion (PRD) quantifies the reconstruction error:

$$\mathrm{PRD}(\%) = \frac{\| \mathbf{A} - \mathbf{CX} \|_F}{\| \mathbf{A} \|_F}. \quad (20)$$

An alternative measure of the reconstruction error is the mean absolute percentage error (MAPE) defined as:

$$\mathrm{MAPE} = \frac{1}{mn} \sum_i \sum_j \frac{|a_{i,j} - \hat{a}_{i,j}|}{a_{i,j}}, \quad (21)$$

where $a_{i,j}$ and $\hat{a}_{i,j}$ are true and predicted value, respectively.

At last, the mean squared error (MSE) is defined as:

$$\mathrm{MSE} = \frac{1}{mn} \|\mathbf{A} - \mathbf{CX}\|_F^2. \quad (22)$$

V. Results

First we investigate the CX method for compression of traffic data. We apply three different sampling strategies: random, energy and SVD. For each strategy, we repeat sampling few times and report the average reconstruction accuracy. Our benchmark is Principal Component Analysis (PCA), as it is considered as the optimal linear transformation. Fig. 2 shows the compression performance of different methods and sampling schemes. As expected, PCA slightly outperforms the proposed method in terms of compression error. This is in agreement with Theorem 1, which relates the CX decomposition to SVD. However, the proposed CX-based approach is easier to interpret, and is also the stepping stone to compressed sensing and compressed prediction, as we will discuss in the following.

Let us also analyze the efficiency of different sampling schemes. Amongst the proposed sampling strategies, the so-called SVD scheme provides the best results. Fig. 3a depicts the assigned importance for the SVD sampling method. From that figure, it can be seen that the SVD scheme assigns higher selection probability to roads with large traffic speed variations. Fig. 3b shows the assigned importance for energy sampling method. Unlike the SVD sampling strategy, the energy scheme mostly relies on high-speed expressways segments and thereby it almost entirely ignores the variations across non-primary roads. The energy sampling method is more suitable for traffic parameters (e.g., number of OD trips) that simultaneously can exhibit large energy and variations [21].

The random sampling scheme yields similar performance to the SVD scheme, especially for high compression ratios (see Fig. 2). However, for low compression ratios, the SVD strategy outperforms the random sampling scheme.

As the SVD sampling method outshines the other two sampling schemes, we will only consider the SVD sampling method from now on. An important question is whether the SVD sampling scheme leads to subnetworks that are stable over time. To assess the stability of the subnetwork generated by the SVD sampling method, we applied this method to each of the three months (August, September, October, 2011) of traffic data separately. For each month, we sort the road segments in descending order according to the assigned probability by the SVD sampling method. Hence, the most representative roads are at the top of these lists. Next we select the first k links of each list, with k corresponding to 10%, 25%, and 50% of the links in the network. If the subnetwork is stable across time, the three short lists of top-k

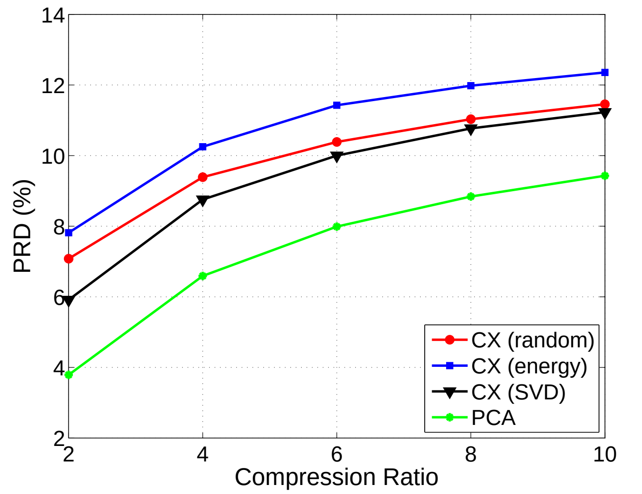


Fig. 2: Performance of CX method for compression and different sampling strategies. The SVD sampling scheme provides the best performance among the proposed strategies. Therefore, for compressed sensing and compressed prediction, we will consider SVD sampling scheme.

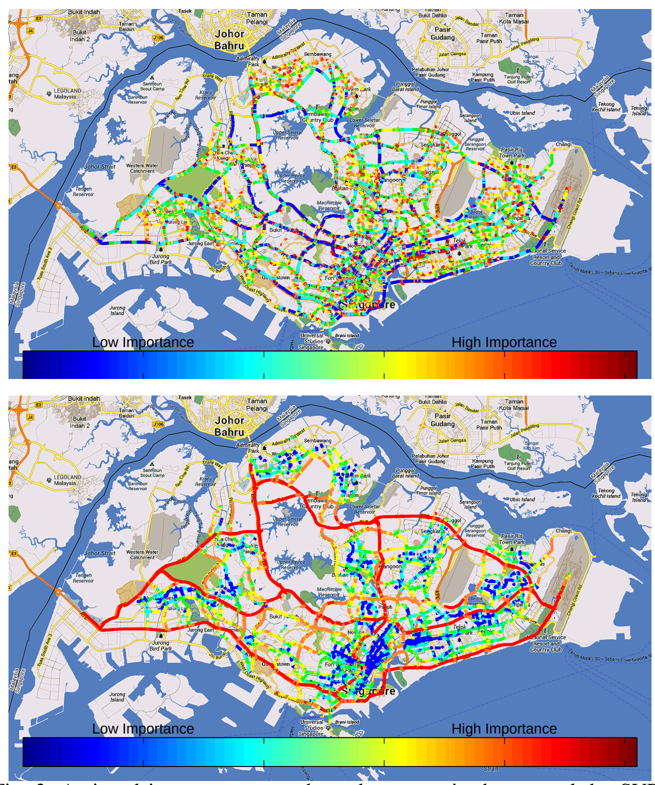


Fig. 3: Assigned importance to each road segment in the network by SVD (top) and the energy sampling method (bottom).

links should have many links in common. The results of this analysis are summarized in Table I, where the percentage of common links is provided. As can be seen from this Table, most links in the subnetwork are consistently selected for all three months, suggesting that the SVD sampling method results in a subnetwork that is stable over time.

Let us now explore the subnetworks obtained by SVD sampling. Fig. 4 shows the road categories of the whole network and selected subnetworks. Subnetworks mostly contain the road segments of lower hierarchy level ("Slip" and "Other"). These road segments exhibit high variations. On the other hand, selected subnetworks contain a small percentage of expressways i.e., "CAT A" roads (see Fig. 4). Nearby

	Aug.	Sep.	Oct.
Aug.	100	93.88	92.26
Sep.	93.88	100	92.82
Oct.	92.26	92.82	100

(a) k = 10% of all roads

	Aug.	Sep.	Oct.
Aug.	100	93.61	91.67
Sep.	93.61	100	92.14
Oct.	91.67	92.14	100

(b) k = 25% of all roads

	Aug.	Sep.	Oct.
Aug.	100	94.65	93.12
Sep.	94.65	100	93.49
Oct.	93.12	93.49	100

(c) k = 50% of all roads

TABLE I: Overlap (%) among the k links with the highest selection probability (calculated by the SVD sampling method) in the three months of data.

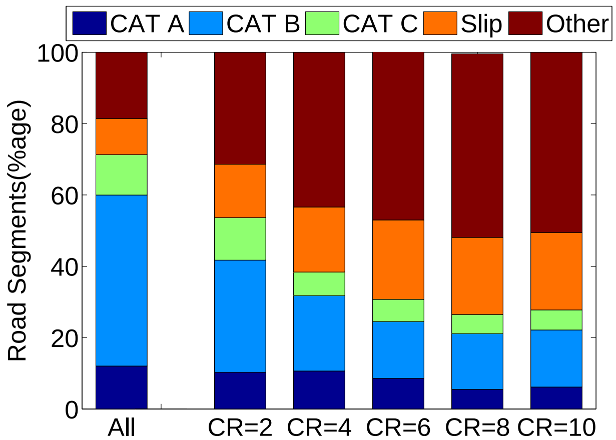


Fig. 4: Categories of road segments for the entire network (bar graph for "All") and selected subnetworks.

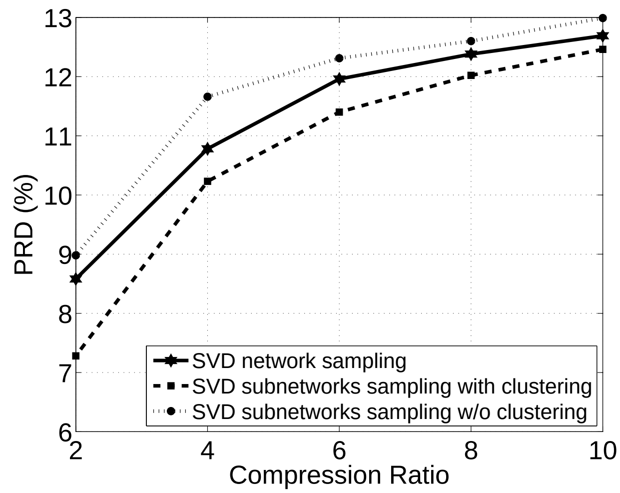


Fig. 5: Accuracy of compressed sensing method for different approaches and compression ratios.

	2	4	6	8	10
MSE (Comp)	0.01	2.18	7.27	11.95	15.93
MSE (Ns)	19.77	29.07	31.20	29.23	27.35
ξ(Corr)	0.00	0.00	0.00	0.00	0.00
Total MSE	19.78	31.25	38.47	41.18	43.28

TABLE II: MSE of the proposed method for application of compressed sensing and for different compression ratios.

expressway segments exhibit similar behavior and hence they can be modeled by fewer components, in comparison with the roads of lower hierarchy level which exhibit more irregular traffic patterns.

We now investigate the case of compressed sensing. We aim to reconstruct the average speed for the entire network from observations in a small subnetwork. The relationship matrix $\mathbf{X}_h$ is determined from the training set, and the reconstruction error is assessed on the test set. Fig. 5 shows the reconstruction accuracy of the proposed method for three different approaches: (i) we select the subset of road segments according to SVD sampling scheme (see solid line in Fig. 5); (ii) we cluster the network according to the category of the road. For each cluster, we select the subset of the road segments using SVD sampling scheme. Then, we perform compressed sensing for each cluster, separately (see dashed line in Fig. 5); (iii) we use the identical set of roads as defined in (ii) to perform network estimation. Unlike in (ii), we do not perform any clustering here (see dotted line in Fig. 5). Intuitively, the difference between (ii) and (iii) shows the gain obtained by network clustering. Fig. 5 indicates that applying the compressed sensing method to different road categories leads to better estimation performance for the entire network. As expected, the reconstruction accuracy of all three approaches increases with the size of subnetwork where traffic variable is explicitly observed. In the following, we investigate the error of compressed sensing in more details. In our analysis, we observe the subset of links as defined in (i).

The overall compressed sensing (estimation) error is caused by information loss due to compression of traffic data and changes in traffic behavior between training and testing periods. Table II shows the mean squared error (MSE) of the individual error components, the correlation between the two errors components and the total MSE, for different compression ratios. From that Table, it can be seen that the non-stationarity of the traffic data is the main contributor to the estimation error. As expected, the error associated with the compressibility of traffic data increases with the compression ratio. Table II shows that there is no correlation between compressibility and non-stationarity of traffic data. Information loss (compressibility) of upcoming data is not related to the changes (non-stationarity) in spatial relationship between past and new data (see (9)). Therefore, better compressed performance will lead to the lower estimation error.

The third application of the proposed CX method is traffic prediction. In compressed prediction, we use the future state of small subset of roads to predict future traffic condition for the whole network. We also consider the traditional approach where the speed for each road segment is explicitly predicted. For both approaches we use the identical SVR settings as explained in Section IV-B. Fig. 6 depicts the prediction accuracy of the proposed and traditional methods for different prediction horizons and various compression ratios. As expected, the compressed method has slightly larger PRD and MAPE errors than the traditional approach. This additional error decreases if the portion of the network increases where the traffic speed is explicitly predicted. Also, the additional error decreases for large prediction horizons (see Fig. 6). Naturally, it is difficult to predict for

	2	4	6	8	10
MSE (EST)	19.78	31.25	38.47	41.18	43.28
MSE (SVR)	33.73	28.23	24.15	20.95	18.61
ξ(Corr)	5.80	6.01	5.52	4.83	4.42
Total MSE	41.91	47.46	51.58	52.47	53.05

TABLE III: The MSE error of the proposed method for application of compressed prediction and for different compression ratios.

larger horizons even with traditional approach. Hence, in such cases the error due to prediction tends to become the dominant component.

Fig. 7 shows the prediction performance for different road categories for the (proposed) compressed and (traditional) uncompressed prediction method. As expected, the expressways ("CAT A") are the easiest to predict. Interestingly, the absolute additional error associated with the CX-approach seems to be similar for most road categories.

We decompose the mean squared error (MSE) of compressed prediction into estimation and prediction components. Table III shows the contribution of these two error components as well as the correlation between them. From Table III, we can see that there is some correlation between the two error components. Minor changes in traffic behavior between training and testing periods would lead to higher estimation accuracy (see (14)). At the same time, the future state of subnetwork is easily predicted due to regular traffic behavior. Similarly, diverse traffic behavior during testing period would cause higher error for both estimation and prediction components. From Table II and III, we can assess the contributions of compressibility, non-stationarity and predictability of traffic data to the MSE of compressed prediction. As can be seen from Table III, the estimation error increases with the compression ratio. This increase in estimation error is mainly due to non-stationarity error component (see Table II).

The proposed approach of compressed prediction provides substantial reduction in computational complexity by explicitly predicting the variables at a small representative set, followed by (fast and efficient) extrapolation to the entire network. This reduction in computational time is obtained at the expense of a small increase in the prediction error (see Fig 6). The computational times for the compressed and traditional methods are reported in Table IV. For the purpose of benchmarking, we tested the compressed and uncompressed prediction algorithms on 2.76 GHz MacPro server on a single core with 32GB of random-access memory (RAM). Note that the MacPro server has 32 cores, and therefore, the total computational time for updating the predictions for a single horizon in the entire network is about 3s for the uncompressed method, compared to 0.3s for the compressed method with a compression factor of 10. We assume that training phases are performed offline for both methods. Prediction time for compressed method involves the necessary time to predict traffic variable for subset of the links and time required to perform network wide extrapolation. As Table IV shows, the latter can be neglected. Consequently, the required computational time for compressed prediction is proportional to the number of the road segments where speed is explicitly predicted. In other words, the reduction in computational complexity is (approximately) proportional to the compression ratio of the compressed method (see Table IV).

Compression Ratio	2	4	6	8	10
SVR	45.82	22.91	15.27	11.45	9.16
Matrix multiplication	0.33	0.24	0.15	0.12	0.11
Total	46.15	23.15	15.42	11.57	9.27
Complexity Savings	49.6%	74.7%	83.2%	87.4%	89.9%

TABLE IV: Computational time (in seconds) of the compressed method. The traditional approach requires 91.63 sec to perform prediction for the whole network.

VI. Conclusions

In this paper we applied column based (CX) method for three traffic applications: compression, compressed sensing and prediction. For compression of traffic data, we represented collected data as a product of two low-rank matrices: (i) traffic data for a subset of road segments; (ii) relationship matrix. Such network representation, explicitly expressed in terms of original roads, is stepping stone towards compressed sensing and prediction. In the former, we assess the state of the entire network by explicitly monitoring specific road segments. In the latter, we provide large-scale prediction by using the prediction models for a small subset of roads. We demonstrated that CX compressed scheme provides competitive performance compared to principal component analysis (PCA). Furthermore, we showed that proposed method significantly reduces the computational cost at the expense of a slightly increased prediction error. Therefore, proposed method may be used to store traffic data for later analysis (CX compression) and for real-time applications (compressed sensing and prediction). For future work, we propose to develop route guidance algorithm, based on future traffic conditions. This seems to be significant improvement, in comparison with the existing algorithms which rely on current and/or historical data. Also, we propose to test column based (CX) decomposition methods for similar connected systems such as power grids and water distribution networks.

VII. Acknowledgment

Justin Dauwels wishes to thank Andrzej Cichocki for helpful discussions.

References

[1] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 141–154.

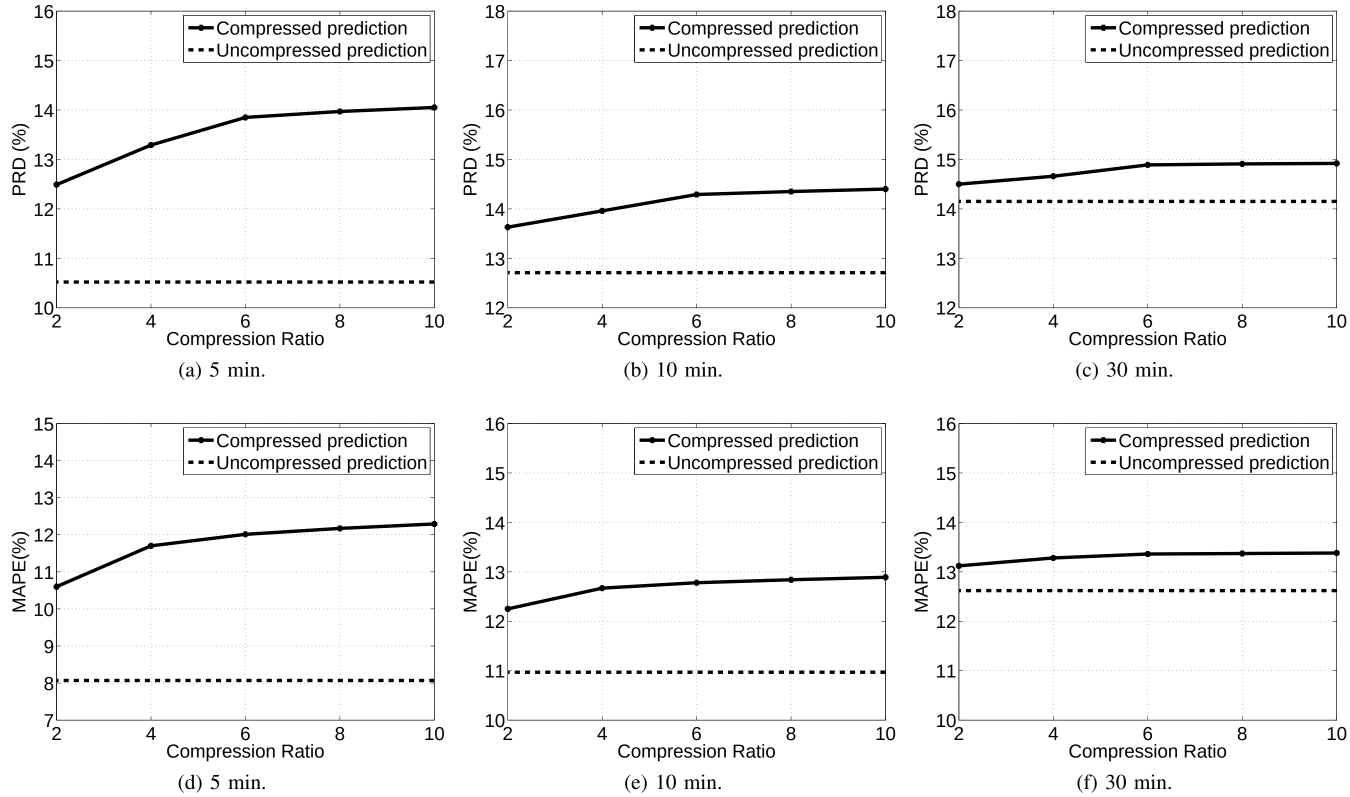
[2] Q. Li, H. Jianming, and Z. Yi, "A flow volumes data compression approach for traffic network based on principal component analysis," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*. IEEE, 2007, pp. 125–130.

[3] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1624–1639, 2011.

[4] T. Djukic, G. Flotterod, H. van Lint, and S. Hoogendoorn, "Efficient real time od matrix estimation based on principal component analysis," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 115–121.

[5] T. Djukic, J. W. van Lint, and S. Hoogendoorn, "Application of principal component analysis to predict dynamic origin-destination matrices," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2283, no. 1, pp. 81–89, 2012.

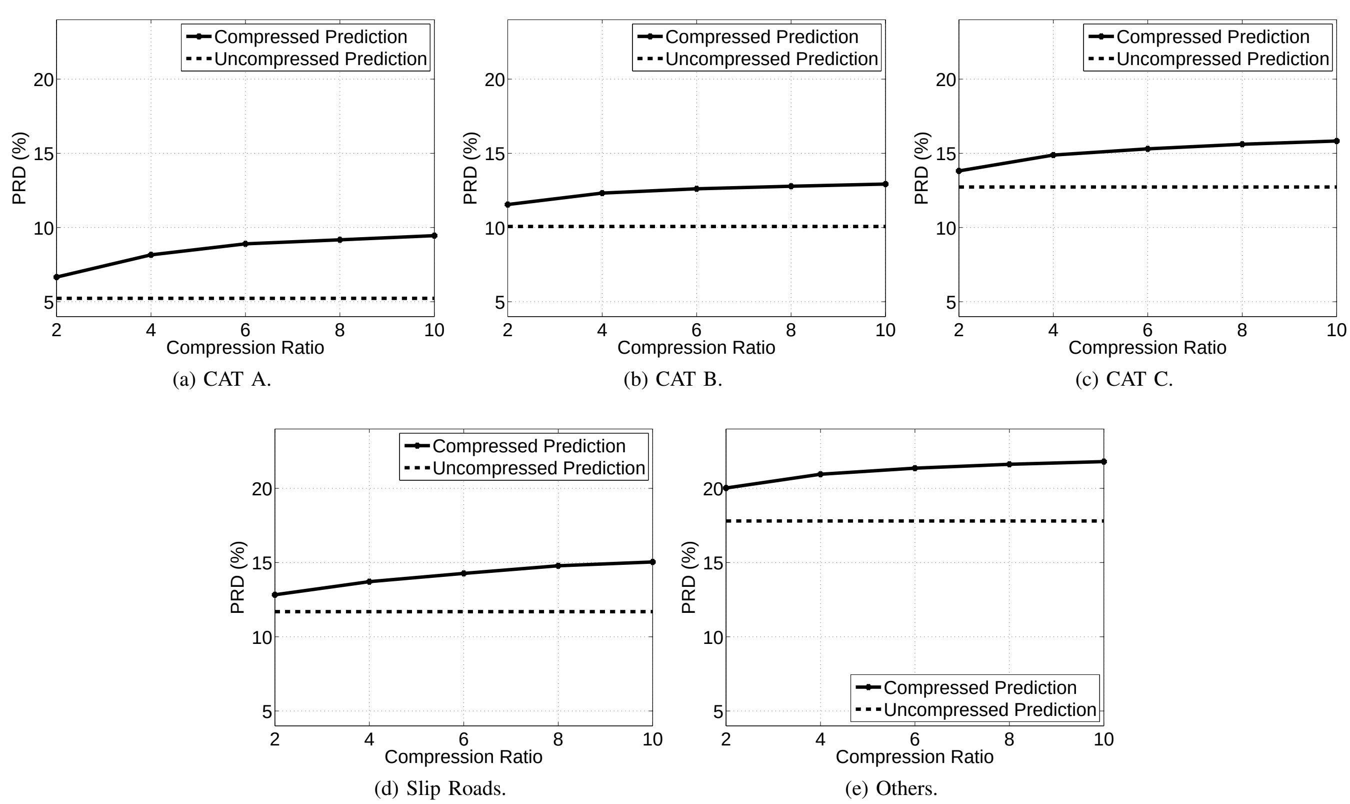
[6] M. T. Asif, S. Kannan, J. Dauwels, and P. Jaillet, "Data compression techniques for urban traffic data," in *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2013 IEEE Symposium on*, 2013, pp. 44–49.



(a) 5 min. (b) 10 min. (c) 30 min.

(d) 5 min. (e) 10 min. (f) 30 min.

Fig. 6: Prediction performance of the proposed and traditional prediction methods for 5 (left), 10 (middle) and 30 minutes (right) prediction horizons. The figures show the PRD (top) and the MAPE (bottom) error.



(a) CAT A. (b) CAT B. (c) CAT C.

(d) Slip Roads. (e) Others.

Fig. 7: Prediction performance of the proposed and traditional prediction methods for different road categories and 5 min prediction horizon.

[7] N. Mitrovic, M. T. Asif, U. Rasheed, J. Dauwels, and P. Jaillet, "CUR decomposition for compression and compressed sensing of large-scale traffic data," in *Intelligent Transportation Systems (ITSC), 2013 16th International IEEE Conference on*, oct. 2013, pp. 1475–1480.

[8] M. P. Miska, *Microscopic online simulation for real time traffic management*. Netherlands TRAIL Research School, 2007, no. Trail Thesis Series T2007/1.

[9] S. Sundaram, H. N. Koutsopoulos, M. Ben-Akiva, C. Antoniou, and R. Balakrishna, "Simulation-based dynamic traffic assignment for short-term planning applications," *Simulation Modelling Practice and Theory*, vol. 19, no. 1, pp. 450–462, 2011.

[10] M. Ben-Akiva, M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani, "Network state estimation and prediction for real-time traffic management," *Networks and Spatial Economics*, vol. 1, no. 3-4, pp. 293–318, 2001.

[11] Y. Wen, "Scalability of dynamic traffic assignment," Ph.D. dissertation, Massachusetts Institute of Technology, 2008.

[12] Y. Zhang and Y. Liu, "Traffic forecasting using least squares support vector machines," *Transportmetrica*, vol. 5, no. 3, pp. 193–213, 2009.

[13] J. Park, D. Li, Y. Murphey, J. Kristinsson, R. McGee, M. Kuang, and T. Phillips, "Real time vehicle speed prediction using a neural network traffic model," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, 2011, pp. 2991–2996.

[14] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 871–882, 2013.

[15] M. Asif, J. Dauwels, C. Goh, A. Oran, E. Fathi, M. Xu, M. Dhanya, N. Mitrovic, and P. Jaillet, "Spatiotemporal patterns in large-scale traffic speed prediction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–11, 2013.

[16] C. Quek, M. Pasquier, and B. Lim, "Pop-traffic: A novel fuzzy neural approach to road traffic analysis and prediction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 2, pp. 133–146, 2006.

[17] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C*, vol. 19, no. 4, pp. 606 – 616, 2011.

[18] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4. ACM, 2009, pp. 267–278.

[19] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Relative-error cur matrix decompositions," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 844–881, 2008.

[20] C. Boutsidis, M. W. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2009, pp. 968–977.

[21] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos, "Less is more: Sparse graph mining with compact matrix decomposition," *Statistical Analysis and Data Mining*, vol. 1, no. 1, pp. 6–22, 2008. [Online]. Available: http://dx.doi.org/10.1002/sam.102

[22] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proceedings of the National Academy of Sciences*, vol. 106, no. 3, pp. 697–702, 2009.

[23] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.

[24] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, 2010.

[25] Y. Han and F. Moutarde, "Analysis of network-level traffic states using locality preservative non-negative matrix factorization," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 501–506.

[26] Y. Han and F. Moutarde, "Statistical traffic state analysis in large-scale transportation networks using locality-preserving non-negative matrix factorisation," *Intelligent Transport Systems, IET*, vol. 7, no. 3, pp. 283–295, 2013.

[27] M. T. Asif, J. Dauwels, C. Y. Goh, A. Oran, E. Fathi, M. Xu, M. M. Dhanya, N. Mitrovic, and P. Jaillet, "Unsupervised learning based performance analysis of n-support vector regression for speed prediction of a large road network," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, sept. 2012, pp. 983 –988.

[28] M. Asif, N. Mitrovic, L. Garg, J. Dauwels, and P. Jaillet, "Low-dimensional models for missing data imputation in road networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 3527–3531.

[29] W.-C. Hong, "Traffic flow forecasting by seasonal svr with chaotic simulated annealing algorithm," *Neurocomputing*, vol. 74, no. 12, pp. 2096–2107, 2011.

[30] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson, "Shrinking the tube: a new support vector regression algorithm," *Advances in neural information processing systems*, pp. 330–336, 1999.

[31] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[32] S. Keerthi and C. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural computation*, vol. 15, no. 7, pp. 1667–1689, 2003.

[33] C. Chang and C. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[34] J. Dauwels, S. Kannan, R. Ramasubba, and A. Cichocki, "Near-lossless multi-channel EEG compression based on matrix and tensor decompositions," *Biomedical and Health Informatics, IEEE Journal of*, vol. PP, no. 99, p. 1, 2012.

Nikola Mitrovic received the Bachelor of Engineering degree from the University of Traffic and Transportation Engineering, Belgrade, Serbia. in 2009. He obtained Masters degree from the Department of Civil engineering at Florida Atlantic University, USA, in 2011. He is currently a Ph.D. student in the School of Electrical and Electronic Engineering at Nanyang Technological University in Singapore. His research topics are ITS, traffic modeling, and transportation planning.

Muhammad Tayyab Asif received the Bachelor of Engineering degree from the University of Engineering and Technology Lahore, Lahore, Pakistan. He is currently working toward the Ph.D. degree in the School of Electrical and Electronic Engineering,College of Engineering, Nanyang Technological University, Singapore. Previously, he was with Ericsson as a Design Engineer in the domain of mobile packet core networks. His research interests include sensor fusion, network optimization, and modeling of large-scale networks.

Justin Dauwels is an Assistant Professor with School of Electrical & Electronic Engineering at the Nanyang Technological University (NTU) in Singapore. His research interests are in Bayesian statistics, iterative signal processing, and computational neuroscience. He obtained the PhD degree in electrical engineering at the Swiss Polytechnical Institute of Technology (ETH) in Zurich in December 2005. He was a postdoctoral fellow at the RIKEN Brain Science Institute (2006-2007) and a research scientist at the Massachusetts Institute of Technology (2008-2010). He has been a JSPS postdoctoral fellow (2007), a BAEF fellow (2008), a Henri-Benedictus Fellow of the King Baudouin Foundation (2008), and a JSPS invited fellow (2010,2011). His research on Intelligent Transportation Systems (ITS) has been featured by the BBC, Straits Times, and various other media outlets. His research on Alzheimer's disease is featured at a 5-year exposition at the Science Centre in Singapore. His research team has won several best paper awards at international conferences. He has filed 5 US patents related to data analytics.

Patrick Jaillet received the Ph.D. degree in operations research from the Massachusetts Institute of Technology, Cambridge, USA, in 1985. He is currently the Dugald C. Jackson Professor in the Department of Electrical Engineering and Computer Science, and the co-director of the Operations Research Center, Massachusetts Institute of Technology, Cambridge, USA. His research interests include algorithm design and analysis for on-line problems; real-time and dynamic optimization; network design and optimization; and probabilistic combinatorial optimization.